

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ХАРЧОВИХ ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

в.о.Ректора _____ Іванов С.В.
_____ 2011р.

О.М. ПУПЕНА

ПРОМИСЛОВІ МЕРЕЖІ ТА ІНТЕГРАЦІЙНІ ТЕХНОЛОГІЇ

КУРС ЛЕКЦІЙ

ЧАСТИНА 1. ПРОМИСЛОВІ МЕРЕЖІ.

ля студентів напрямку 6050202
«Автоматизація та комп'ютерно-інтегровані технології» денної та заочної форм
навчання

Реєстраційний номер
електронних методичних вказівок
у НМУ _____

СХВАЛЕНО
на засіданні кафедри автоматизації
та комп'ютерно-інтегрованих
технологій
Протокол №__ від _____ 2011 р.

КИЇВ НУХТ 2011

О.М. ПУПЕНА. Промислові мережі та інтеграційні технології: курс лекцій для студ. напрямку 6050202 «Автоматизація та комп'ютерно-інтегровані технології» денної та заочної форм навчання. Частина 1. Промислові мережі. – К.: НУХТ, 2011. – 51 с.

Укладачі: **О.М. Пупена**, канд. техн. наук

1. МІСЦЕ ТА РОЛЬ МЕРЕЖНИХ ТЕХНОЛОГІЙ В ІНТЕГРОВАНИХ АВТОМАТИЗОВАНИХ СИСТЕМАХ УПРАВЛІННЯ

ПЛАН.

- 1.1. Інтегровані автоматизовані системи управління
- 1.2. Функціональна інтеграція
- 1.3. Інформаційна інтеграція
- 1.4. Технічна та програмна інтеграція
- 1.5. Обчислювальні мережі в ієрархії ІАСУ

1.1. Інтегровані автоматизовані системи управління

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

Інтегрована автоматизована система (**ІАС**) – сукупність двох або більше взаємопов'язаних автоматизованих систем (**АС**), в якій функціонування однієї з них залежать 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). **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 **СІМ** (Computer Integrated Manufacturing – комп'ютерно-інтегроване виробництво) та **СІР** (Computer Integrated Processing – комп'ютерно-інтегрований процес).

Однією з основних функцій **горизонтальної інтеграції** (рис.1.) є забезпечення інформаційної взаємодії між існуючими підсистемами одного рівня. На рівні АСУТП горизонтальна інтеграція передбачає об'єднання між собою АС управління технологічних і виробничих процесів, а також адміністративних відділень виробництва у єдину систему. На рівні АСУП горизонтальна інтеграція призначена для об'єднання робочих станцій організаційно-економічного та планового рівня управління підприємством.

Вертикальна інтеграція призначена для інтеграції 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.

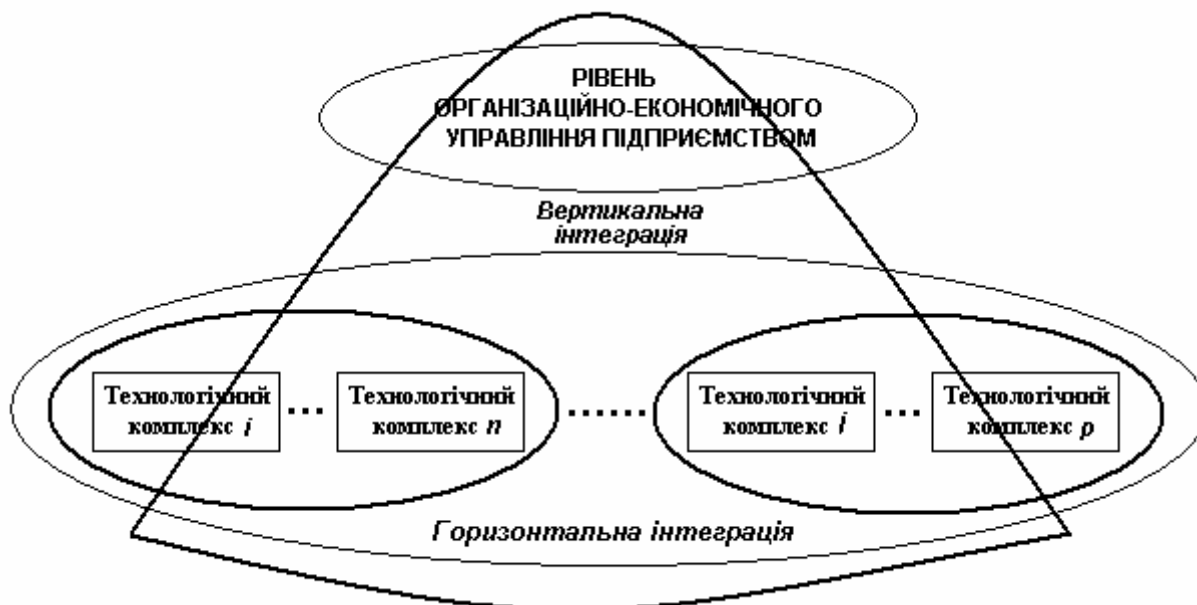


Рис.1. Напрями інтеграції в автоматизованих системах управління

Створення ІАСУ передбачає роботи по об'єднанню 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 по всім рівням 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. **Інформаційна інтеграція** полягає у створенні умов, за яких можливий доступ до всіх необхідних даних для реалізації функцій системи. **Програмна інтеграція** полягає у забезпеченні сумісного функціонування окремих складових програмного забезпечення з метою їх взаємодії.

Технічна інтеграція полягає у поєднанні 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.

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. При створенні ІАСУ розроблюють моделі і алгоритми вирішення функціональних задач та алгоритми зв'язку і узгодження локальних задач.

У результаті декомпозиції отримують багаторівневу ієрархічну функціональну структуру, для ІАСУ вона складається з наступних 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 завдання 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 ділянки 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. Задача інформаційної інтеграції є забезпечення доступу 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, реалізовані на різних вузлах об'єднані між собою мережними інформаційними потоками.

1.4. Технічна та програмна інтеграція

Технічна та програмна інтеграція проводиться з урахуванням специфіки використовуваних засобів та технологій, на яких базується їх робота. Програмно-технічні засоби (**ПТЗ**), які використовуються в інтегрованих автоматизованих системах управління можна розділити на два класи: ПТЗ рівня АСУТП та ПТЗ рівня АСУП.

ПТЗ рівня АСУТП, можна класифікувати по функціональному призначенню 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 і інформаційними зв'язками.

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:

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

Для засобів людино-машинного інтерфейсу виділимо наступні 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 можна віднести:

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

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;

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 по запиту;
- наявність реального часу;

Взагалі, 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 та навпаки та обмінюються даними процесу з процесорним вузлом (контролером).

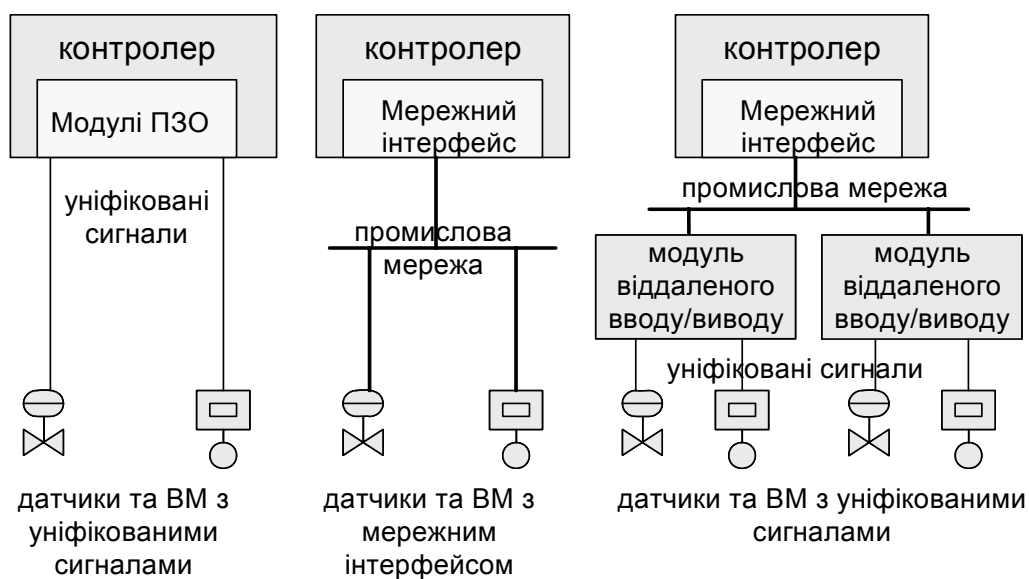


Рис.2. Різні методи підключення периферійних пристроїв до ПЛК

Порівняно з технологією передачі уніфікованого сигналу (0-10В, 4-20 мА) цифровий зв'язок дає наступні переваги:

- менш чутливий до зовнішніх перешкод;
- 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.

Окремо необхідно виділити засоби управління приводами (POWER DRIVE SYSTEMS – **PDS**), до яких відносяться частотні перетворювачі, сервоприводи та приводи з позиціонуванням. Все що стосується особливостей обміну даними для вище наведених типів периферійних засобів, може бути актуальними і для PDS. З'єднання по цифровому зв'язку 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.

Взагалі, 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 або комп'ютери з встановленим спеціальним програмним забезпеченням або спеціалізовані пристрої. Програматори мають інтерфейс для підключення до мережі, який використовується тільки при необхідності перепрограмування або переконфігурування. Виділимо такі характеристики зв'язку:

- підключення до системи тільки при необхідності;
- спеціально виділений логічний канал зв'язку;
- наявність команд на запис конфігурації, діагностики, зупинки, рестарту і інше.;
- низька пріоритетність повідомлень;
- можливість гарячого переконфігурування системи зв'язку;

Процес конфігурації одного вузла мережі, не повинен заважати її нормальному функціонуванню. Низька пріоритетність повідомлень дає можливість не заважати реальному часу обміну між іншими вузлами на тій самій мережі.

Програматори/конфігуратори в основному оперують конфігураційними або іншими словами параметричними даними. Винятком є режим відладки, коли налаדчику необхідно спостерігати та змінювати дані процесу.

1.4.5. Програмно-технічні засоби рівня АСУП

Як вже зазначалося програмно-технічними засобами рівня АСУП є комп'ютери. На сьогоднішній день для програмного забезпечення цього рівня в області ІАСУ найбільш популярною апаратною платформою є ІВМ-РС сумісні комп'ютери з ОС WINDOWS.

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

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

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

- відновлення даних по запити;
- відсутність реального часу;
- необхідність в обміні великими обсягами різномірної інформації;
- великий обсяг даних для загального доступу;

1.5. Обчислювальні мережі в ієрархії ІАСУ

На сьогоднішній день технології інформаційної, технічної та програмної інтеграції базуються на цифрових технологіях з використанням обчислювальних мереж. Тобто інтегроване виробництво сьогодні – це єдина система, що являє собою об'єднання різнотипних мікропроцесорних вузлів з використанням обчислювальних мереж.

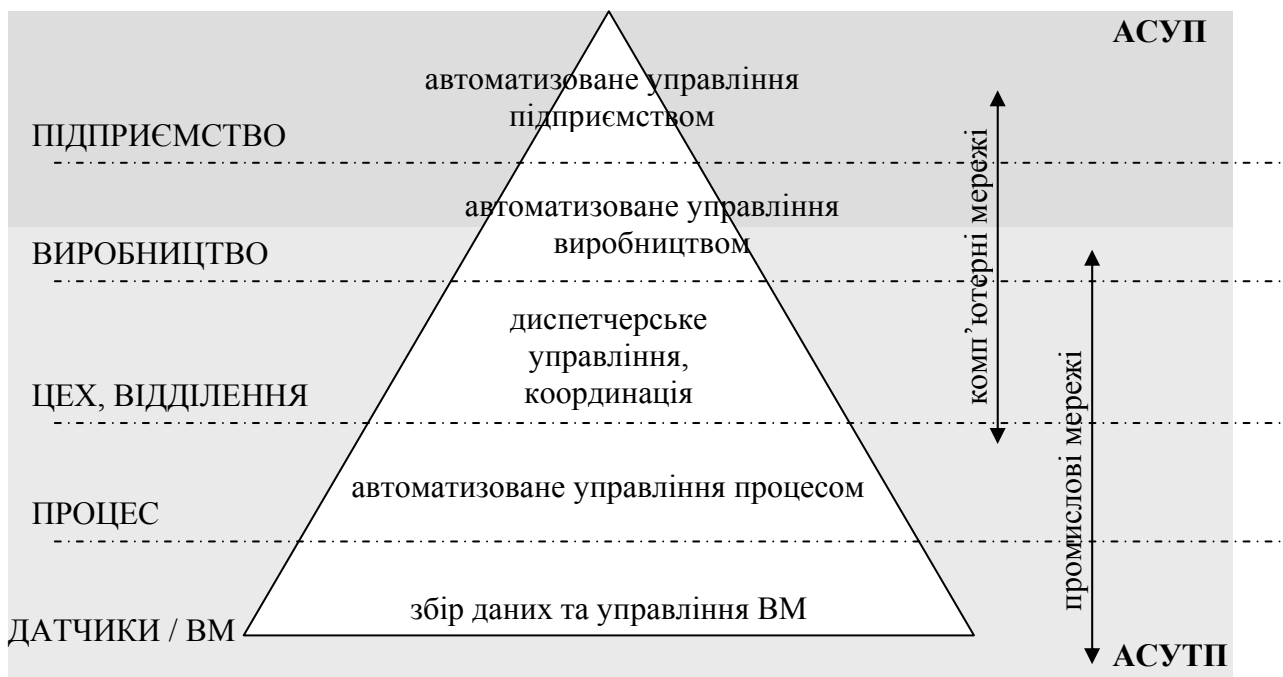


Рис. 3. Обчислювальні мережі в інтегрованих автоматизованих системах управління.

У даній дисципліні розглядаються технології інформаційної, технічної та програмної інтеграції, які використовуються в інтегрованих автоматизованих системах управління: промислові мережі та технології міжпрограмного зв'язку.

Література: [1], розділ 1.

2. ВЛАСТИВОСТІ ПРОМИСЛОВИХ МЕРЕЖ ПЛАН.

- 2.1. Визначення промислової мережі та їх градація.
- 2.2. Функціональне призначення промислових мереж.
- 2.3. Обмін даними.
- 2.4. Загальні вимоги до промислових мереж.
- 2.5. Історія розвитку та стандартизації промислових мереж.

2.1. Визначення промислової мережі та їх градація

Згідно українських стандартів 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). 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.

У даній дисципліні англomовний термін *Fieldbus* (дослівний переклад „польова шина”) переводиться як "промислова мережа", а під *промисловою мережею* розуміється промислова комунікаційна система з послідовною передачею бітів, яка використовує металевий кабель, оптоволокну або радіохвилі для зв'язку між мікропроцесорними засобами автоматизації. Слід зазначити, що в деяких джерелах терміни "польова шина" та "промислова мережа" не являються синонімами.

У подальшому при розгляді матеріалу будемо використовувати умовний розподіл промислових мереж в залежності від області застосування на два рівня.

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 підрівня: 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 (розподіленим засобами вводу/виводу, приводними засобами, операторськими терміналами та панелями). *Рівень контролерів (Controller level або Cell Level)*, промислові мережі якого приймають участь у вирішенні завдань по керуванню виробництвом в цілому або комплексом технологічних процесів і забезпечують обмін між контролерами, засобами SCADA/HMI та засобами рівня АСУП.

2.2. Функціональне призначення промислових мереж

До типових функцій промислових мереж можна віднести:

- Обмін даними процесу в реальному часі.
- Програмування та конфігурація вузлів.

- Діагностика вузлів.
- Управління станом вузла.
- Функції резервного переключення між вузлами мережі.

Доставка даних процесу в реальному часі є одним із показників ефективності конкретної реалізації мережі. Фізичні рамки реального часу чисельно визначаються для конкретної системи управління. Відповідно до стандартів ДСТУ **режимом реального часу** називається режим оброблення даних, який забезпечує взаємодію обчислювальної системи з зовнішніми, по відношенню до неї, процесами у темпі порівнянному зі швидкістю протікання цих процесів.

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

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. Функція вияву факту несправності базується на високо-пріоритетних повідомленнях, ідентифікація аварії особливих вимог до реального часу не потребує.

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

За рахунок цих функцій резервування вузлів в системі 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). У нормальному режимі функціонування у системі працюють основні вузли, у випадку виходу яких з ладу, їх функції беруть на себе резервні. 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).

2.3. Обмін даними

За призначенням. Для забезпечення вищевказаних функцій, в мережі циркулюють дані та команди. У 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 другі – для функцій програмування/конфігурування та функцій діагностики що відповідають за ідентифікацію несправності.

За форматом даних. В залежності від типу обміну (обмін даними процесу або параметричними даними) дані можуть бути наступних 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);

За відновленням. В одній і тій самій системі можуть бути дані, які необхідно відновлювати постійно (циклічно або періодично) та по 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) по запиту:** дані надходять до споживача після його запиту.

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 в мережу.

При передачі даних говорять 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.

2.4. 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 наступним вимогам.

- Властивість детермінованості.
- Завадостійкість та промислові умови експлуатації.
- Надійність та живучість.
- Простота, зручність інсталяції та обслуговування.
- Можливість подачі живлення по кабельній системі мережі.
- Вільна топологія.

Детермінованість. Промислова мережа повинна надавати можливість обміну обидвома типами даних за призначенням, тобто надавати час для **реал-**

тайм трафіку (для обміну даними процесу) та *не реал-тайм трафіку* (для обміну параметричними даними). Це можливо при умові визначення пріоритетності повідомлень для обміну даними процесу, або розділення загального мережного часу на циклічно-періодичний обмін даними процесу та ациклічний обмін параметричними даними.

Завадостійкість та промислові умови експлуатації. У промислових умовах експлуатації поряд з кабелем мережі може знаходитися силове електрообладнання, 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 і інше.

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

Щодо пило- та вологозахисту технічних засобів в промислових мережах, згідно стандартів ІЕС 529, EN 60529 та ГОСТ 14254-96, всі електричні засоби прийнято класифікувати та маркувати по ступені захисту **ІР-кодом** (Ingress Protection). ІР-код - це двохзначне число, перша цифра якого вказує на ступінь захисту електрообладнання від твердих тіл, а друга – від води.

Стандарт ІЕС 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.

Надійність та живучість. 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 всієї мережі.

Простота, зручність інсталяції та обслуговування. Обслуговуючий персонал повинен швидко виявити причину зупинки та замінити дефектну частину системи, при цьому не зупиняючи функціонування мережі. Процедура заміни або добавлення нового вузла повинна проходити швидко та легко.

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 по тому ж самому кабелю, що і цифровий сигнал, а інколи і по тій самій інформаційній парі проводів, створюючи несучий сигнал для модуляції.

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.

Таблиця 1.

Характеристики промислових мереж (Підсумкова таблиця розділу 2.1)

Визначення	Промислова мережа – промислова комунікаційна система з послідовною передачею бітів, яка використовує металевий кабель, оптоволокно або радіохвилі для зв'язку між мікропроцесорними 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: а. обмін даними процесу; б. обмін параметричними даними; 2. Формати даних: числові (аналогові); дискретні (бінарні); часові; масиви (блоки даних); структурні даністрокові. 3. Спосіб відновлення даних - 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 - синхронна передача - асинхронна передача
Вимоги до промислових мереж	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 по кабельній системі мережі. 6. Вільна топологія.

2.5. Історія розвитку та стандартизації промислових мереж

Виділимо п'ять етапів розвитку стандартів промислових мереж:

1986-1990 рр. – поява нових 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 MEK.

У межах європейської організації *CENELEC* всі національні стандарти при розгляді компілювалися в єдиний стандарт в тій же редакції: EN 50170 – загального призначення, EN 50254 – високошвидкісні системи обміну невеликими об'ємами даних, та EN 50325 – мережі, які базуються на CAN-технологіях.

В 1999 році, після численних "fieldbus-війн" був підписаний "Меморандум про порозуміння" між основним гравцями ринку промислових мереж. Внаслідок прийнятого меморандуму був створений всеохоплюючий стандарт IEC 61158, який був прийнятий у 2000 р. та включав найбільш відомі системи промислових мереж.

На сьогоднішній день до стандарту включені ряд мереж, які представлені своїм профілем (CP), який відноситься до сімейства профілів (CPF). Представлені наступні технології CPF (в 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 сьогоднішній 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)

На сьогоднішній час стандарт охоплює всі аспекти розробки, впровадження та експлуатації промислових мереж.

Література: [1], розділ 2 (2.1, 2.2).

3. РОБОЧІ ХАРАКТЕРИСТИКИ ПРОМИСЛОВИХ МЕРЕЖ.

ПЛАН.

- 3.1. Промислові мережі в контексті моделі ISO OSI.
- 3.2. Основні робочі характеристики промислових мереж.
- 3.3. Прикладні сервіси, інтерфейс прикладного рівня
- 3.4. Забезпечення каналом зв'язку між вузлами та методи доступу
- 3.5. Формування кадру та контроль за помилками
- 3.6. Топологія мережі та сегментація
- 3.7. Фізичні інтерфейси передачі даних
- 3.8. Маршрутизація
- 3.9. Транспортування даних
- 3.10. Підсумкова таблиця з основними робочими характеристиками.

3.1. Промислові мережі в контексті моделі ISO OSI

Тільки при використанні принципів "Відкритих систем" інтеграція виробів різних виробників в одну мережу може бути вирішена без особливих проблем. У 1978 році Міжнародною організацією по 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 взаємозв'язку відкритих систем" (*OSI-модель*, Open System Interconnection Reference Model). На сьогоднішній день в стандартах МЕК визначена своя трирівнева модель архітектури промислових мереж, однак у технічній документації до опису функціонування мереж поки що прийнято давати їх архітектуру в контексті OSI.

У моделі OSI більшість мереж організуються у набори рівнів або слоїв (layers), кожен з яких має своє призначення. Ціллю кожного рівня є представлення деяких сервісів для вищого рівня і підтримування зв'язку з однойменним рівнем на іншій пристрої. Правила і домовленості, які використовуються у даному спілкуванні називаються *протоколом* рівня (layer protocol). Апаратуру або/і програму, яка займається задачами кожного з рівнів будемо називати *об'єктом рівня* або *сутністю рівня*. Кожен із рівнів у мережі надає набір сервісів (служб) для реалізації певних задач. *Сервіс* або служба (Service) – це набір операцій, які нижній рівень представляє верхньому. Сервіс визначає які саме операції рівень буде виконувати, але не оговорює яким чином. Між кожною парою суміжних рівнів знаходиться *інтерфейс* (interface), який визначає набір примітивних операцій, які надаються верхньому рівню від нижнього. Під інтерфейсом можна розуміти ті програмні чи апаратні функції, які надаються верхньому рівню для доступу до своїх сервісів.

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) повинен забезпечити достовірну доставку бітів від передавача до приймача.

2. Рівень передачі даних (Data Link Layer) або каналний рівень повинен забезпечити достовірну доставку даних від одного вузла до іншого в одній і тій самій мережі.

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

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

5. Сеансовий рівень (Session Layer) Дозволяє організувати сеанси обміну між прикладними програми (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) повинен забезпечити одній прикладній програмі доступ до об'єктів іншої прикладної програми через систему домовленостей.

3.2. Основні робочі характеристики промислових мереж

При ознайомленні з можливостями промислової мережі взагалі та конкретного вузла, звертають увагу на основні її робочі характеристики, по яким 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.

3.3. 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**. 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. Можна виділити відповідні до функцій 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.

Інтерфейс прикладного рівня. Для користування сервісами прикладного рівня користувачу надається інтерфейс прикладного рівня. Інтерфейс прикладного рівня може представляти собою комунікаційні функції, доступ до таблиці мережних об'єктів (мережні змінні, словник об'єктів) тощо. Деякі промислові мережі надають можливість графічного конфігурування обміну

даними на прикладному рівні, наприклад через мову функціональних блоків (FF, LONWorks).

Взаємодія між прикладними Процесами. Для того, щоб прикладні Процеси на вузлах могли обмінюватися даними, необхідно налаштувати між ними зв'язок. В залежності 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 реалізації протоколу.

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. В залежності від того, який прикладний Процес генерує публікацію, виділяють два типи моделі Видавець-Абонент (рис.4):

- *pull model*, коли момент публікації визначає прикладний Процес одного із вузлів, який у необхідний момент відправляє запит на публікацію (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.

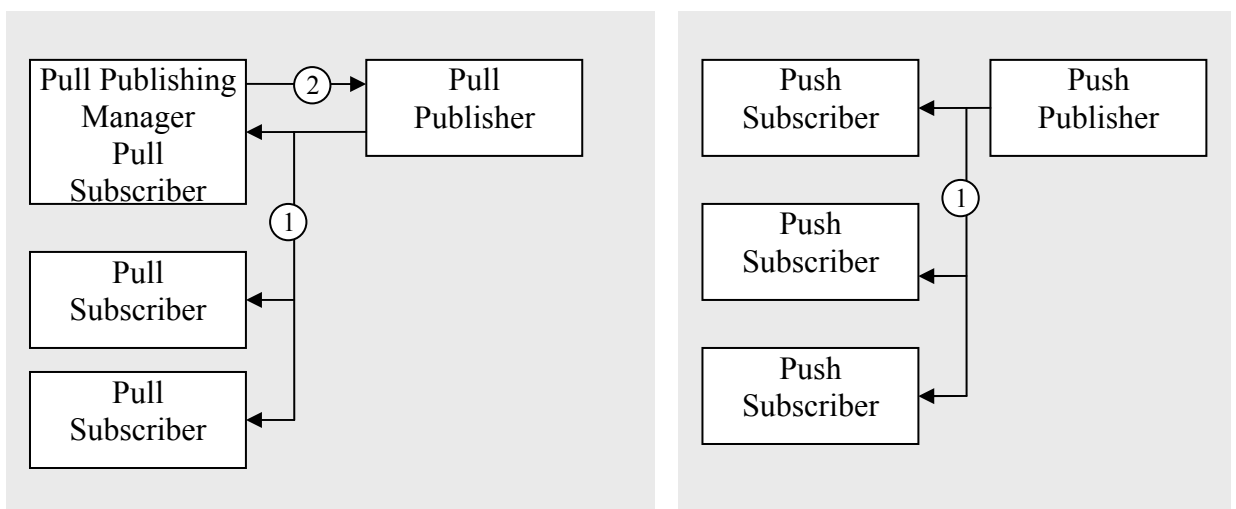


Рис.4. Функціонування моделі видавець-Абонент: pull model – ліворуч, push model – праворуч. 1 – публікація, 2 – запит на публікацію.

Модель Виробник-Споживач (Producer-Consumer) з точки зору користувача

аналогічна моделі Видавець-Абонент, за винятком того, що для адресації отримувачів використовується фільтрація по ідентифікатору повідомлень типу Виробник-Споживач-msg. Прикладний Процес, який в широкомовному режимі видає (виробляє) дані в мережу називається Виробником, а який їх приймає – Споживачем. Всі інші характеристики аналогічні моделі Видавець-Абонент, тому при розгляді функціонування мереж на прикладному рівні – ці моделі ототожнюються.

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 в прикладному Процесі вузла-джерела так і вузла-отримувача. Можна виділити два *способи ідентифікації даних*: вказати необхідні дані для обміну на початку функціонування мережі або вказувати область та формат даних безпосередньо в момент обміну між прикладними Процесами. Перший спосіб будемо називати **ідентифікованим обміном** (identified data), а другий **обміном повідомленнями** (messaging). Визначену область даних в ідентифікованому обміні будемо називати **ідентифікованими даними**.

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.

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).

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 по зміні	push	Publish-Subscribe (Producer-Consumer) ідентифікованого обміну
	ациклічний по запиту	клієнт-серверна модель обміну повідомленнями	
програмування/конфігурування вузла (обмін параметричними даними)		клієнт-серверна модель обміну повідомленнями	
управління станом вузла		клієнт-серверна модель 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	

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

Створенням каналу зв'язку між вузлами мережі займаються сутності каналного рівня. Канальний рівень оперує *кадрами (Frame)*, що являють собою послідовність інформаційних та службових байтів, які по чергово відправляються сутністю фізичного рівня. У ширококомовних мережах задачею каналного рівня є забезпечення доставки кадрів конкретному(-ним) адресатові. Так як одночасно шину прослуховують всі приймачі, необхідно ідентифікувати того, кому призначений даний кадр.

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

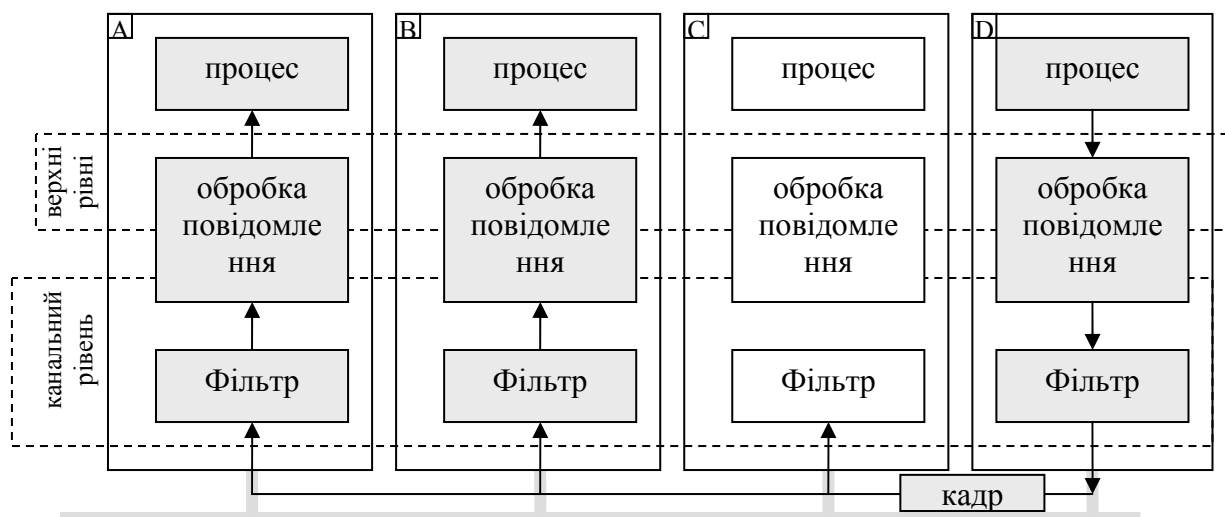


Рис.5. Відфільтрування кадрів на каналному рівні.

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).

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 в мережі має свою унікальну адресу. При формуванні кадру, каналний рівень вузла відправника додає до нього заголовки з адресою (-сами) вузлів отримувачів. Всі вузли отримують цей кадр, але тільки вузли зі вказаною адресою пропускають 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)**.

Ряд мереж з загальним доступом використовують інший спосіб фільтрації кадрів – на основі ідентифікаторів повідомлення, який будемо називати **Виробник-Споживач-msg (Producer-Consumer-msg)**. При такому способі адресації доставки кадрів, кожний кадр 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 кожен кадр що передається по мережі 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.

Методи доступу. Для ширококомовних мереж (шинна та деревовидна топологія) необхідно визначити порядок доступу до єдиного каналу. Широкомовні мережі, які використовують у 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. Існує декілька варіантів вирішення порядку доступу до шини: централізований Ведучий-Ведений; централізований з Арбітром шини; децентралізований з маркерним кільцем; множинні випадкові методи доступу CSMA; множинний з використанням поля арбітражу; передбачуваний псевдо-постійний CSMA; TDMA; гібридні.

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.

3.5. Формування кадру та контроль за помилками

Потік бітів на фізичному рівні не застрахований від помилок. Кількість прийнятих бітів та значення прийнятих бітів може відрізнятись від переданих. Рівень передачі даних повинен визначити факт помилки. Для боротьби з помилками у мережах, до кадру добавляються допоміжні дані, які при його отриманні дозволяють отримувачу визначити помилку. Отримувач по надлишковим даним аналізує правильність отриманих даних, і якщо помилка присутня - робить повторний запит, або відправляє негативне підтвердження. Надлишкові дані, які добавляються до кадру для контролю за помилками являються результатом певної функції над інформаційними даними (**контрольною сумою** - Check Sum)

3.6. Топологія мережі та сегментація

Мережна топологія визначає фізичну структуру мережі. Вона накладає обмеження на розташування ліній зв'язку та впливає на функціонування сервісів канального рівня. Основними топологіями є: кільце, зірка, шина, лінійна та дерево (рис.6).

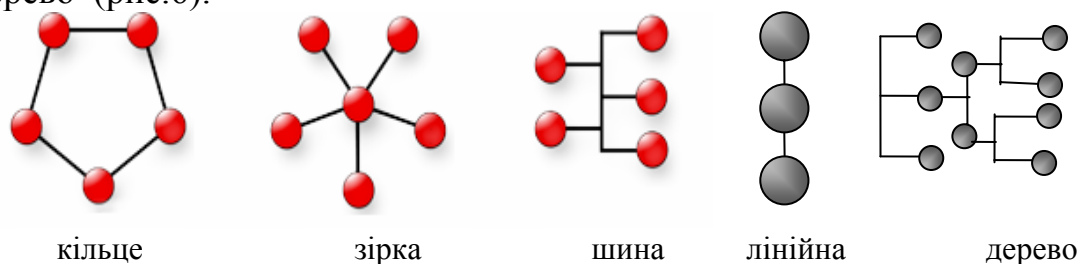


Рис. 6. Базові топології мереж

Сегментація. Під **фізичним сегментом** мережі розуміють фізичну частину мережі, відокремлену спеціальним мережним пристроєм:

- на фізичному рівні моделі OSI **репітером** (повторювачем) або **концентратором** (hub);
- на канальному 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 в мережі, максимальну довжину ліній зв'язку, однак обмежує швидкісні характеристики передачі. Концентратори використовуються для об'єднання декількох пристроїв в мережу, а також можуть виконувати функції репітерів, за рахунок чого територіальна "зірка" реалізовується в фізичну "шину". З точки зору засобів канального рівня мережа поєднана репітером або концентратором являється одним цілим.

Міст (bridge) забезпечує пересилку кадрів між сегментами по їх MAC-адресам. Тобто міст забезпечує фільтрацію кадрів, що циркулюють між сегментами по їх MAC-адресам, 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.

3.7. Фізичні інтерфейси передачі даних

На фізичному рівні для передачі даних може використовуватись 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.

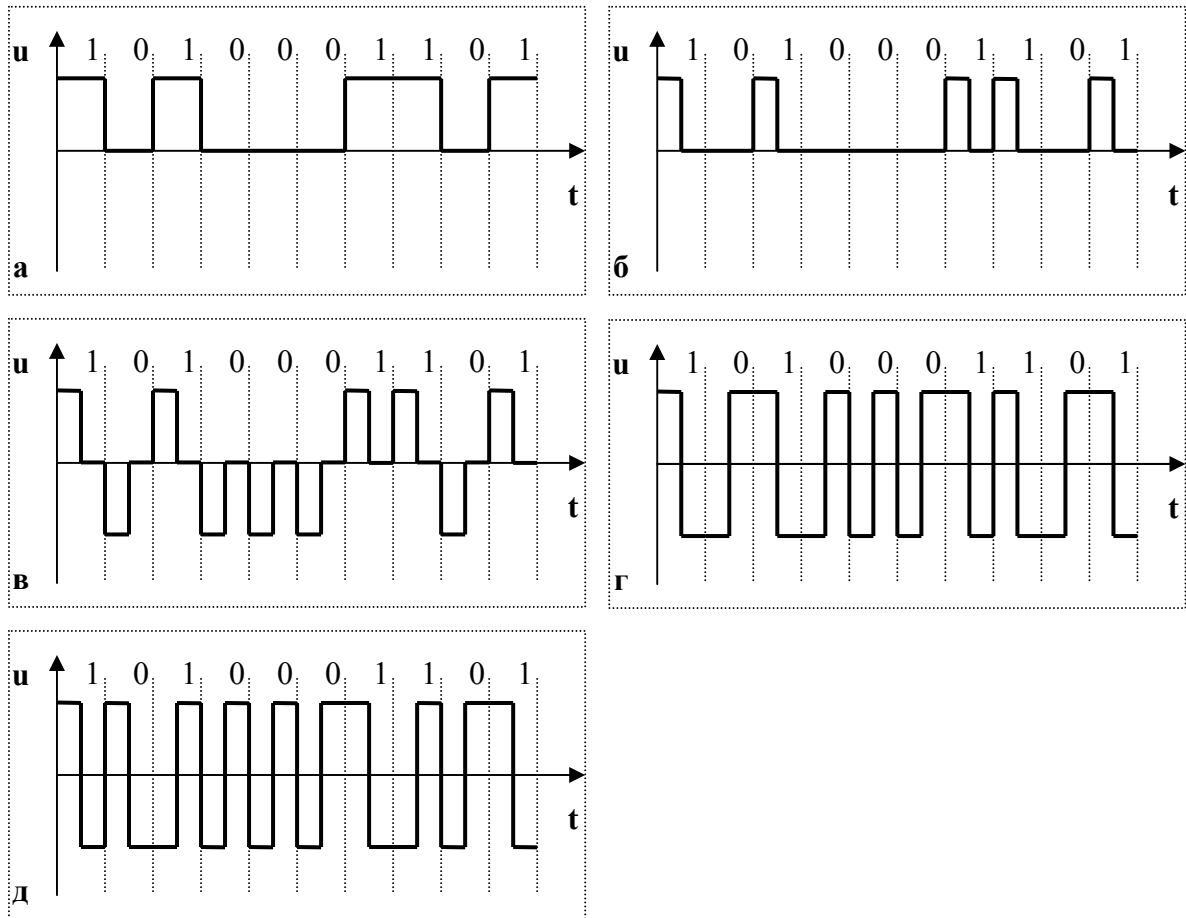


Рис.7. Способи цифрового кодування:

а – пряме двійкове без повернення в нуль (NRZ); б – пряме двійкове з поверненням в нуль (RZ); в – трирівневе з поверненням в нуль; г – манчестерське; д – диференційне манчестерське.

При модулюванні сигналу використовують амплітудну, фазову, частотну модуляцію, або їх поєднання. Передачу бітів можна реалізувати електричним сигналом по струму або по напрузі. При використанні напруги можуть застосовувати асиметричний або симетричний (диференційний) метод передачі.

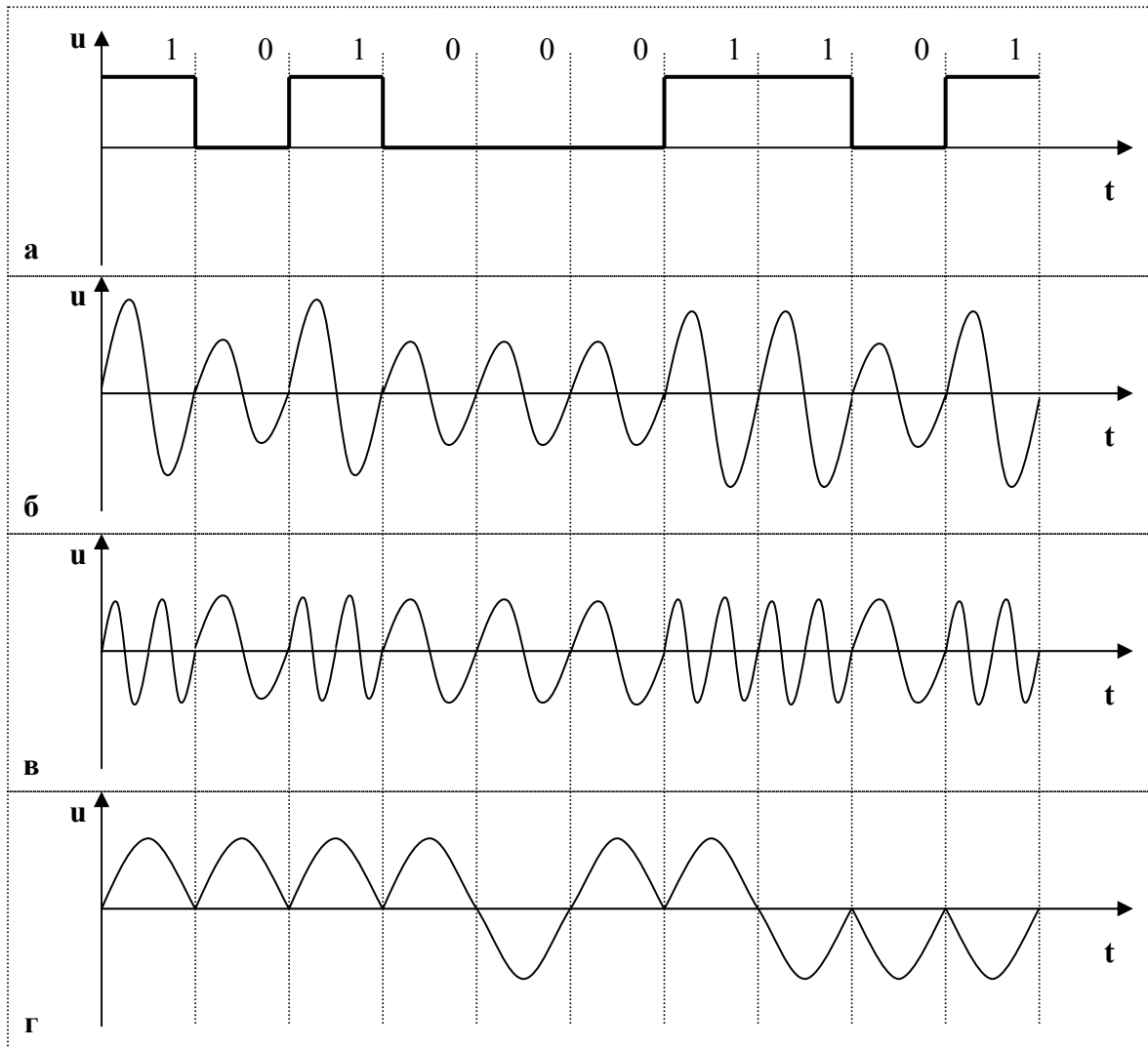


Рис. 8. Модуляція несучої: а – двійковий сигнал; б – амплітудна модуляція; в – частотна модуляція; г – фазова модуляція

Для підключення технічних засобів до мережі, та реалізацію кабельної інфраструктури використовують різні типи з'єднувачів (конекторів). Для промислових умов експлуатації їх роблять в герметичному виконанні, 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 цих інтерфейсів пов'язано з відносною дешевизною організації зв'язку (існують готові мікросхеми з їх реалізацією) та їх популярністю.

3.8. 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 сегменту в інший, навіть якщо ці сегменти відрізняються

свою архітектурою. Вузли, які займаються цією передачею називають маршрутизаторами. Об'єднання сегментів в цьому випадку будемо називати **інтермережею**, а самі сегменти – **підмережею**. Маршрутизатори одночасно підключаються до декількох мереж, тобто мають вихід на декілька фізичних ліній зв'язку. Їх задача отримати пакет з однієї мережі, по необхідності розбити на частини і відправити в іншу мережу по іншій лінії у відповідності з **маршрутною таблицею**. У маршрутній таблиці міститься інформація про те, яким чином можна потрапити в необхідну мережу (мережі).

Разом з маршрутизацією пакетів, мережний рівень підтримує ще ряд сервісів: маршрутизація пакетів; побудова маршрутних таблиць; роздача мережних адрес.

3.9. Транспортування даних

При доставленні 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. 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)**. Кожній прикладній сутностізначається свій порт, за допомогою якого до неї можна звернутися (серверний порт).

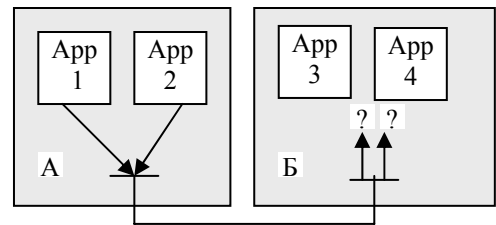


Рис.9. Вибір маршруту для даних.

3.10. 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 абрєвіатурами, які надалі будемо використовувати в посібнику.

Таблиця 3

Можливі варіанти реалізації основних характеристик промислових мереж.

OSI	характеристика	варіанти реалізації	коментар
	область призначення NetArea	- рівень датчиків - рівень контролерів - рівень датчиків/контролерів	деякі мережі призначені для кількох рівнів; рівень датчиків включає польовий рівень (field level)
прикладний	наявні сервіси AppService	- циклічний обмін даними процесу; - періодичний обмін даними процесу; - ациклічний по зміні значення обмін даними процесу; - ациклічний по запиту обмін даними процесу; - обмін параметричними даними	визначається набором протоколів прикладного рівня; вузол мережі не завжди підтримує всі сервіси;

		(програмування та конфігурування вузла); - управління станом вузлів; - діагностичні сервіси; - функції резервування;	
	модель обміну AppModel	- 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, AppProfile	- профілі не визначені; - перелік підтримуваних профілями типів вузлів;	вузол мережі як правило підтримує тільки один профіль пристрою;
	кількісна характеристика відновлення даних процесу AppProcData	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 AppResolut	кількість_інформації/час_оновлення	наводяться заявлені дані в офіційних джерелах
транспортний	наявні сервіси TpService	- 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 TpModel	- надійні: з встановлення з'єднання, з підтвердженнями (Acknowledge); - ненадійні: без встановлення з'єднання, без підтвердження;	можливі комбінації
мережний	наявні сервіси NtService	- маршрутизація пакетів; - побудова маршрутних таблиць; - роздача мережних адрес.	визначається набором протоколів мережного рівня; вузол мережі не завжди підтримує всі сервіси;
канальний	метод адресації ChAddModel	- Відправник-Адресат-msg (правило адресації); - Виробник-Споживач-msg;	визначається протоколом канального рівня; може включати обидві моделі
	метод доступу ChAccess	- Ведучий-Ведений; - Ведучий-Ведений з активними Веденими; - з Арбітром шини; - CSMA/CD, CSMA/CA, predective p-persistent CSMA; - TDMA, STDMA; - маркерний; - гібридний -	визначається протоколом канального рівня;
	контроль за помилками ChChecksum	- LRC; - CRC (різні варіанти); - інший;	визначається протоколом канального рівня;
	можливість сегментації ChSegment	- так: кількість сегментів, спосіб сегментації, обмеження; - ні;	використовуються мости та комутатори;
фізичний інтерфейс PhInterface		- стандартні: RS-232, RS-422, RS-485, CL, USB, BlueTooth, IR; - інші: передача напругою/струмом, метод	для багатьох мереж можливий вибір;

	кодування/модуляції, симетричний/асиметричний;	
фізичне середовище передачі PhMedia	- звичайний або спец. кабель (кількість жил); - вита пара (кількість витих пар); - екранована вита пара (кількість витих пар); - коаксіальний кабель; - оптоволокно (одно-, багатомодовий); - радіохвилі (тип);	для багатьох мереж можливий вибір;
топологія PhTopology	- шина; - зірка; - лінійна; - кільце; - дерево; - вільна;	для багатьох мереж можливий вибір; комбінація можлива при сегментації;
максимальна довжина відгалужень PhLdrop	- не допускається; - довжина в метрах, вплив на загальну довжину лінії, обмеження;	визначається тільки для шини та дерева; залежить від бітової швидкості;
бітова швидкість передачі PhBaudRate	- вибирається з ряду можливих (вказується мін-макс); - фіксована;	залежить від вибраного інтерфейсу, середовища передачі, довжини та кількості сегментів; обмеження вказуються
можливість сегментації PhSegment	- так (кількість сегментів); - ні;	залежить від інтерфейсу та середовища; використовуються репітери та концентратори;
максимальна кількість вузлів на сегмент PhNodes	максимальна кількість вузлів в сегменті, максимальна кількість всього в мережі	залежить від інтерфейсу та середовища;
максимальна довжина сегменту PhLength	максимальна довжина в метрах сегменту та всієї мережі; правила термінування;	
живлення по мережі PhSupply	- не дозволяється; - по інформаційних проводах, макс. струм; - по окремих проводах кабелю, макс. струм;	вказується напруга та максимальний струм; залежить від вузла мережі;

Література: [1], розділ 2 (2.3, 2.4), розділ 3 (3.6).

4. Стандарти послідовні інтерфейси.

- 4.1. Синхронна та асинхронна передача.
- 4.2. Інтерфейс RS-232.
- 4.3. RS-422
- 4.4. RS-485
- 4.5. Порівняльна характеристика стандартних послідовних інтерфейсів.
- 4.6. Використання адаптерів-перетворювачів інтерфейсів.

4.1. Синхронна та асинхронна передача

Необхідність синхронізації. При обміні даними між вузлами необхідно щоб передавач і приймач використовували одне і те саме джерело часу, тобто щоб вони були *синхронізовані*. У протилежному випадку, приймач не зможе визначити, коли надходить перший біт, тобто розпізнати початок посилки. 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.10. показана послідовність бітів, яку передає трансмітер і розпізнає ресивер. Пунктирними лініями показані такти кожного з пристроїв. Оскільки неможливо фізично організувати абсолютно однакові часові інтервали тактів в пристроях, то в певний інтервал часу передачі – біти будуть зміщені, отже посилка неправильно сприйнята (на рисунку виділено штриховою лінією).

При синхронній передачі приймач і передавач постійно синхронізуються під час передачі. Синхронізація проходить через певні проміжки часу або даних за допомогою *синхронізуючої посилки*, яка представляє 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 *преамбули*). У першому випадку необхідний допоміжний провід (канал), в другому – допоміжна смуга пропускання, в третьому – допоміжні біти.

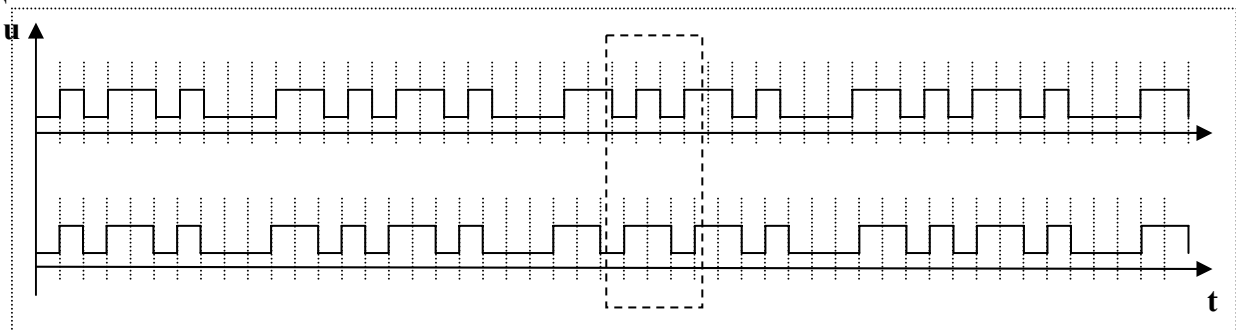


Рис. 10. Обмін інформацією між пристроями без синхронізації:
зверху – передавач, знизу – приймач.

При *асинхронній* передачі, обмін даними може проходити і без визначеного джерела часу. Передача починається в будь який момент часу з

сигналу початку передачі – стартового біта. В цей час 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 поступають.

Символьна передача. Один з асинхронних способів передачі є *символьна передача*. При такому способі, дані які передаються діляться на *символи* по декілька біт (як правило по 8 або 7) і обрамляються службовими бітами. На рис. 11 графічно представлено передачу двох байт $D8_{16}(11011000_2)$ та $C4_{16}(11000100_2)$ при наступних настройках: біт парності – «непарний», 1 стоповий біт, бітова швидкість 9600 біт/с. При відсутності передачі на лінії утримується логічна "1". При передачі символу передавач скидає сигнал на "0" і тримає його протягом 1-го біту, який називають *стартовим бітом*. За цей час приймач синхронізується з передавачем і готується до прийому *бітів даних*, які передаються відразу після стартового біту. 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 рівним 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 в бітах даних разом паритетним була непарною. Приймач перевіряє суму прийнятих одиничних бітів даних та паритету і якщо не вона не співпадає з типом наперед визначеного біта паритету, сигналізує про це верхнім рівням, які вирішують можливість повторного запиту.

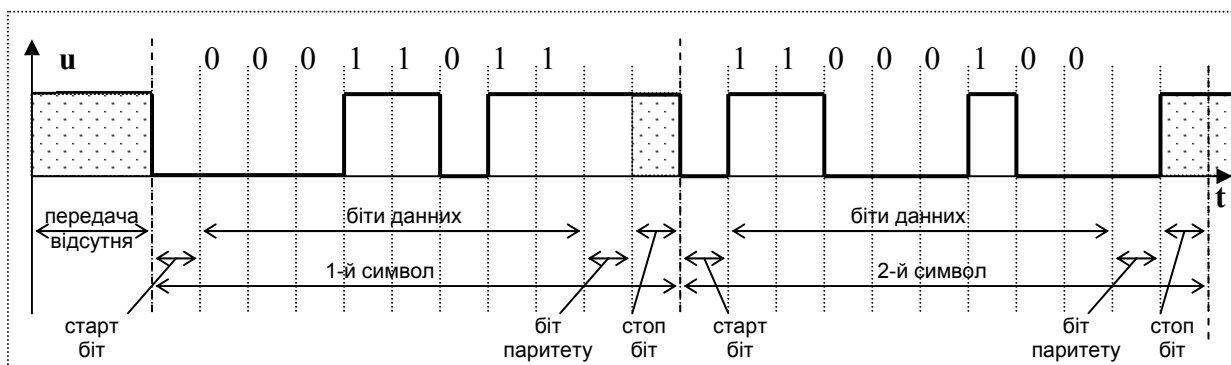


Рис. 11. Передача двох символів: 1-й – 11011000; 2-й – 00100011. Біт паритету – непарний; 1 стоповий біт

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.

Використання схем UART. Символьну передачу просто і недорого реалізувати завдяки існуванню спеціальної мікросхеми – універсального асинхронного приймача-передавача УАПП (*UART* – Universal Asynchronous

Receiver Transmitter), яка застосовується в багатьох пристроях. UART являється 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. Для прикладу на рис.3.22 показаний приклад використання UART разом з трансивером RS-232.

Різні схеми по типу UART використовуються також в синхронній передачі даних і називаються USRT, а їх комбінація **USART**.

При передачі UART послідовно виконує наступні дії: встановлює для трансивера бітову швидкість; приймає від мікропроцесору біти даних символу через паралельну шину; генерує для трансивера стартовий біт; послідовно передає трансиверу біти даних; визначає та передає трансиверу біт парності; передає трансиверу стопові біти; сигналізує процесору про готовність до наступного символу; при необхідності координує квітування;

При прийомі від трансиверу UART послідовно виконує наступні дії: встановлює для трансиверу бітову швидкість; розпізнає стартовий біт; зчитує біти даних, що поступають від трансиверу у вигляді послідовності бітів; зчитує біт парності і робить перевірку на парність; передає символ в паралельному вигляді мікропроцесору; при необхідності координує квітування; перевіряє дані на наявність помилки і при її знаходженні мітить в регістрі станів біт помилки.

Крім сигналів передачі (TD), прийому (RD) та шини даних в схемі UART також визначені додаткові управляючі сигнали для синхронізації як з мікропроцесором так і з трансивером.

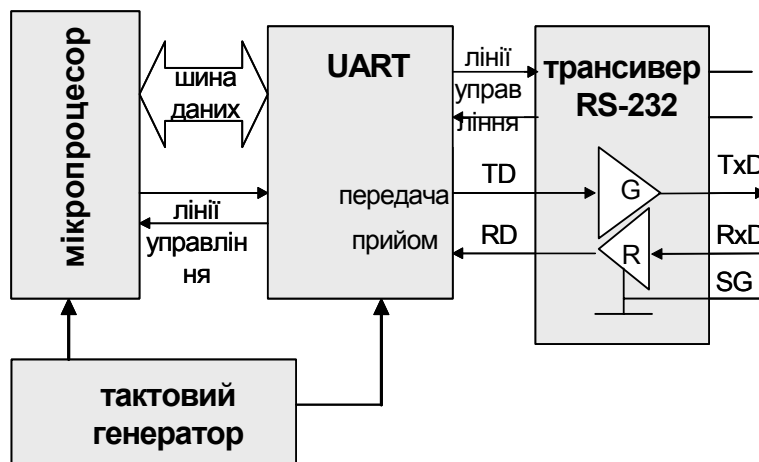


Рис. 12. Приклад використання UART з трансивером RS-232

4.2.Інтерфейс RS-232

Стандарт інтерфейсу **RS-232C** розроблений в 1969 році і опублікований асоціацією EIA як варіант "C" рекомендованого стандарту (RS -Recommended Standard) номер 232. Інтерфейс призначений для підключення апаратури, яка передає 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.13 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.13.б).

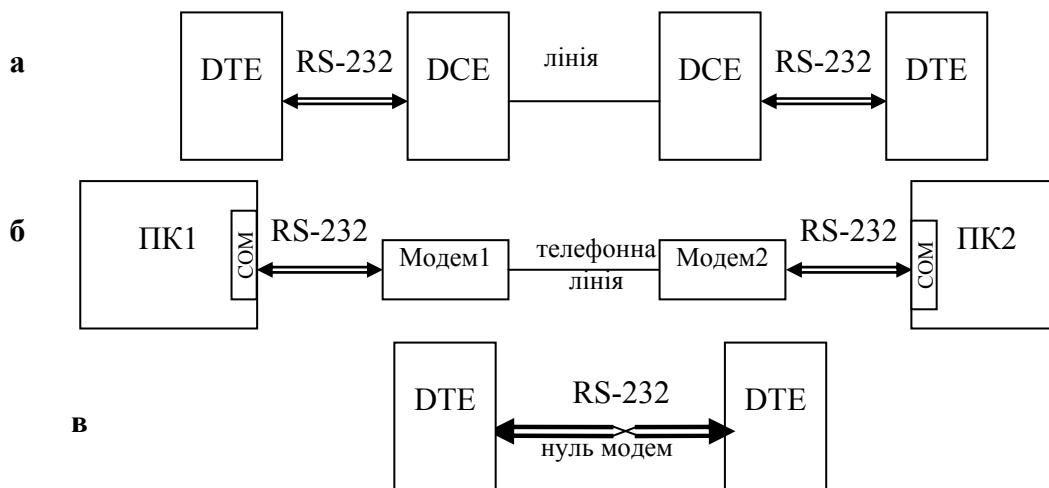


Рис.13. З'єднання за допомогою RS-232: а – схема з'єднання DTE з використанням DCE; б – приклад з'єднання двох ПК (як DTE) через модем (як DCE); в – схема з'єднання двох DTE без DCE (нуль-модемний зв'язок)

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.

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). Логічній "1" на вході відповідає діапазон напруги від -12В до -3В; логічному "0" – від +3В до +12В. Діапазон від -3В до +3В – зона нечутливості, яка обумовлює гістерезис приймача: логічний стан лінії поміняється тільки після переходу через поріг. Рівні сигналів "1" і "0" на виходах передавачів повинні лежати в межах відповідно від -5В до -12В і від +5В до +12В. 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 з швидкістю до 115200 біт/с на відстані до 20 м.

Типи роз'ємів. На обладнанні DTE прийнято встановлювати вижки 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 пристрої DTE (наприклад ПК) та DCE (наприклад модем)

можна підключати безпосередньо один до одного, або через "прямий" кабель з вилкою з одного боку та розеткою з іншого (рис.14).

У системах автоматизації частіше використовується нуль-модемне з'єднання, для якого використовується нуль-модемні кабелі. Якщо при з'єднанні використовуються тільки інформаційні сигнальні лінії (TxD, RxD, SG), то з'єднання проводиться мінімальним нуль-модемним кабелем (рис.15.б), в протилежному випадку – повним (рис.15.а).

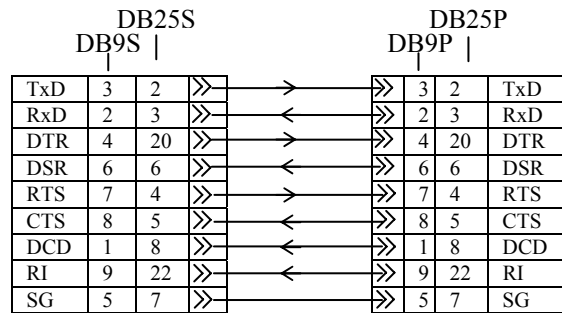


Рис. 14. Підключення DTE (зліва) до DCE (справа)

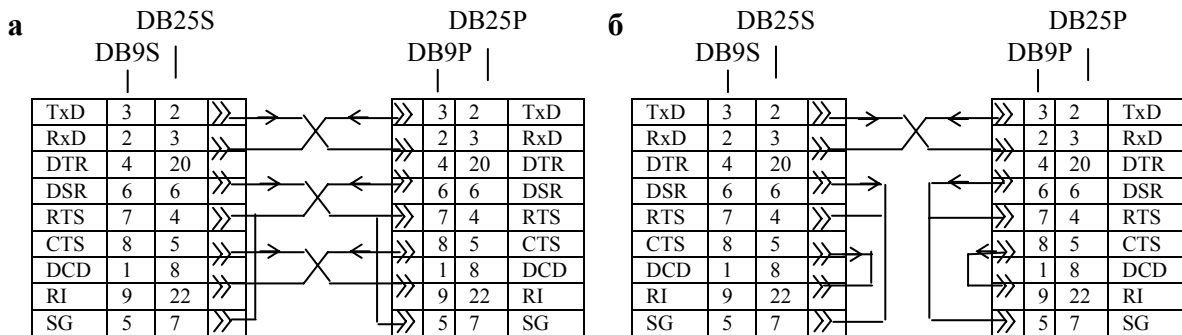


Рис.15. Нуль-модемне підключення, за допомогою нуль-модемного кабелю: а – повного, б – мінімального

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. 4 наведені сигнальні лінії RS-232C та їх призначення. В другій та третій колонці вказані відповідні контакти роз'ємів, а в четвертій колонці також вказана додаткова інформація, яка вказує напрямок передачі даних по лінії: IN – в комп'ютер, OUT – з комп'ютера.

Таблиця 4

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 – перемикає модем у режим передачі.
CTS	5	8	IN Готовність передачі (Clear To Send). "Відкл" - модем забороняє

			комп'ютеру передавати йому дані.
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)

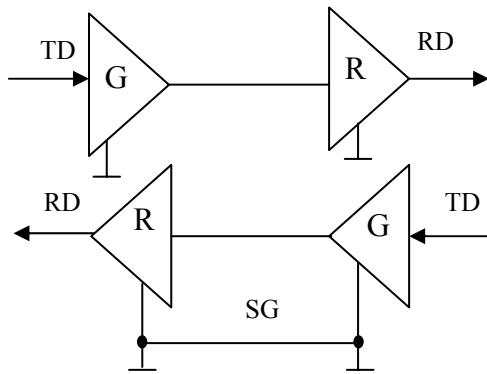


Рис. 16. Функціональна схема нуль-модемного з'єднання

Функціонально нуль-модемне з'єднання двох трансиверів можна показати у вигляді, представленому на рис.16. 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.

Апаратно один передавач даного інтерфейсу може забезпечити необхідним рівнем сигналу тільки одного приймача. Таким чином інтерфейс RS-232 дозволяє

об'єднати між собою тільки два пристрої з одним передавачем і одним приймачем на максимальній відстані 20 м.

4.3.Інтерфейс RS-422

Інтерфейс RS-232 має ряд недоліків, які обмежують область його використання в промислових умовах. Насамперед це пов'язано з чутливістю його 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 з 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, тип та довжину кабелів та ін. Інтерфейс базується на збалансованих (симетричних) лініях передавачів та приймачів, тобто напруга передається по трьом проводам - два сигнальних та сигнальна земля.

На рис.17 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.17 показаний знак інверсії). При передачі логічного „0” передавач інвертує знак сигналу. Приймач вимірює напругу між „А” і „В” і якщо різниця буде більше ніж +200 мВ то приймач буде розпізнавати його як певний логічний стан лінії. Якщо різниця змінить свій знак і стане менше ніж -200 мВ, приймач змінить логічний стан на протилежний.

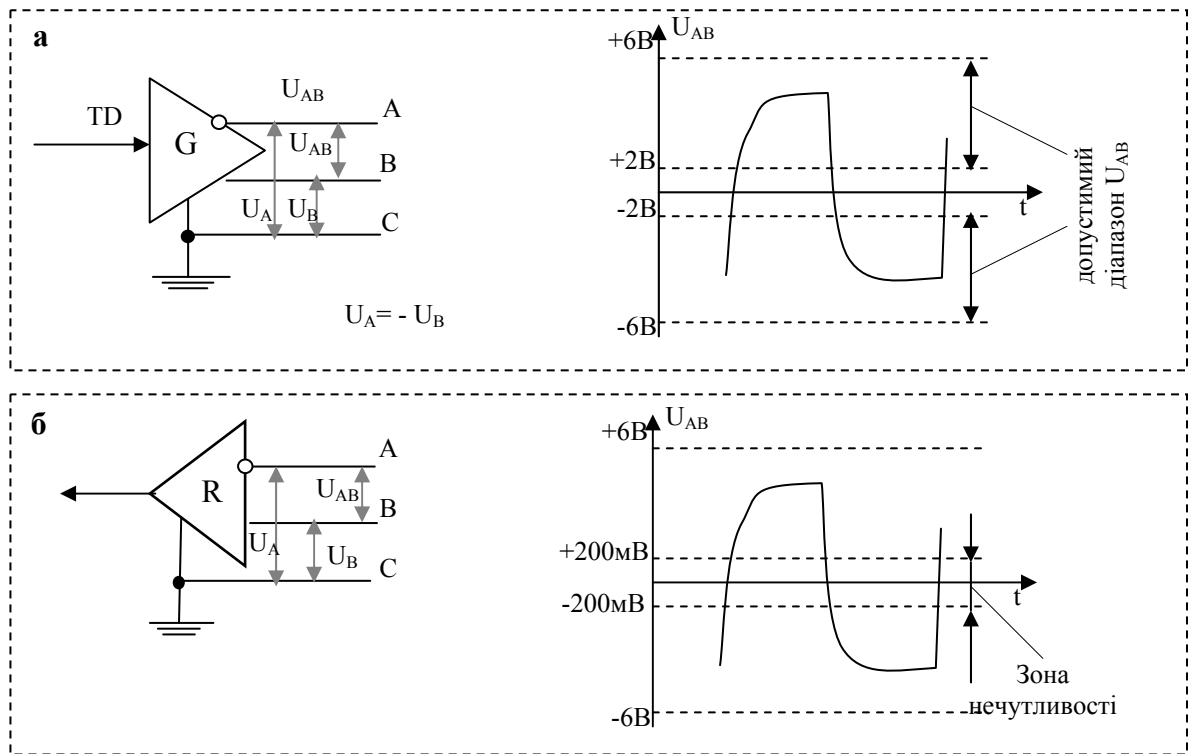


Рис. 17. Функціональна схема передавача (а) та приймача (б) в інтерфейсі RS-422A.

Якщо на проводі „A” передавача в інтерфейсі RS-422A по відношенню до проводу „B” від’ємна напруга, лінія перебуває в стані логічної „1”, якщо додатна – логічного „0”. Позначення проводу „A” як правило еквівалентне „–”, а позначення „B” - „+” (інколи приймають навпаки). RS-422A витримує синфазну напругу $\pm 7V$ (середня напруга на виходах A і B 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.

На основі даного стандарту з'явилися деякі рекомендації по його використанню. Це схеми підключення, способи заземлення, максимальні відстані, використання резисторів-термінаторів, максимальні бітові швидкості і т.д. Так установлено, що максимальна бітова швидкість – становить біля 10 Мбіт/с, максимальна відстань (при невеликих швидкостях) – біля 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.18 показаний один із способів реалізації дуплексного з'єднання двох пристроїв по RS-422A.

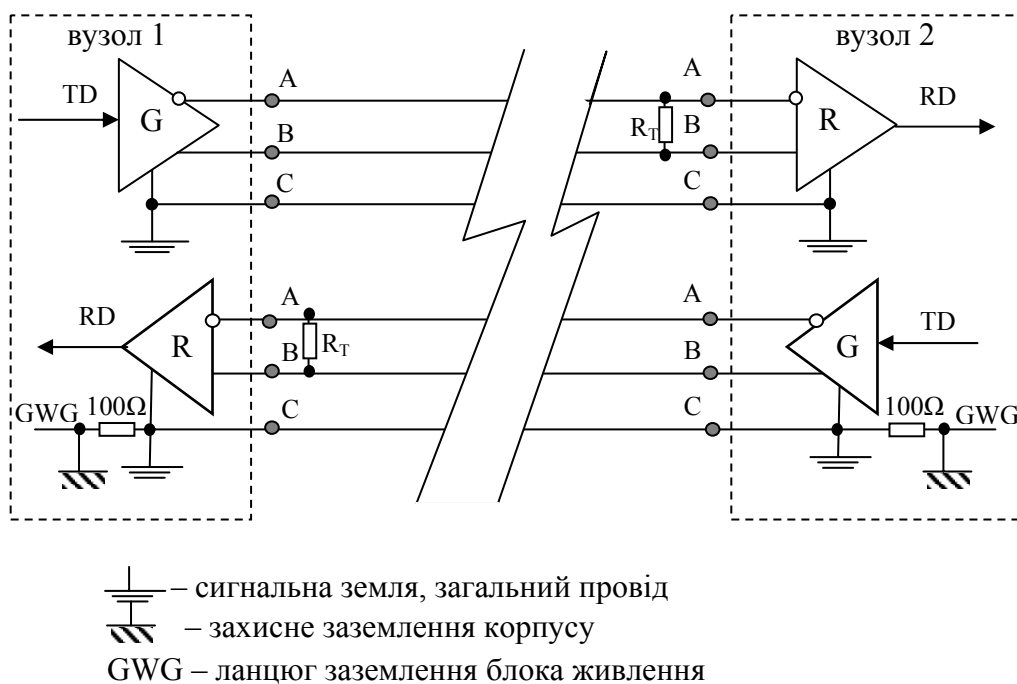


Рис. 18. Функціональна схема 5 – провідного дуплексного з'єднання двох вузлів по RS-422A

Інтерфейс RS-422A має один значний недолік – на ньому не можна побудувати мережі з 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 по напрузі. Наприклад в символному режимі, при відсутності передачі на лінії приймача повинна бути присутня 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 – це може привести до виходу з ладу трансиверів, оскільки кожен з передавачів буде генерувати свої сигнали. Цей недолік відсутній у інтерфейса RS-485.

4.4.Інтерфейс RS-485

Даний стандарт був сумісно розроблений двома асоціаціями 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. На сьогоднішній день 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.19). Тепер, при закінченні передачі пристрій може відключити свій передавач (перевести його в 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 знаходитись в пасивному стані. В який час і в якому порядку передавачі вузлів будуть доступатися до шини стандарт не оговорює, дане питання повинне вирішуватись на каналному рівні (організація доступу до шини). Реалізація інтерфейсу повинна забезпечити працездатність обладнання при короткочасних колізіях (коли два передавача 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 приймача з опором 12 кОм, які підключені до загальної шини паралельно. Якщо приймачі мають більш високий опір, наприклад 48 кОм ($4 * 12$ кОм), то до однієї шини можна підключити відразу 128 таких приймачів, при збільшенні опору до 96 кОм – 256 приймачів і т.д. Слід розуміти, що збільшення імпедансу приймача робить систему чутливою до завад.

Схеми з'єднання. На рис.19 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 підключається 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, всі 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.19 показана 3-х провідна схема, однак на практиці нерідко використовують 2-х провідну (без сигнальної землі).

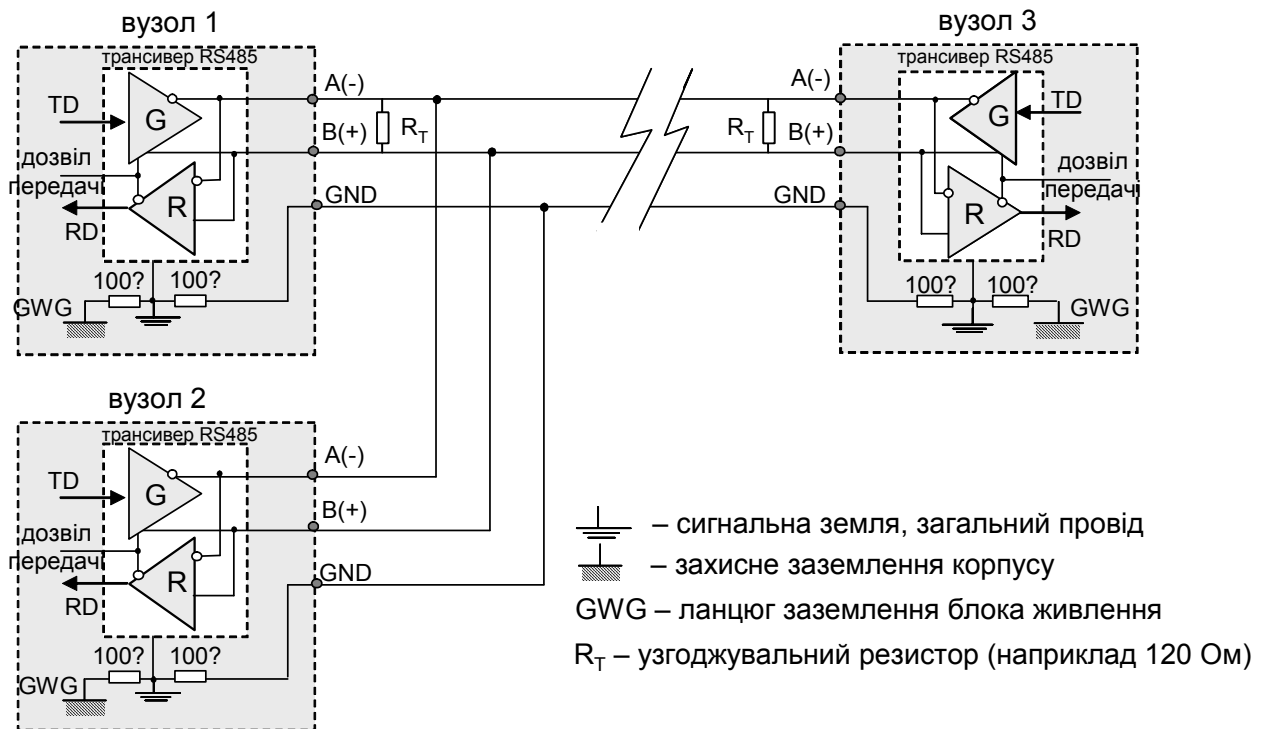


Рис. 19. Функціональна схема трьохпровідного пів-дуплексного з'єднання вузлів по RS-485

Для реалізації 2-х провідної схеми з'єднання достатньо однієї витой пари, трьох-провідної – подвійної витой пари (пара сигнальних проводів, пара земляних).

Бітова швидкість вибирається в залежності від сумарної довжини лінії, характеристик кабелю, і, як правило, приводиться в документації до обладнання у вигляді номограм.

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.19 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 представляють активний опір, тому розсіюють певну потужність, що вимагає використання більш потужних джерел струму. Це спонукає до використання альтернативних

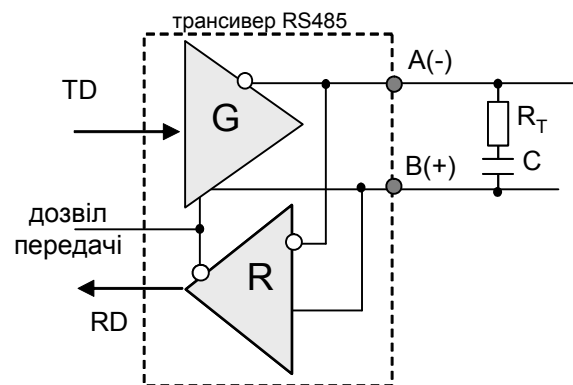


Рис. 20. Схема RC узгодження

схем узгодження. Додавання конденсатора послідовно з резистором виключає протікання постійного струму, і не створює бар'єру для височастотного сигналу (рис.20).

Забезпечення захисного зміщення Реалізація інтерфейсу передбачає, що при 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 буде генерувати сигнал, на шині буде 0 В. Для більшості мереж, зокрема для тих, які працюють в символному режимі, відсутність передачі обов'язково позначається певним рівнем сигналу, наприклад логічною "1". Однак при напрузі на входах приймачів від -200мВ до 200мВ, вихідний стан залишається невизначеним. У цьому випадку логічна „1” або „0” на виході RD приймачів рівноймовірні (або запам'ятовують попередній стан).

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. Задача захисного зміщення зводиться до забезпечення цього ланцюгу таким струмом, щоб напруга на ній була в межах, що відповідає логічній "1", тобто $>200\text{мВ}$. У більшості випадків 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.21). Значення опору резистора зміщення залежить від виду узгодження (термінування) і кількості вузлів тобто приймачів, підключених до шини. Номінали опорів зміщення розраховують виходячи з необхідного мінімального струму зміщення в мережі для забезпечення різниці потенціалів між А і В $>200\text{мВ}$.

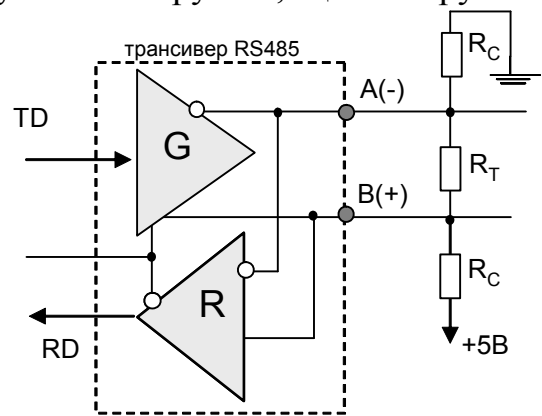


Рис. 21. Використання резисторів захисного (R_C) зміщення разом з узгоджувальними резисторами (R_T)

4.5. Порівняльна характеристика стандартних послідовних інтерфейсів

Для порівняння інтерфейсів, розглянутих вище, найбільш важливі характеристики зведені в таблицю 5.

Порівняльні характеристики послідовних інтерфейсів.

Назва, стандарт	EIA RS-232C	EIA RS-422A	EIA/TIA-485, RS-485	20 мА „струмова петля”, CL, ИРПС,
Тип сигналу	Напруга, асиметричний	Напруга, диференційний симетричний	Напруга, диференційний симетричний	струм
Режим з'єднання	дуплекс	дуплекс	напівдуплекс, дуплекс	дуплекс
Максимальна кількість передавачі/приймачів	1/1	1/10	32/32 (при опорі приймачів 12кОм)	1/1 у багаточкових 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 по 5-провідній дуплексній схемі. Вона передбачає з'єднання одного передавача (з правами Ведучого) з іншими приймачами (на вузлах з правами Ведених) по одній парі проводів. По другій парі 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 має місце тільки в мережах типу Ведучий/Ведені і не може бути застосована в ширококомовних мережах. При використанні RS-485 кількість приймачів вказана рівною 32, це при опорі приймачів 12кОм. При інших опорах, наприклад 48кОм, 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. В більшості випадків 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 по інтерфейсу "струмова петля". Якщо необхідне з'єднання точка-точка – вибирають дуплексний зв'язок, оскільки, при тих самих бітових швидкостях досягається майже в 2 рази більша швидкість передачі даних, за рахунок подвійного каналу. У цьому випадку можна використати RS-422A або RS-485 в 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 області промислових мереж, інші тільки почали активно впроваджуватися. Один з найбільш перспективних інтерфейсів

являється **USB**, який практично витіснив COM-порт (RS-232) та паралельний порт з персональних комп'ютерів.

4.6. Використання адаптерів-перетворювачів інтерфейсів

Типи адаптерів-перетворювачів. Описані вище послідовні інтерфейси складають далеко не повний перелік інтерфейсів фізичного рівня. На фізичному рівні нерідко використовуються унікальні інтерфейси. Однак навіть для мереж, які базуються на стандартних послідовних інтерфейсах, нерідко виникає задача в об'єднанні пристроїв з різними реалізаціями фізичного рівня. Для вирішення такої задачі існують **адаптери-перетворювачі (конвертери) інтерфейсів**. Адаптер можна представити у вигляді пристрою, що має два різні інтерфейси. Задача даного пристрою електрично перетворити вхідні сигнали RD одного інтерфейсу у вихідні TD іншого. Можливі випадки з'єднань пристроїв з різними інтерфейсами: RS-422A <-> RS-485, RS-232 <-> Current Loop 20 mA, RS-232 <-> RS-485.

Зупинимося на принципах роботи адаптерів RS-232C <-> RS-485 більш детально, оскільки вони одні з найбільш поширених в 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 при 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 в конвертері повинно бути забезпечене переключення передавача RS-485 зі сторони RS-232 в активний чи пасивний (високоімпедансний) стан. Виробники адаптерів вирішують цю проблему по різному. Розглянемо декілька способів.

Управління адаптером RS-232<->RS-485 зі сторони RS-232. Один із варіантів вирішення проблеми управління станом трансмітера RS-485 є використання одного з сигналів управління потоком зі сторони інтерфейсу RS-232C: RTS або DTR. На рис.22 показана модель перетворювача, який управляється сигналом RTS. На рисунку перетворювач умовно поділений на 2 частини з відповідними інтерфейсами. Коли пристрій з RS-232C має передавати дані, він активує передавач адаптера за допомогою RTS, виставляючи на цій лінії логічну „1”. Далі він надсилає послідовність бітів в символному вигляді по лінії TD. Перетворювач за допомогою схемних рішень відсилає цей сигнал на передавач адаптера зі сторони RS-485. Оскільки той вже активований, цю ж послідовність він видає на лінію RS-485 інтерфейсу. Так буде тривати, поки пристрій зі сторони RS-232C не зніме „1” з RTS, що приведе до переключення передавача в високоімпедансний (неактивний) стан.

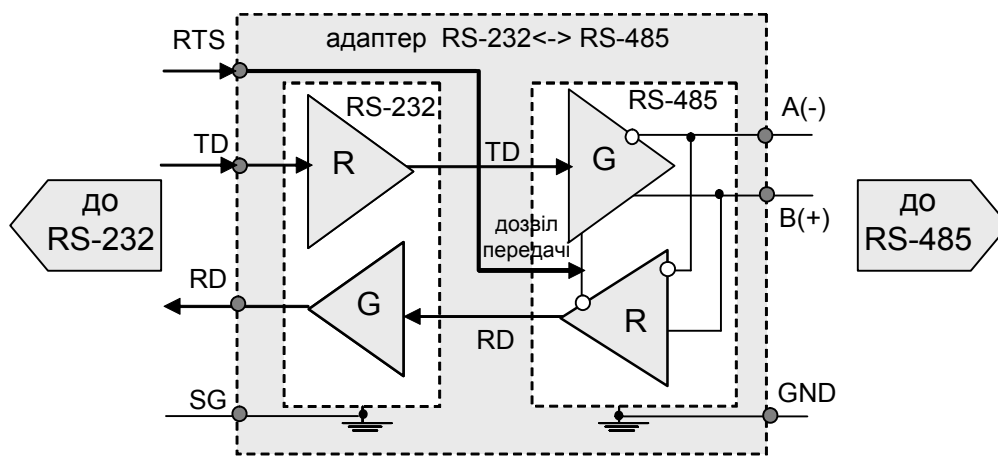


Рис. 22. Управління передавачем адаптера RS-232<->RS-485 сигналом RTS

Алгоритм управління сигналом RTS для RS-232C може бути різним. Для комп'ю-терів, як правило, налаштовують роботу COM-порту так, щоб сигнал RTS виставлявся автоматично в „1”, коли буфер передачі заповнений байтами, і „0”, коли він пустий. 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 біту. Такі проблеми повинні вирішуватись на пристрої зі сторони інтерфейсу RS-232C. Ще одна з проблем заключається в тому, що не всі пристрої вміють управляти сигналом RTS. У цьому випадку існує інший, більш універсальний спосіб управління передавачем адаптера.

Автоматичне управління адаптером RS-232<->RS-485. Цей метод у різних фірм називається по різному: Automatic Send Data Control (ASDC), Automatic Data Direction Control (ADDC), Automatic RTS Signal Control (ARSC), але всі вони базуються на принципі управління по сигналу даних, які передаються (рис.23). Тобто адаптер сам відслідковує, коли включати і коли відключати передавач по даним, які надходять зі сторони RS-232. Один з варіантів такої реалізації – відслідковування приймального буферу. Як тільки адаптер з боку RS-232 отримав перший біт символу (стартовий біт) – відразу активує передавач. Оскільки біт в 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 в високоімпедансний стан. Якщо затягнути цю паузу, може виникнути ситуація, коли інший передавач в системі зі сторони RS-485 захоче передати дані (адже лінія вже звільнилася) і виникне колізія.

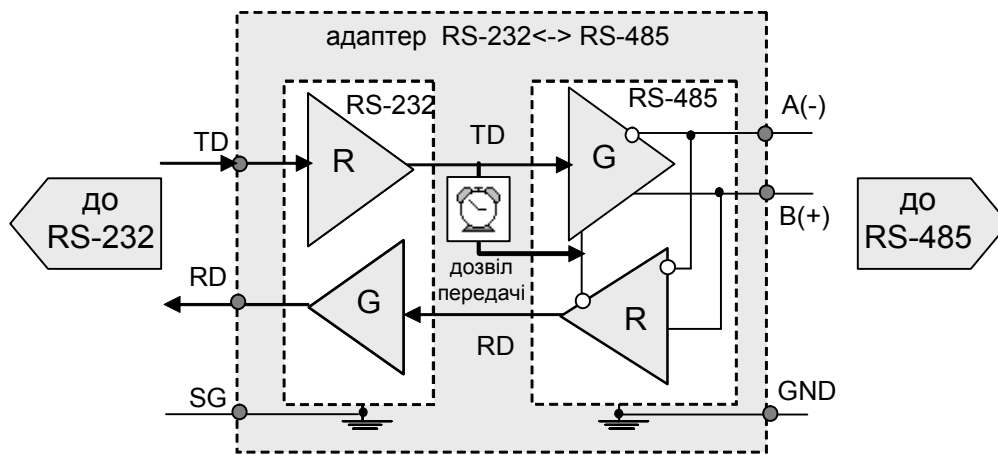


Рис. 23. Управління передавачем адаптера за сигналом даних, що передаються.

Адаптери з управлінням по сигналу даних по приймальному буферу користуються більшою популярністю ніж адаптери з управлінням сигналами RTS чи DTR.

Ще один з методів управління передавачем, який не знайшов широкого використання, по сигналу зі сторони 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 в стан прийому, тобто вимикає передавач. Специфіка таких перетворювачів заключається в тому, що використати їх можна тільки в режимі точка-точка, що сильно звужує коло їх використання. Такий перетворювач не накладає ніяких вимог по управлінню на пристрій з RS-232C, що звісно ж таки повинно компенсуватись зі сторони RS-485.

Література: [1], розділ 3 (3.7-3.14).

5. Мережі MODBUS.

- 5.1. Мережі MODBUS в контексті моделі OSI.
- 5.2. Реалізація MODBUS на прикладному рівні
- 5.3. Формат основних функцій
- 5.4. MODBUS Serial
- 5.5. MODBUS TCP/IP

5.1. Мережі MODBUS в контексті моделі OSI

MODBUS розроблений в 1979 р. фірмою Modicon Gould (зараз в складі Schneider Electric). Це один із найстаріших протоколів промислової мережі, який на сьогоднішній день є одним з найбільш популярних. Основна причина такої популярності – це простота в реалізації. Даний розділ присвячений мережам, які базуються на протоколі прикладного рівня MODBUS.

На сьогоднішній день MODBUS підтримує і розвиває організація **MODBUS-IDA**, яка являє собою групу незалежних споживачів та поставників пристроїв автоматизації. Вона забезпечує відкритість даного протоколу та розробляє готові

компоненти для спрощення реалізації. Будемо розглядати MODBUS таким, яким він існує на сьогоднішній день в 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, є доцільним розглядати його в контексті моделі OSI а не МЕК.

Згідно стандартів MODBUS-IDA – MODBUS являється протоколом прикладного рівня для зв'язку типу Клієнт-Сервер між прикладними Процесами пристроїв, які під'єднані до різноманітних типів шин або мереж. В контексті OSI-моделі, ці мережі мають архітектуру, наведену на рис.24.

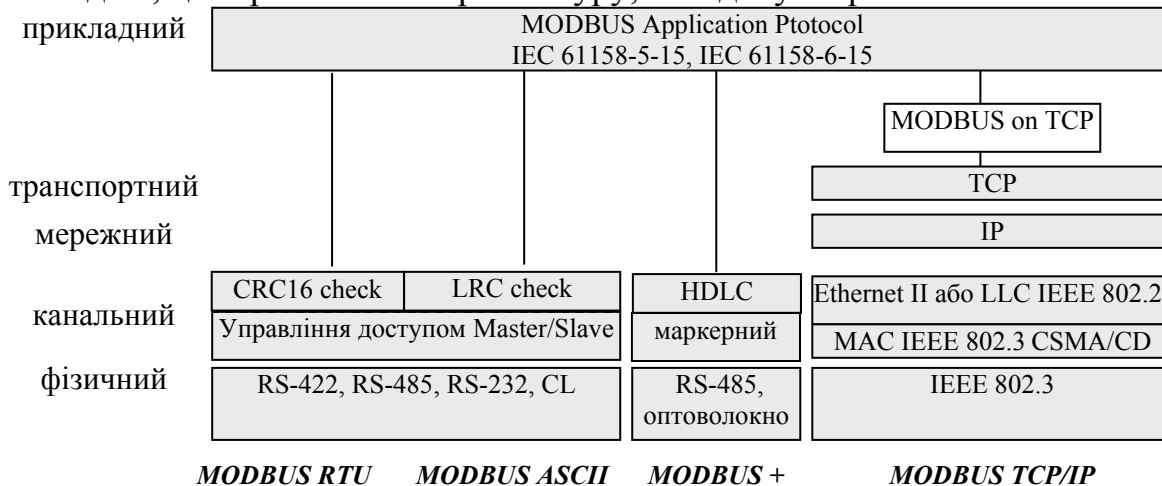


Рис.24. MODBUS в контексті OSI-моделі

Як видно з рисунка, MODBUS на сьогоднішній день 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.

5.2. Реалізація MODBUS на прикладному рівні

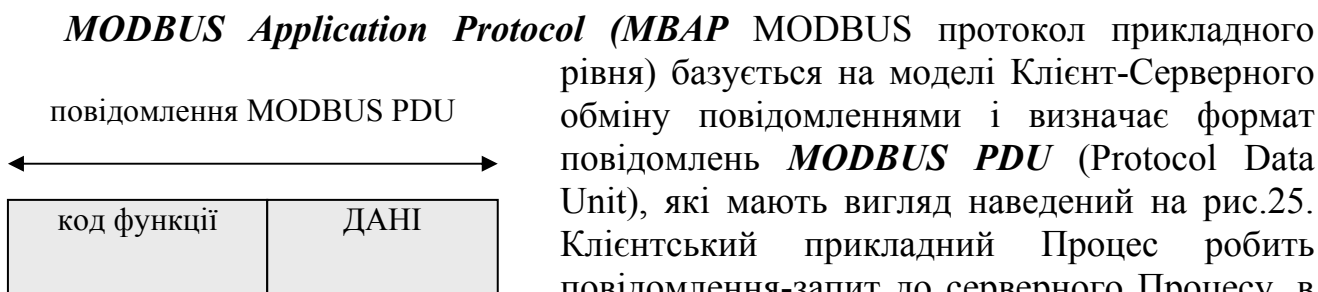


Рис.25. Формат повідомлення MODBUS PDU.

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 байту, яке може приймати значення від 1 до 255 (коди 128-255 зарезервовані під коди повідомлень-відповідей при помилкових діях). 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.

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.

5.3. Формат основних функцій

Повний список кодів а також специфікацію протоколу можна знайти на офіційному Веб сайті MODBUS-IDA - www.MOVBUS.org. Тут детально розглянемо тільки найбільш вживані функції MODBUS для обміну даними процесу. Номер функції дається в шістнадцятковому форматі. Скорочення в дужках *Hi* та *Lo* що вказують відповідно на старший та молодший байти. Тобто, якщо для вказівки адреси початкової змінної необхідно двобайтове слово, то значення старшого байта буде передаватись в полі з позначенням *Hi*, а молодшого – відповідно *Lo*.

Код функції 01_{16} – читання статусу Coils (дискретних вихідних бітів). Повідомлення-запит вміщує адресу початкового біту 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 не кратно восьми, інші біти в байті заповнюються нулями. Лічильник вміщує кількість байт в полі даних.

Запит:

Код функції	01
Адреса початкового біту (HI)	0 до $FFFF_{16}$
Адреса початкового біту (LO)	
Кількість біт (HI)	1 до $7D0_{16}$ (2000)
Кількість біт (LO)	

Відповідь:

код функції	01
лічильник байт	N
Значення бітів (перші 8 біт)	0 до FF_{16}
Значення бітів (наступні 8 біт)	0 до FF_{16}
...	
Значення бітів (N-ні 8 біт)	0 до FF_{16}

Код функції 02_{16} – читання статусу дискретних входів. Формат даного запиту такий же як попереднього, за винятком поля функції.

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
Адреса початкового регістру (Lo)	FFFF ₁₆
Кількість регістрів (Hi)	від 1 до
Кількість регістрів (Lo)	7D ₁₆ (125)

Відповідь:

код функції	01
лічильник байт	N*2
Значення 1-го регістру (Hi)	0 до
Значення 1-го регістру (Lo)	FFFF ₁₆
...	
Значення N-го регістру (Hi)	0 до
Значення N-го регістру (Lo)	FFFF ₁₆

У відповідному повідомленні в полі даних кожний регістр передається двома байтами.

Код функції 04₁₆ – читання значення вхідних регістрів. Формат даного запиту такий же як попереднього, за винятком поля функції.

Код функції 05₁₆ – запис вихідного/внутрішнього біту. В запиті вказується номер бітової змінної та значення: 0 – 0000, а 1 – FF00, всі інші значення не міняють стан змінних. В ширококомовній передачі клієнтський запит виставляє значення даної змінної для всіх серверів.

Функція	05
Адреса біту (Hi)	від 0 до FFFF ₁₆
Адреса біту (Lo)	
Значення біту (Hi)	0000 або FF00 ₁₆
Значення біту (Lo)	

Нормальна відповідь серверу являється повторенням запиту до клієнта.

Код функції 06₁₆ – запис вихідного/внутрішнього регістру. Функція аналогічна попередній, але оперує з регістрами(словами). В запиті вказується номер вихідного/внутрішнього регістру та його значення. В ширококомовній передачі запит виставляє значення даної змінної для всіх серверів.

Функція	06
Адреса регістру (Hi)	від 0 до FFFF ₁₆
Адреса регістру (Lo)	
Значення регістру (Hi)	від 0 до FFFF ₁₆
Значення регістру (Lo)	

Нормальна відповідь сервера являється повторенням запиту клієнту.

Код функції 0F₁₆ – запис декількох вихідних/внутрішніх бітів. В запиті вказується початкова адреса біту, кількість біт для запису, лічильник байтів і безпосередньо значення. В ширококомовній передачі біти записуються всім серверам. Розглянемо приклад для встановлення наступних 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

В таблиці показана відповідність адреси змінної, починаючи з 19-ї, і значення біту. 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 вказані у кадрі. Запит та відповідь будуть мати такий вигляд:

Запит:

Функція	0F
Адреса початкового біту (Hi)	00
Адреса початкового біту (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 біту (Hi)	00
Адреса початкового біту (Lo)	13
Кількість бітів (Hi)	00
Кількість бітів (Lo)	0A

Код функції 10₁₆ – запис декількох вихідних/внутрішніх регістрів.

Запит:

Функція	10 ₁₆
Адреса початкового регістру (Hi)	0 до
Адреса початкового регістру (Lo)	FFFF ₁₆
Кількість регістрів (Hi)	1 до
Кількість регістрів (Lo)	007B ₁₆ (123)
Лічильник байтів	2*N
Дані (1-й регістр Hi)	0 до
Дані (1-й регістр Lo)	FFFF ₁₆
...	
Дані (N-й регістр Hi)	0 до
Дані (N-й регістр Lo)	FFFF ₁₆

Відповідь:

Функція	10
Адреса початкового регістру (Hi)	0 до FFFF ₁₆
Адреса початкового регістру Lo	
Кількість регістрів (Hi)	1 до 007B ₁₆
Кількість регістрів (Lo)	(123)

Приклад MODBUS. Запит на читання значення вихідних/внутрішніх регістрів.

Завдання. Сформувані повідомлення-запит та повідомлення-відповідь на читання вихідних/внутрішніх регістрів починаючи з 108-го по 110-й при позитивній обробці запиту Сервером.

Рішення. Формат повідомлень показаний на рис.26. Як і в попередньому випадку 108-й регістр в запиті вказується під номером 107 (6B₁₆).



Рис.26. Формат повідомлень для запиту читання значень вихідних/внутрішніх регістрів

5.4. 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 з символьним режимом передачі (див. рис.24).

На сьогоднішній день 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

рівні. Оскільки мережа MODBUS RTU/ASCII може мати шинну топологію, то визначений метод доступу до шини – це модель Ведучий/Ведений. В мережах MODBUS RTU та MODBUS ASCII Процес Ведучого завжди являється 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 з 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.

Канальний рівень. На рис.27 показаний загальний вигляд кадру MODBUS Serial.



Зверніть увагу, що розмежування між кадрами та тип контрольної суми тут не вказано, 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.

MODBUS RTU Даний режим передбачає використання 8 біт даних в 11-бітному символі, що дозволяє передавати по байту на символ. Формат символу в RTU режимі: 1 стартовий біт; 8 біт даних (молодший біт передається першим); 1 біт паритету + 1 стоповий біт або без паритету + 2 стопових біта.

Формат кадру MODBUS RTU наведений на рисунку 28. 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 $3.5 \cdot T_c$ приймач ідентифікує помилку. З іншого боку, появлення нового кадру раніше ніж $3.5 \cdot T_c$, теж приведе до помилки.

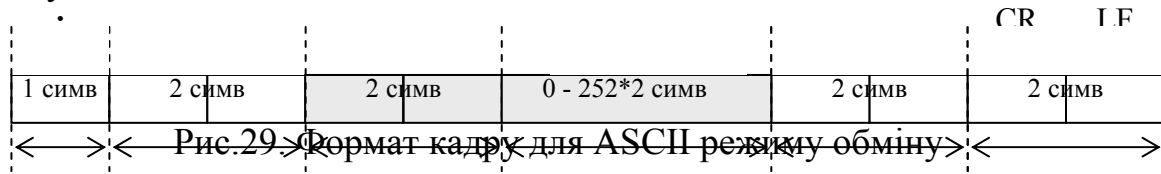


Рис.28. Формат кадру для RTU режиму обміну

Поле адреси і коду функції в RTU режимі займають по одному байту, оскільки байти передаються по символу на кожний. У якості контрольної суми використовується два байти, обраховані по алгоритму CRC16.

MODBUS ASCII У даному режимі кожний байт повідомлення передається як два ASCII символи їх шістнадцяткового представлення, тобто значення байта 03_{16} буде передаватись як ASCII-код символів „0” і „3” (0110000 0110011) Отже байти даних, код функції і байт поля перевірки буде передаватись 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.29. Як бачимо, для розмежування між кадрами використовуються стартовий символ „:” та стопова послідовність „CR LF”. Приймачі на шині безперервно відслідковують символ „:” який однозначно вказує на початок кадру. Коли він прийнятий, приймачі відловлюють поле адреси і т.д. Це дуже простий спосіб синхронізації, який дозволяє некритично відноситись до пауз між символами (до 1 сек.). Адреса Веденого та код функції займають по два символи, відповідно до значення одного байта. Далі йдуть $n*2$ 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.



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 попередній.

У ряді випадків реалізація функцій MODBUS-Клієнта лягає на операційну систему, а доступ до них в програмі ПЛК відбувається через інтерфейсні комунікаційні функції. Зокрема це характерно для більшості ПЛК від Schneider Electric (Momentum, Quantum, TSX Micro, TSX Premium, M340). В ряді інших систем – клієнтську сторону на прикладному рівні необхідно повністю прописувати в програмі ПЛК, а інтерфейс надається тільки для обміну з комунікаційним портом. В цьому випадку система надає сервіси відправки та отримання повідомлення (яке формує і аналізує сама програма користувача), і генерації та перевірки контрольної суми.

5.5. MODBUS TCP/IP

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 в специфікаціях MODBUS-IDA, в яких комунікаційна система MODBUS TCP/IP може включати різні типи пристроїв (рис.30): 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 MODBUS Serial Клієнтськими і Серверними пристроями.

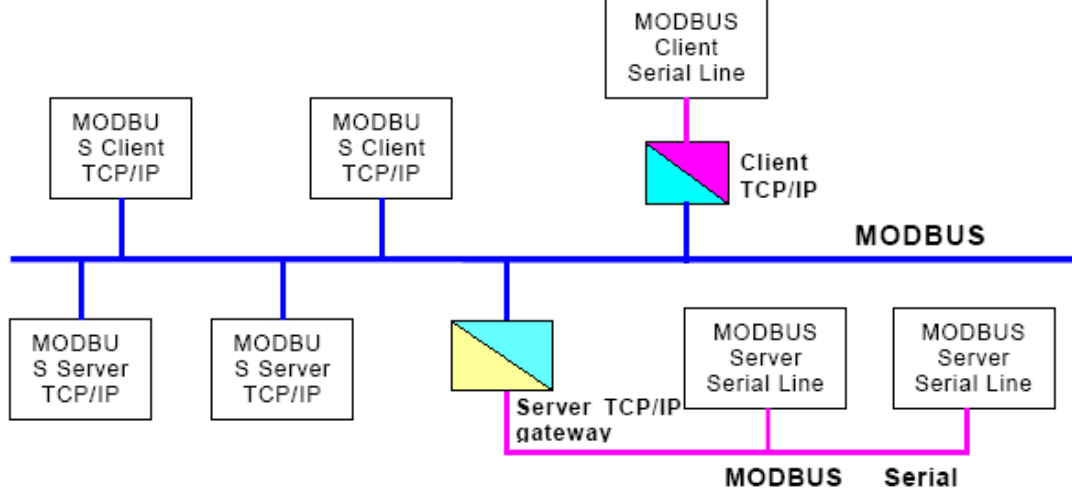


Рис.30. Комунікаційна архітектура MODBUS TCP/IP

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 привести рис.31, взятий зі специфікації протоколу:

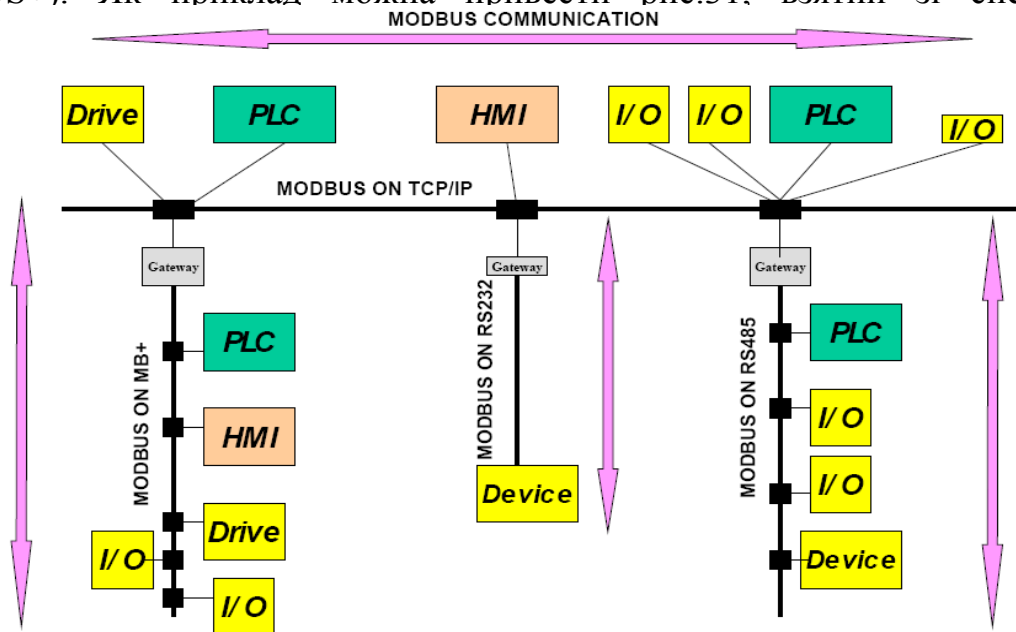


Рис.31. Приклад архітектури мережі MODBUS

Особливості реалізації протоколу Аналогічно всім мережам MODBUS, дана мережа використовує MODBUS Application Protocol. Нагадаємо, що в MODBUS Serial на каналному рівні до PDU додається адреса веденого і контрольна сума, сам PDU не модифікується. Однак в MODBUS TCP/IP перед попаданням на транспортний рівень, до PDU (код функції та дані) додається додатковий МВАР-заголовок. (рис.32). Заголовок складається з полів, які описані в табл.6.6. Отриманий модуль передається рівню TCP.

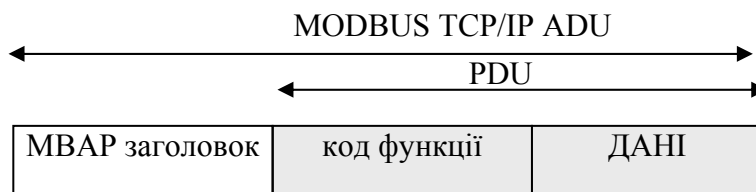


Рис.32. Формат модуля даних прикладного рівня (APDU) MODBUS TCP/IP

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 замінивши поле ProtocolID Ethernet модулі ПЛК Schneider Electric одночасно підтримують два протоколи: MODBUS та UNI-TE.

Таблиця 6

Поля МВАР заголовка

Поле	Довжина (байт)	Пояснення	Клієнт	Сервер
TransactionID	2	ідентифікація транзакцій запитів/ відповідей	ініціалізує Клієнт в запиті	копіює з запиту у повідомлення - відповідь
ProtocolID	2	тип протоколу, 0=MODBUS протокол	ініціалізує Клієнт в запиті	копіює з запиту у повідомлення - відповідь
Length	2	кількість наступних байтів	ініціалізує Клієнт в запиті	ініціалізує Сервер у відповіді
UnitID	1	адреса Веденого, який підключений до вузла	ініціалізує Клієнт в запиті	копіює з 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 стеку TCP/IP а також Ethernet можна ознайомитись в літературі по комп'ютерним мережам а також в розділі 10 навчального посібника. Крім того

там наведені деякі пояснення з приводу обмежень використання рішень подібних MODBUS TCP/IP, зокрема в системах з жорсткими вимогами до реального часу.

Література: [1], розділ 6.

Список літератури.

1. Пупена О.М., Ельперін І.В., Луцька Н.М., Ладанюк А.П. Промислові мережі та інтеграційні технології в автоматизованих системах: Навчальний посібник. – К.: Вид-во "Ліра-К", 2011. – 552 с.