МІНІСТРЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Кафедра електроніки і комп'ютерної техніки

«До захисту допущено»
Завідувач кафедри
_____ ОПАНАСЮК
(підпис)
Анатолій 2023р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавра

зі спеціальності 172 «Телекомунікації та радіотехніка»,
освітньо-професійної програми «Мережеві та інтернет технології»
На тему: пристрій розподілу інформації
на основі послідовних інтерфейсів.

Здобувача групи ТК-91 Растворцева Лева Дмитровича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Лев РАСТВОРЦЕВ
(підпис)

Керівник, заступник завідувача кафедри, кандидат технічних наук, доцент

_____ Ігор КУЛИК
(підпис)

Суми – 2023

ВСТУП

Вітчизняний та зарубіжний досвід переконливо свідчать про те, що розвиток підприємств металургійного, енергетичного і хімічного комплексу, вирішення проблем якості та конкурентоспроможності продукції на світовому ринку вимагають кардинального вдосконалення систем збирання, зберігання, обробки, передачі та використання інформації. Сучасні тенденції розвитку промисловості характеризуються розробкою, впровадженням і широким застосуванням інформаційних систем і технологій. Основою інформаційних технологій і систем є комп'ютери та комп'ютерні мережі з розгалуженим програмним забезпеченням, а також системи управління базами даних, комп'ютерні системи підтримки прийняття рішень, методологічна основа яких ґрунтується на теорії систем і системному аналізі, теорії моделювання даних, технологічних процесів та знань.

Пристрій розподілу інформації на основі послідовних інтерфейсів є компонентом системи, що використовується для передачі даних між пристроями чи системами через послідовний канал зв'язку. Він забезпечує організацію передачі інформації в послідовному форматі, де біти передаються один за одним у послідовному порядку.

Основними складовими пристрою розподілу інформації на основі послідовних інтерфейсів є передавач та приймач. Передавач приймає дані для передачі та конвертує їх у послідовний формат, після чого вони передаються через послідовний канал. Приймач отримує послідовні дані та відновлює їх у вихідний формат.

Такі пристрої розподілу інформації широко використовуються у різних сферах, включаючи телекомунікації, комп'ютерні мережі, автоматизацію та контроль, електроніку, медичні пристрої, автомобільну промисловість та інші. Вони дозволяють ефективно передавати дані в режимі реального часу і забезпечують надійність передачі навіть на великі відстані.

У цій кваліфікаційній роботі будуть досліджені основні аспекти пристроїв розподілу інформації на основі послідовних інтерфейсів, їх принципи роботи, протоколи комунікації та застосування. Також будуть розглянуті переваги та недоліки використання послідовних інтерфейсів порівняно з іншими типами з'єднань, що дозволить краще розуміти їхню роль у сучасних технологіях передачі даних.

1 ОГЛЯД ЛІТЕРАТУРИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Загальна інформація**.**

Інтерфейс RS-485 - широко поширений швидкісний та стійкий до перешкод промисловий послідовний інтерфейс передачі даних. Практично всі сучасні комп'ютери в промисловому виконанні, більшість інтелектуальних датчиків і виконавчих пристроїв, програмовані логічні контролери разом з традиційним інтерфейсом RS-232 містять в своєму складі реалізацію інтерфейсу RS-485. Інтерфейс RS-485 базується на стандарті EIA RS-422/RS-485. Основна відмінність RS-485 від також широко поширеного RS-232 - можливість об'єднання декількох пристроїв. RS-485 підтримує диференційний сигнал, де дані передаються як різниця напруги між двома проводниками (позитивний та негативний). Це дозволяє забезпечити стійкість до шумів та впливу електромагнітних полів. Крім того, RS-485 має дуже низьку споживану потужність, що робить його енергоефективним для довготривалих додатків з живленням від батарей або акумуляторів. RS-485 має свої стандарти та специфікації, які регулюють передачу даних, електричні характеристики, протоколи комунікації та інші параметри.

RS-485 широко використовується у багатьох галузях, включаючи автоматизацію, промисловість, будівництво, системи безпеки, контроль доступу, системи вимірювання та багато інших. Інтерфейс RS-485 забезпечує обмін даними між кількома пристроями по одній двожильній лінії зв'язку у напівдуплексному режимі. Він широко використовується в промисловості для створення АСУ ТП. Інтерфейс RS-485 найчастіше використовується при створенні сучасних локальних мереж різного призначення, як у промислових виробах, так і в аматорській практиці. Основними перевагами цього інтерфейсу є:

відносно низька вартість мікросхем драйверів, що знижує вартість апаратної реалізації мережевих диспетчерів, тобто вузлів зв'язку між мережевим середовищем (лініями зв'язку) та ядром станції (вузла) мережі, тобто мікроконтролерною або мікропроцесорною системою;

використання всього трьох проводів (третій, загальний, не завжди є обов'язковим) у мережах на основі інтерфейсу RS-485, що значно знижує вартість всієї системи, оскільки відомо, що вартість мережевого середовища сучасних локальних мереж майже завжди становить більше 60% вартості всієї системи;

мікросхеми драйверів мають невеликі розміри. Найчастіше використовуються мікросхеми в корпусі DIP8 зі стандартним розташуванням виводів, що стало де-факто промисловим стандартом. Мікросхеми драйверів використовують лише

декілька дискретних елементів для захисних схем, використання яких не є обов'язковим. Невеликі розміри мікросхем драйверів і мінімальна кількість підключення дозволяють економити площу печатної плати, що також позитивно впливає на вартість системи;

сучасні мікросхеми мають досить низьке споживання енергії, багато з них автоматично переходять у режим енергозбереження при відсутності активності в мережі, що знижує енергоспоживання системи;

сучасні мікросхеми драйверів мають підвищену навантажувальну здатність. Якщо раніше більшість мікросхем призначалися для роботи з 32 станціями, то сучасні моделі забезпечують нормальне функціонування до 256 станцій;

на сьогоднішній день випускаються мікросхеми з високою максимальною швидкістю передачі. Це дозволяє створювати високошвидкісні мережі і зменшує кількість помилок в мережі за рахунок поліпшення форми передаваного сигналу;

драйвери інтерфейсу RS-485 мають досить просте керування. Особливості організації мереж, їх схемотехніка, методи керування доступом до каналу та приклади програмування досить добре описані;

мікросхеми інтерфейсу RS-485 випускають багато фірм у світі. Однак незаперечним лідером у розробці та випуску нових мікросхем драйверів є відома фірма MAXIM. На сьогоднішній день фірма випускає понад 80 типів мікросхем драйверів інтерфейсу RS-485/422. Мережа, побудована на основі інтерфейсу RS-485, складається з передавачів та приймачів, які з'єднані за допомогою витої пари - двох перехресно скручених проводів. В основі інтерфейсу RS-485 лежить принцип диференційної (балансної) передачі даних. Його суть полягає в передачі одного сигналу по двом проводам. По одному проводу (уявно позначеному як А) передається оригінальний сигнал, а по іншому (уявно позначеному як В) - його інвертована копія. Іншими словами, якщо на одному проводі "1", то на іншому "0" і навпаки. Таким чином, між двома проводами витої пари завжди є різниця потенціалів: при "1" вона позитивна, при "0" - від'ємна. Саме цією різницею потенціалів і передається сигнал. Такий спосіб передачі забезпечує високу стійкість до спільної помилки. Спільною помилкою називають помилку, яка діє на обидва проводи лінії однаково. Наприклад, електромагнітна хвиля, проходячи через ділянку лінії зв'язку, наводить потенціал у обох проводах. Якщо сигнал передається потенціалом на одному проводі відносно загального, як у RS-232, то наведення на цей провід може спотворити сигнал відносно добре збирати наведення загального ("землі"). Крім того, на опорі довгого загального проводу буде падати різниця потенціалів земель - додаткове

джерело спотворень. При диференціальній передачі спотворень не відбувається. У самому ділі, якщо два проводи проходять близько один до одного, ще й перехресно перевиті, то наведення на обидва проводи однакове. Потенціал у обох однаково навантажених проводах змінюється однаково, при цьому інформативна різниця потенціалів залишається без змін.

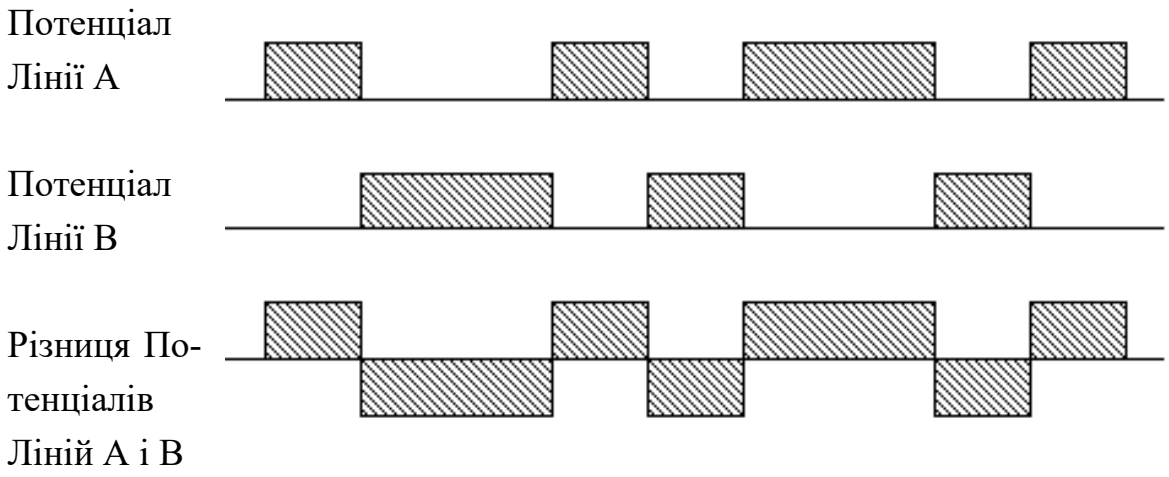

Рисунок. 1.1 – Принцип підключення прийомопередавачів і структура локальної мережі на основі інтерфейсу RS-485.

На рисунку 1. 1 показано, як підключаються приймопередавачі та структура локальної мережі, побудованої на основі інтерфейсу RS-485.

Прийомопередавачі підключаються до двохпровідної лінії RS-485, яка передає дані між пристроями.

Лінія RS-485 може містити до 32 пристроїв, які навантажують лінію на одиницю навантаження (UL). Сумарна кількість навантажень UL на лінії не повинна перевищувати 32.

Така структура мережі на основі інтерфейсу RS-485 дозволяє передавати дані між пристроями на великій відстані, забезпечуючи високу швидкість передачі та стійкість до шумів і перешкод.

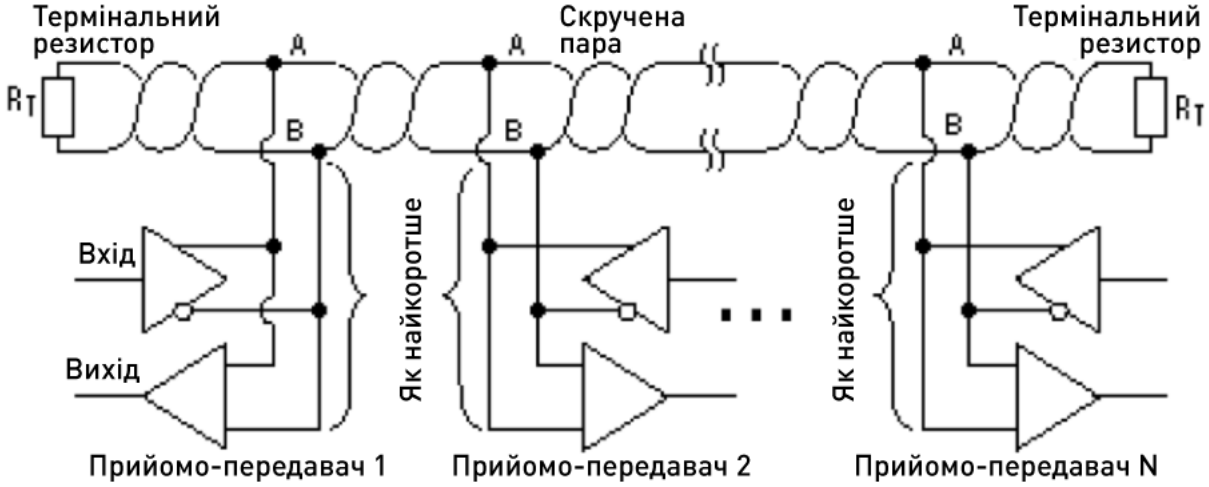

Рисунок 1.2 - Локальна мережа на основі інтерфейсу RS-485

Параметри інтерфейсу RS – 485 наведені в таблиці 1.1.

Таблиця 1.1 – Параметри інтерфейсу RS – 485

Параметр	Значення
Допустима кількість передавачів / приймачів.	32 / 32
Максимальна довжина кабелю.	1200 м
Максимальна швидкість зв’язку	10 Мбіт/с
Діапазон напруги "1" передавача.	+1.5...+6 В
Діапазон напруги "0" передавача.	-1.5...-6 В
Діапазон синфазної напруги передавача.	-1...+3 В
Допустимий діапазон напруги приймача.	-7...+12 В
Пороговий діапазон чутливості приймача.	±200 мВ
Максимальний струм короткого замикання драйвера.	250 мА
Допустимий опір навантаження передавача.	54 Ом
Вхідний опір приймача.	12 кОм
Максимальний час наростання сигналу передавача.	30% біта

Під час підключення слід правильно приєднати сигнальні ланцюги, зазвичай називаємі A і B. Переполярізація не страшна, але пристрій працювати не буде. Мережу слід прокладати за топологією шини, без відгалужень. Пристрої слід підключати до кабелю дротами мінімальної довжини.

За своєю природою інтерфейс RS-485 пристрої не можуть передавати одночасно - буде конфлікт передавачів. Отже, потрібно розподілити між пристроями право на передачу. Звідси основний поділ: централізований (одновладний) обмін і децентралізований (багатовладний).

1.2 Модулі розподіленої системи збору даних. За використання сучасної ідеології автоматизації різних технологічних процесів все більшої поширеності набувають модулі ADAM від компанії "Advantech".

Модулі серії ADAM 5000 - це модулі розподілених систем збору даних та управління. Вони призначені для створення територіально розподілених систем збору даних та управління і забезпечують виконання таких функцій:

- аналоговий ввід-вивід;
- дискретний ввід-вивід;
- первинне перетворення інформації;

ADAM 5000 отримує команди від віддаленої обчислювальної системи і передає перетворені дані за допомогою інтерфейсу RS-485.

Компанія "Advantech" виробляє моделі інтелектуальних УЗО серії ADAM 5000 - модель ADAM 5510, яка, крім іншого, є вільнопрограмованим мікроконтролером, сумісним з IBM.

Програмований мікроконтролер ADAM 5510 призначений для використання в локальних та розподілених системах автоматизації в якості автономного контролера. Він забезпечує прийом і видачу аналогових та дискретних сигналів, первинне перетворення сигналів згідно з алгоритмами, програмованими користувачем, та обмін інформацією за допомогою послідовних каналів зв'язку на основі інтерфейсу RS-485. Приклад комплектування системи на основі модулів серії ADAM 50xx представлений на рис. 1.3.

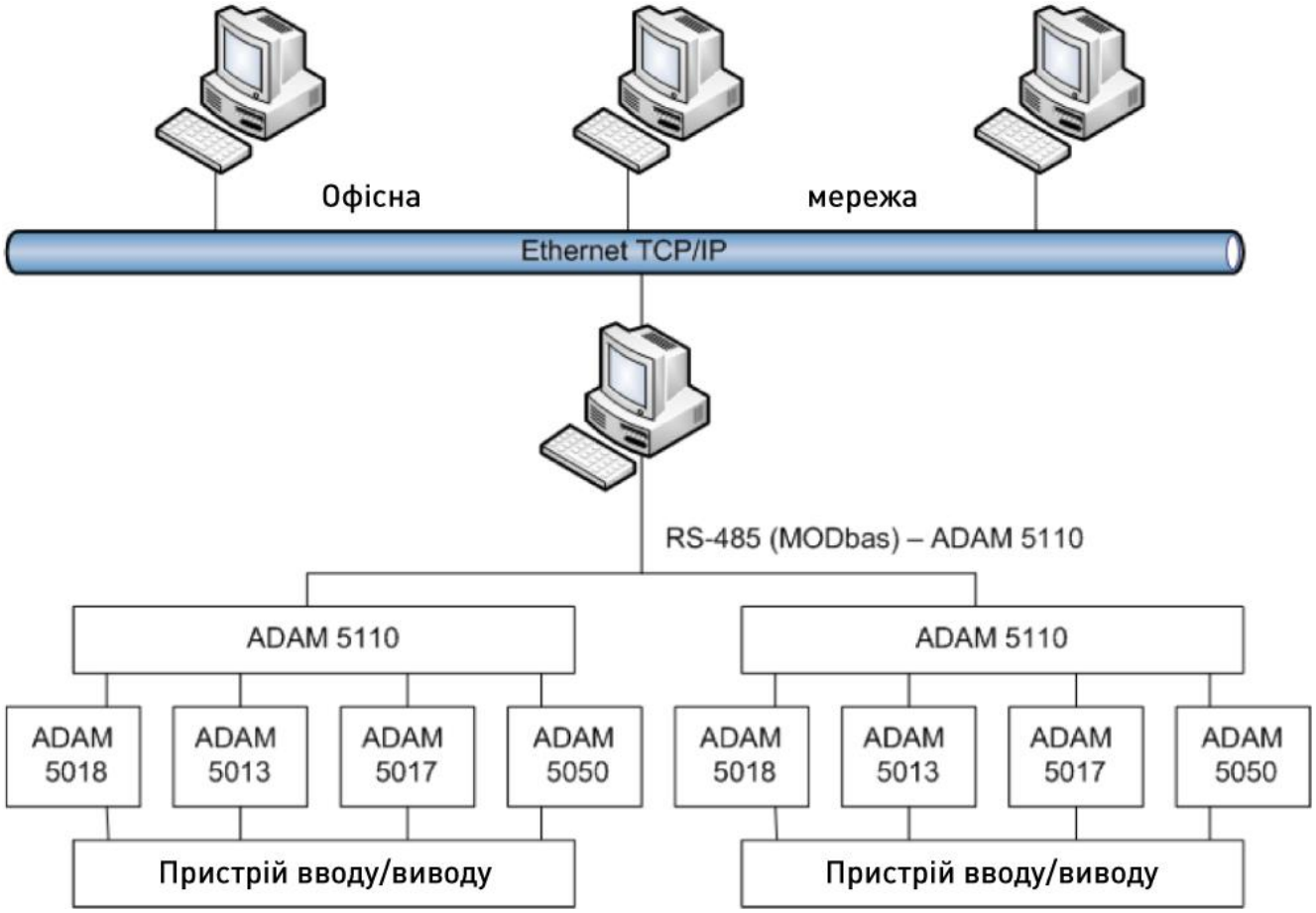

Рисунок 1.3 – Інформаційна система на основі модулів ADAM 50XX

З'явлення модуля ADAM 5510 дозволило розмістити управляючу програму користувача безпосередньо в пам'яті контролера, що скоротило час реакції в контурі управління практично до 5-10 мкс. Цей факт дозволив застосовувати системи на базі ADAM 5510 для вирішення задач жорсткого реального часу.

Контролер має відкриту архітектуру і може програмуватись за допомогою традиційних мов програмування (С, асемблер), а також функціональних блочних діаграм. Для цього є можливість підтримки пакету Ultralogik (FBD - Function Block Diagram, мова функціональних блокових діаграм, стандарт МІК 61131-3).

Ultralogik - це система розробки програмного забезпечення збору даних і управління для промислових контролерів. Програміст, який використовує Ultralogik, може нічого не знати про внутрішнє устрійство ADAM 5510. Для нього важливо розуміти, які модулі введення-виведення присутні в його контролері і на які саме лінії в них надходять різні сигнали.

Ultralogik включає бібліотеки для всіх модулів введення-виведення ADAM 5510, бібліотеки алгоритмів збору даних і управління (ПІ-закони, ПІД-закони і т.д.), передбачена можливість підключення функцій на мовах С, С++, Паскаль, Асемблер.
ADAM 5018 - 8-канальний модуль введення для підключення термопар;
ADAM 5013 - 3-канальний модуль введення для підключення термометрів зіпротивлення;
ADAM 5017 - 8-канальний модуль аналогового введення;
ADAM 5050 - 16-канальний універсальний модуль дискретного введення-виведення.

1.3 Постановка завдання дослідження.

У даній роботі необхідно провести дослідження і порівняльний аналіз промислових протоколів збору і передачі даних, реалізованих на послідовному інтерфейсі. Провести аналіз обчислювальної потужності, що споживається при реалізації промислових протоколів.
Спроектувати модуль розподіленого збору даних зі швидкістю передачі даних 3 Мбіт на канал зв'язку, реалізованому на інтерфейсі RS-485 і промисловим протоколом передачі даних ModBus для забезпечення обміну даними зі слугуючими контролерами.
Цей модуль має бути оснащений двома COM-портами для зв'язку з персональним комп'ютером і модемом, а також двома Ethernet каналами з системою 10Base-T для клієнтсько-серверного зв'язку з сервером зі швидкістю передачі даних 10 Мбіт/с. Провести розрахунки ефективності використання Ethernet каналу при наявності 10 клієнтів.

2. Науково-дослідна частина

2.1 Протоколи передачі даних. У більшості випадків при асинхронній передачі файлів на ПК базова структура пакета (або кадру) складається з груп полів. Лише одне з цих полів містить фактичні дані. Інші поля, які називаються службовими, містять інформацію, необхідну отримувачу для підтвердження того, що пакет не містить помилок. Поміж службових полів, які зазвичай містяться в структурі пакета асинхронної передачі даних на основі RS-485, можуть бути наступні:

1. Початок пакета (Start-of-Frame, SOF): Це службове поле вказує на початок пакета і допомагає отримувачу синхронізуватися з передавачем. Зазвичай це один або кілька бітів, які мають фіксоване значення або особливий код.

2. Адреса приймача і передавача: Ці поля вказують на адресу приймача та передавача пакета. Вони використовуються для ідентифікації пристроїв, яким адресовані дані.

3. Контрольна сума (Checksum): Це поле містить обчислену контрольну суму або код перевірки цілісності (CRC), яка дозволяє отримувачу перевірити, чи були передані дані без помилок. Контрольна сума обчислюється на основі даних пакета та використовується для виявлення та корекції помилок.

4. Довжина пакета: Це поле вказує на загальну довжину пакета, включаючи дані та службові поля. Ця інформація допомагає отримувачу коректно розпізнати та обробити пакет.

5. Прапорець закінчення пакета (End-of-Frame, EOF): Це службове поле вказує на кінець пакета. Зазвичай це один або кілька бітів, які мають фіксоване значення або особливий код.

Ці службові поля відіграють важливу роль у передачі даних через RS-485, оскільки допомагають забезпечити правильність та надійність передачі. Вони дозволяють виявляти та виправляти помилки, забезпечують синхронізацію між передавачем та приймачем, а також допомагають відокремлювати дані.

Протоколи автоматичного запиту повторної передачі (ARQ)

Найбільш поширеним типом пакетного протоколу є протокол автоматичного запиту повторної передачі (ARQ), при використанні якого в прийнятих пакетах визначається наявність помилки, а відсутність підтвердження правильності пакета автоматично призводить до його повторної передачі.

Принцип передачі та очікування ARQ полягає в тому, що отримувач отримує пакет і після перевірки правильності його номера відносно попереднього обчислює часткову контрольну суму для даних цього пакета. У разі відповідності

контрольної суми, зазначеної в пакеті, отримувач видає підтвердження у формі ACK (або він надсилає NAK). Коли передавач отримує ACK, він передає наступний пакет.

Принцип неперервного ARQ (ковзке вікно) полягає в тому, що передавач надсилає кілька пакетів підряд без затримки між пакетами. Отримувач разом з номером пакета надсилає NAC або ACK (як і в разі передачі та очікування ARQ).

Передавач постійно перевіряє потік повертаються підтверджень і відстежує пакети з помилками. В кінці передачі пакети, в яких були виявлені помилки, передаються повторно.

Протокол передачі та очікування ARQ залишається найбільш поширеним протоколом передачі файлів для ПК. Для передачі файлів використовуються два типи протоколів - XMODEM та Kermit.

Протокол XMODEM. XMODEM є простим протоколом передачі та очікування ARQ, який використовує поле даних фіксованої довжини. Контрольне число є однобайтовою арифметичною контрольною сумою. Структура пакета зображена на рис. 2.1.

Початок заголовку	Номер поточного пакету	Доповнення до 1 номера поточного пакету	Дані (128 байт)	Арифметична Контрольна сума

Рисунок 2.1 - Структура кадру протоколу XMODEM

Початок заголовка (SOH) - це значення у двійковому коді ASCII. Поле номера поточного пакета приймає значення 256 і є доповненням до першого пакета. Довжина даних є фіксованою і може містити як двійкові дані, так і слова двійкового коду ASCII. Арифметична контрольна сума обчислюється для поля даних за допомогою методу ділення за модулем 226 і має розмір один байт.

Коротко алгоритм роботи протоколу XMODEM можна описати так:

Приймач надсилає передавачу символ NAK, щоб ініціювати передачу.

Потім передавач надсилає об'єднаний пакет з 128-байтним блоком даних. Символ ACK, отриманий передавачем, починає передачу наступного пакета, а символ CAN скасовує передачу.

Коли всі дані передані, відправник надсилає єдиний символ EOT, який приймач підтверджує символом ACK.

Інші версії протоколу XMODEM використовують однобайтову контрольну суму (XMODEM-CRC) замість арифметичної контрольної суми. Це незворотний алгоритм циклічного зайвого коду (CRC) з CCITT поліноміальним дільником, який представлений у формулі 2.1.

$$П = X^{16} + X^{12} + X^5 + 1, \quad (2.1)$$

Проблеми протоколу XMODEM:

Протокол XMODEM призначений для каналів з низькою частотою появи помилок. Якщо модем не має вбудованого алгоритму корекції помилок, то протокол XMODEM не буде працювати. Висока швидкість передачі даних і погана якість зв'язку (з пропозицією більш дешевих послуг) погіршують цю проблему;

Лінійні перешкоди, що призводять до численних змін бітів, часто проходять через просту систему виявлення помилок протоколу XMODEM і залишаються непоміченими;

Для створення пакету символів потрібні 8 бітів, що не дозволяє їх використовувати в комунікаційних системах, які виділяють лише 7 бітів на кожен символ.

Великою перевагою протоколу XMODEM є те, що він підтримується всіма поширеними комунікаційними програмами (такими, як Windows Terminal). Цей стандарт de facto є корисним для передачі файлів між різними несумісними комп'ютерними системами, які мають спільним тільки протокол XMODEM.

Основні можливості протоколу YMODEM:

Протокол YMODEM має менше сервісних витрат, ніж XMODEM, оскільки його кадр містить 1024 байти, які можна передавати одним блоком. Протокол YMODEM є досить гнучким і дозволяє зменшити розмір блока даних до 128 байтів, якщо частота появи помилок дуже велика (що призводить до численних повторних передач);

Протокол YMODEM надійніший, ніж XMODEM. Щоб припинити передачу файлу, повинна бути прийнята послідовність символів: Control-X, Control-X (або два символи CAN). Це запобігає ненавмисним зупинкам передачі файлу через помилки на лінії;

Для забезпечення високої надійності використовується контроль за допомогою циклічного зайвого коду;

Інформація, пов'язана з файлом (ім'я, час, дата та розмір файлу), також передається на приймальний комп'ютер;

Основа передачі файлу - 8-розрядні дані. Це призводить до певних проблем, які були розглянуті раніше для протоколу XMODEM.

Протокол ZMODEM не буде передавати файл, якщо такий файл вже існує на жорсткому диску приймаючого комп'ютера. Ця перевірка виконується протоколом ZMODEM автоматично і дозволяє заощадити час;

Забезпечує краще виявлення і корекцію помилок за допомогою механізму CRC-32;

Можлива стиснення даних - це збільшує швидкість передачі даних.

Протокол Kermit. На основі протоколу Kermit лежить пакетний протокол передачі та очікування ARQ. Відправник передає пакет, а потім очікує підтвердження приймача про прийняття пакета. Приймач може потім запросити наступний пакет (символ ACK) або повторну передачу попереднього пакета (символ NAK).[1]

Малюнок протоколу Kermit показано на рисунку 2.2.

Мітка	Довжина пакету	Номер пакету	Тип	Дані	Перевірка 1	Перевірка 2	Перевірка 3

Рисунок 2.2 - Пакет протоколу Kermit

Порівняльні характеристики типів передачі файлів і атрибутів протоколів наведені в таблиці 2.1.

Таблиця 2.1 - Порівняння протоколів передачі файлів

Опис		**Xmodem**	**Ymodem**	**Zmodem**	**Kermit**
Типи	Одиночний файл	Так	Так	Так	Так

	Група файлів	Ні	Так	Так	Так
	Текст/дані/двійкова інформація	Так	Так	Так	Так
	Кількість бітів даних	8	8	8	7 або 8
	Стиснення даних	Ні	Ні	Так	Так
	Контроль помилок	Контрольна сума	Контрольна сума /CRC-CCITT	CRC-CCITT/ CRC-32	Контрольна сума / CRC-CCITT
	Відповідь при контролі помилок	Один символ	Один символ	Один символ	Пакетний контроль за
	Розмір пакета даних (байти)	128	1024	1024	0-95
Передача атрибутів файлів	Розмір/час/дата файлу	Ні	Так	Так	Так
	Підтримка ковзкого вікна	Один пакет	Один пакет	1 або кілька пакетів	1 або кілька пакетів
	Параметри домовленості	Ні	Так	Так	Так
	Пакети в Windows	Ні	Ні	Так	Так
	Тип контроля за помилками	Ні	Так	Так	Так
	Стиск даних	Ні	Ні	Так	Так
	Контроль XON/XOFF	Ні	Ні	Так	Так

2.2 Промисловий протокол Modbus. Протокол Modbus отримав широке поширення в сучасних автоматизованих системах. Специфікація протоколу не визначає тип фізичного рівня мережі передачі даних, залишаючи вибір за розробником.

Протокол побудований за схемою "мастер-ведений" (master-slave). В системі є один ведучий пристрій (мастер), який ініціює будь-яку транзакцію в мережі. Всі інші пристрої є веденими і виконують команди мастера або передають інформацію у відповідь на запит мастера. Протокол передбачає два типи адресації: індивідуальну, коли повідомлення адресоване одному веденому пристрою, та широкомовленню (broadcast messages), коли повідомлення адресоване всім пристроям мережі. При індивідуальній адресації ведений пристрій повертає мастеру відповідне повідомлення, при широкомовленні відповідні повідомлення не надсилаються.

Мережа Modbus може працювати у одному з двох режимів: RTU та ASCII. У режимі RTU інформація передається "як є", у двійковому коді. У режимі ASCII інформація передається у текстовому форматі як послідовність символів '0'–'9', 'A'–'F' в ASCII-кодуванні.

Повідомлення представляє собою послідовність символів, передаваних безперервно, без пауз. При виникненні паузи тривалістю більше 1,5 t, де t - час передачі одного символу при заданій швидкості передачі, приймаючий пристрій фіксує помилку в прийманні повідомлення і починає приймати нове повідомлення після паузи. Повідомлення передує і завершується паузою не менше 3,5 t з типовим значенням 4 t. [7]

Ймовірність того, що кодова відстань між отриманим та переданим повідомленнями довжиною 64 біта дорівнює d, розраховується за формулою 2.2:

$$p(d) = p_l^d (1 - p_l)^{64-d}, \tag{2.2}$$

де p_l - ймовірність спотворення одного біта.

Нехай $\Pr_{CRCerr}(d)$ - ймовірність збігу контрольної суми CRC-16 в інформаційному слові, розташованому на відстані d від передаваного інформаційного слова. Тоді повна ймовірність невиявлення помилки контролем перевірочних символів CRC16 розраховується за формулою 2.3:

$$\Pr_{CRCerr}(p_l) = \sum_{d=2}^{p_l - n} p_l^d (1 - p_l)^{64-d} \Pr_{CRCerr}(d), \tag{2.3}$$

Де p_l - ймовірність спотворення одного біта,

$\Pr_{CRCerr}(d)$ - ймовірність помилки, що не виявляється.

Імовірність оцінена за допомогою моделювання та дорівнює $1,9 \cdot 10^{-5}$.

Ми бачимо, що обидва режими мають свої переваги та недоліки, і неможливо однозначно рекомендувати використання одного з них "у всіх випадках життя". Найкращим рішенням буде підтримка обох режимів протоколу Modbus у розроблюваному пристрої.

2.3 Аналіз споживаної обчислювальної потужності.

При формуванні і обробці пакета даних згідно з різними промисловими протоколами передачі даних основний час і обчислювальна потужність, як приймача, так і передавача, йде на обчислення та перевірку контрольної суми. Для протоколів, які під час трансакції формують довгі повідомлення, поширеним і дуже ефективним механізмом перевірки повідомлення на наявність помилки є контроль за допомогою циклічного зайвого коду. CRC базується на використанні понять з області математики, відомої як теорія алгебри. Механізм контролю за допомогою циклічного зайвого коду очевидно, дуже ефективний у виявленні помилок, особливо "пакетних помилок", коли зовнішні перешкоди тимчасово підгнічують сигнал і спотворюють цілий рядок бітів. Контроль CRC є ефективним для повідомлень будь-якої довжини.

Існують два основних методи розрахунку CRC - по бітах і по таблиці. Їх основні характеристики наведені в таблиці 2.2.

Контрольні суми відрізняються за надійністю контролю цілісності даних. Чим більша довжина розрахункової суми, тим вона надійніша, але й тим складніше її реалізація. Під час розрахунку можуть використовуватися різні поліноми, які зазвичай визначаються стандартом використовуваного інтерфейсу.[3]

Метод розрахунку	Переваги	Недоліки
Побітовий	Малі затрати пам’яті	Низька швидкість виконання

Табличний	Більш висока швидкість виконання	Великі затрати пам’яті

Таблиця 2.2 - Характеристики методів розрахунку

Реалізація побітового алгоритму. Схема алгоритму представлена на рисунку 1.6. Вихідний код, що відповідає даному алгоритму, наведений у додатку. На рисунку 1.6 для кожного блоку вказано час виконання операцій у тактах процесора. Підпрограма розрахунку контрольної суми приймає два параметри:
довжину повідомлення, яка передається словом через два регістри LENL і LENH, де LENL - молодший, а LENH - старший байт слова;
вказівник на повідомлення, який передається через пару адресних регістрів XH і XL. Розрахована контрольна сума зберігається в регістрах SUMH і SUML, де SUMH - старший, а SUML - молодший байт суми. У побітовому алгоритмі час розрахунку залежить від вхідних даних. Тому при аналізі часу розрахунку беруться два випадки - найгірший (виконується операція XOR з поліномом) і найкращий (операція XOR не виконується). На практиці можна стверджувати, що дані будуть оброблені не повільніше, ніж у найгіршому випадку, і не швидше, ніж у найкращому випадку. Виділимо постійну і змінну частини часу виконання алгоритму. Постійна частина визначається накладними витратами при реалізації підпрограми - завантаження/вивантаження зі стеку, ініціалізація початкових значень змінних, інверсія суми після завершення розрахунку. Змінна частина визначається реалізацією алгоритму. Дані про часові витрати в тактах процесорного часу наведені в таблиці 1.4.

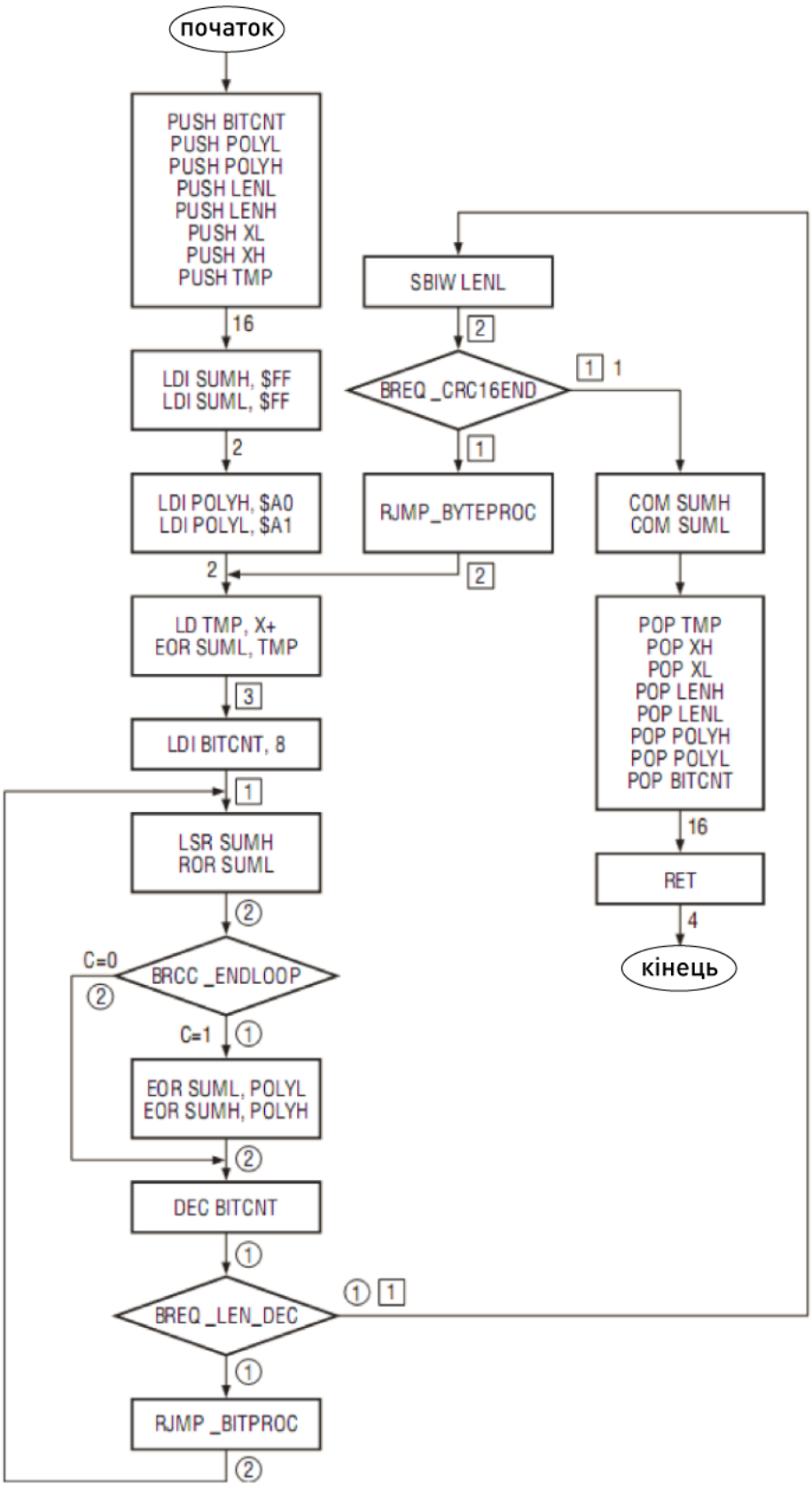

Рисунок 2.2 – Блок-схема алгоритму побітового розрахунку CRC-16

Таблиця 2.3 – Тимчасові витрати при реалізації побітового алгоритму

Розмір повідомлення	Найкращий випадок			Найгірший Випадок		
	Тактів в циклі обробки байту	Всього тактів	Тактів на один байт	Тактів в циклі обробки байту	Всього тактів	Тактів на один байт
10	64	117,0	117,0	72	125	125,0
20	128	191,0	95,5	144	207	103,5
40	256	339,0	84,8	288	371	92,8
80	512	635,0	79,4	576	699	87,4
160	1024	1227,0	76,7	1152	1355	84,7
320	2048	2411,0	75,3	2140	2667	83,3

Блок-схема алгоритму розрахунку контрольної суми CRC16 табличним методом представлена рисунку 2.3.

З блок-схеми алгоритму видно, що його виконання залежить від кількості байт оброблюваного повідомлення не залежить від їх значення. При розрахунку табличний спосіб використовуються теж параметри, що і в підпрограмі розрахунку послідовним методом. Дані про тимчасові витрати в тактах процесорного часу наведені в таблиці 2.4.

Таблиця 2.4 - Тимчасові витрати при реалізації табличного алгоритму

Розмір повідомлення	Тактів поза циклом	Тактів в циклі	Всього тактів	Тактів на один байт
1	29	18	47	47,0
2	29	36	65	32,5
4	29	72	101	25,3
8	29	144	173	21,6
18	29	288	317	19,8
32	29	576	605	18,9

Дані про використовувану Flash-пам'ять та SRAM-пам'ять при реалізації табличного та побітного алгоритму наведені в таблиці 2.4.

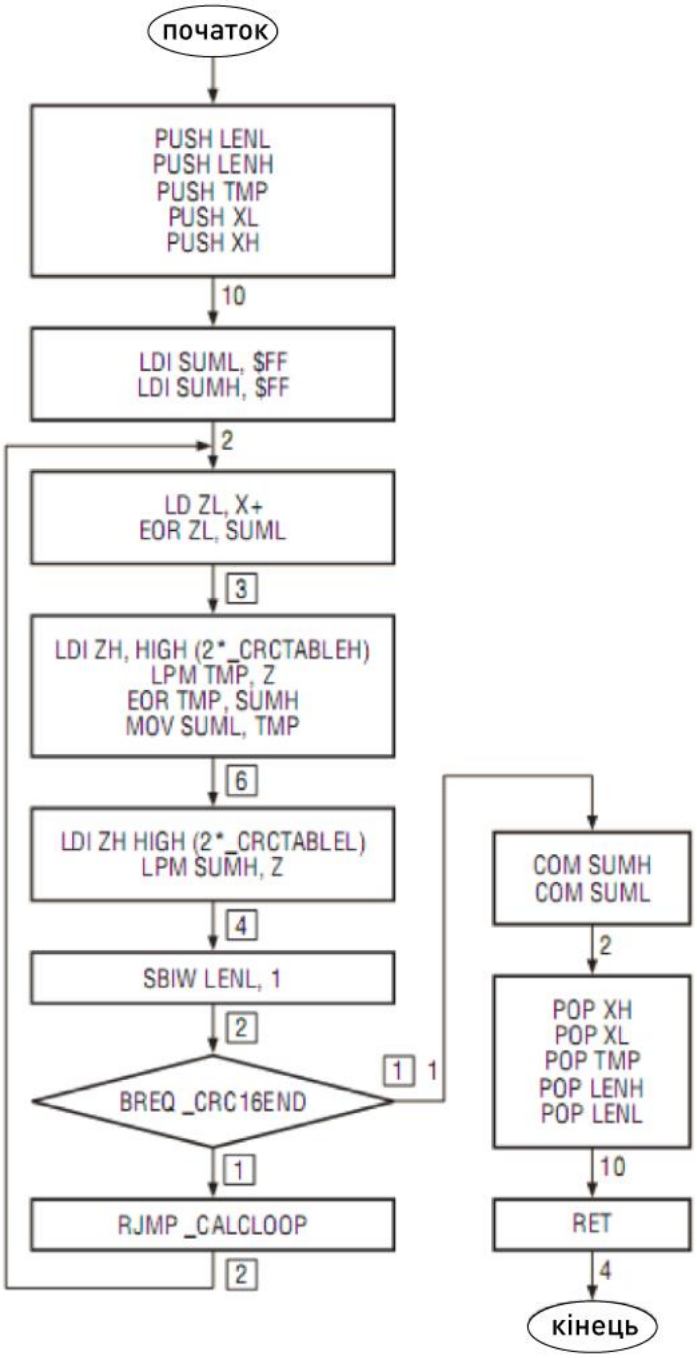

Рисунок 2.3 – Блок-схема табличного алгоритму

Порівняння швидкості виконання алгоритму. Для порівняння швидкості розрахунку побудуємо графіки, що відображають залежність часу виконання розрахунку в процесорних тактах (рис. 2.4) і мілісекунд при тактовій частоті 16 МГц (рис. 2.5) від кількості оброблюваних байт повідомлення.

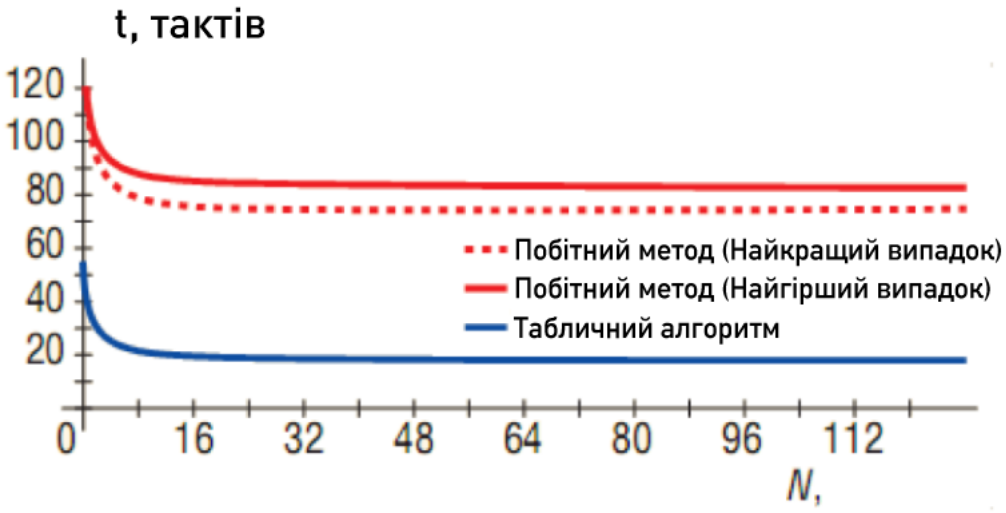

Рисунок 2.4 - Порівняння часу виконання алгоритмів (у тактах)

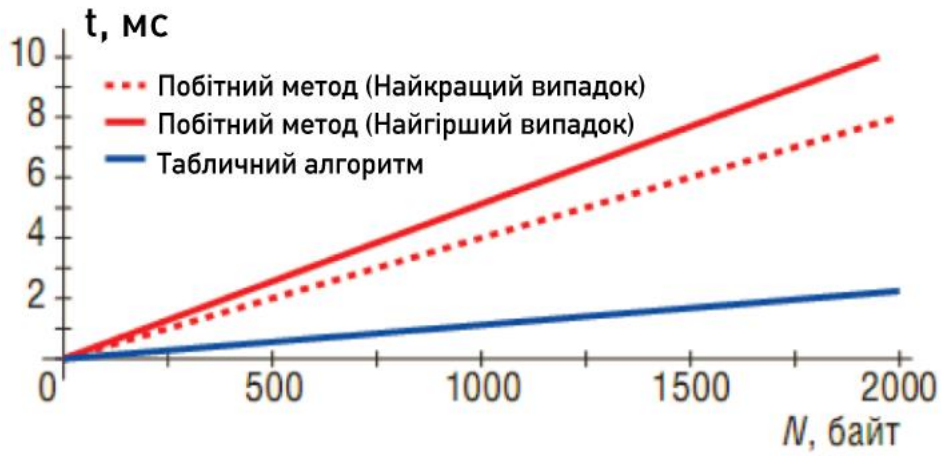

Рисунок 2.5 – Порівняння часу виконання алгоритмів (у мілісекундах)

Таблиця 2.5 - Порівняння обсягів зайнятої пам'яті

Алгоритм	Flash-пам`ять, байт	SRAM, байт
Побітовий	74	0
Табличний	564	0

Отже, використання табличного алгоритму для обчислення контрольної суми CRC програмним способом є більш ефективним, навіть при використанні менших моделей мікроконтролерів серії AVR.

3 РОЗРОБКА ТЕЛЕКОМУНІКАЦІЙНОГО ПРИСТРОЮ З ВИКОРИСТАННЯМ ОТРИМАНИХ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

3.1 Обґрунтування алгоритму функціонування та структурної схеми проектованого пристрою. Значення полів та структура повідомлень, що передаються по каналу зв'язку, а також алгоритми функціонування передавачів та приймачів описуються стандартами протоколів передачі даних. Вибір конкретного рішення залежить від архітектури системи, відстані, кількості контрольованих точок, вимог до пропускної здатності та надійності каналу. На сьогодні існує десятки різноманітних промислових протоколів передачі даних. Найпопулярнішими з них є Industrial Ethernet, Profibus, Controller Area Net Work bus (CANbus), MAP/TOP, Allen Bradley Data Highway, EthernetIP.

Доступ до шини Modbus здійснюється за принципом головне/підпорядковане пристрій, причому протокол передбачає до 247 підпорядкованих пристроїв для одного головного пристрою. Транзакції виконує лише головний пристрій.

Транзакції мають тип запит/відповідь, коли звернення відбувається лише до одного підпорядкованого пристрою або одночасно до всіх підпорядкованих пристроїв (без відповіді). Транзакція включає один кадр запиту і один кадр відповіді або один кадр звернення до всіх пристроїв. Деякі параметри протоколу Modbus є фіксованими, такі як формат кадру, послідовність кадрів, обробка комунікаційних помилок та виключних ситуацій, а також виконувані функції. Інші параметри можна вибирати, такі як середовище, характеристики та режим передачі - RTU або ASCII.

Користувацькі характеристики встановлюються на кожному пристрої і не можуть бути змінені під час роботи системи.

Протокол Modbus формує кадри для передачі повідомлень між головним і підпорядкованими пристроями. Інформація повідомлень включає адресу приймача, дію приймача, дані, необхідні для виконання цієї дії, та засоби контролю помилок. Підпорядкований пристрій зчитує повідомлення і, якщо помилок не виявлено, виконує дії і надсилає відповідь головному пристрою. Інформація в відповідному повідомленні: адреса підпорядкованого пристрою, виконана дія, результат дії та контроль помилок. Якщо початкове повідомлення призначене для всіх пристроїв, то від підпорядкованих пристроїв не потрібне підтвердження.

Перше поле кожного кадра повідомлення є адресовим полем, яке складається з одного байта інформації. У запитових кадрах цей байт ідентифікує контролер, до якого звертається запит. Відповідний кадр відповіді починається з адреси відповідаючого пристрою. Кожен підпорядкований пристрій може мати адресне поле в межах від 1 до 247, хоча кількість підпорядкованих пристроїв на практиці, як правило, менше. Типова конфігурація Modbus складається з одного головного пристрою та двох-трьох підпорядкованих пристроїв.[4]

Адресне поле	Поле функції	DATA Поле даних	Поле контроля за помилками
1 байт	1 байт	Перемінна довжина	2 байти

Таблиця 3.1 – Формат кадра повідомлення Modbus

Друге поле кожного повідомлення є полем функції, яке також складається з одного байта інформації. У запитовому кадрі головного пристрою цей байт ідентифікує функцію, яку цільовий ПЛК повинен виконати.

Якщо цільовий ПЛК може виконати запитувану функцію, то поле функції його відповіді буде ехом оригінального запиту. У протилежному випадку, в полі функції відповіді старшому байту буде присвоєно значення одиниці, що є сигналом від'ємної відповіді.

Третє поле кадру повідомлення є полем даних, і його довжина змінюється в залежності від функції, визначеної в полі функції. У запиті головного пристрою це поле містить інформацію, необхідну ПЛК для виконання запитуваної функції. У відповіді ПЛК це поле містить дані, які запитує головний пристрій.

Контроль помилок. Поширеним і дуже ефективним механізмом перевірки повідомлення на наявність помилок є контроль за допомогою циклічного надлишкового коду (CRC). CRC базується на використанні понять з області математики, відомої як алгебра теорії, і є досить простим у реалізації. Використовуючи 16-бітове контрольне значення, CRC обіцяє виявлення помилок, передбачуване таблицею 3.2. Механізм контролю за допомогою циклічного надлишкового коду очевидно є дуже ефективним у виявленні помилок, особливо "пакетних помилок", коли

зовнішні перешкоди тимчасово пригнічують сигнал і пошкоджують цілий рядок бітів. Контроль CRC є ефективним для повідомлень будь-якої довжини.

Таблиця 3.2 – Виявлення помилок за допомогою CRC

Кратність помилки	Ймовірність виправлення
Помилки в одному біті	100%
Помилки в двох бітах	100%
Непарна кількість помилок	100%
Пакет помилок коротше 16 біт	100%
Пакет помилок рівний 16 бітам	99,9969%
Усі інші пакетні помилки	99,9984%

CRC-16 є широко використовуваним поліномом, особливо в промислових протоколах. Поліном наведено у формулі 3.1.

$$X^{16} + X^{12} + X^{2}+1, \quad (3.1)$$

де X - константа одиниці.

Метод виявлення помилок CRC-16 використовує арифметику за модулем 2, де додавання і віднімання дають один і той же результат. Вихід еквівалентний логічній операції виключаюче АБО (XOR).

Використовуючи цей алгоритм як основу, можна отримати наступне рівняння 3.2:

$$(\text{Повідомлення} \times 2^{16}) / \text{Дільник} = \text{Частка} + \text{Залишок}, \quad (3.2)$$

де: Повідомлення - послідовність бітів, наприклад, послідовність ASCII;

216 - при множенні ефективно додає 16 нулів з правого боку повідомлення;

Дільник - число, на яке проводиться ділення отриманого числа і є породжувальним поліномом;

Частка - результат ділення;

Залишок - значення, яке залишається в результаті ділення і є контрольною сумою CRC.

З рівняння 3.1 ми отримуємо рівняння 3.2:

$$[(\text{Повідомлення} \times 2^{16}) + \text{Залишок}] / \text{Дільник} = \text{Частка}, \quad (3.3)$$

Передавач, відповідно до формули 3.1, формує контрольну суму (Залишок) і передає її отримувачу. Отримувач, отримавши пакет даних, згідно з формулою 3.2,

проводить ділення повідомлення і контрольної суми на генеруючий поліном CRC-16. У разі правильної передачі результат обчислення (Частка) дорівнює "0", в іншому випадку переданий пакет був спотворений.

Розглянуті схеми виявлення помилок дозволяють отримувачеві лише виявити, що дані були пошкоджені. Вони не надають засобів для виправлення помилкового символу або кадру. Зазвичай цю корекцію виконує отримувач, повідомляючи передавачу, що була виявлена помилка, і запитує іншу копію переданого повідомлення. Цей комбінований цикл виявлення помилок/корекції називається контролем помилок.

Середня ефективна швидкість передачі даних при реалізації протоколу Modbus з передачею інформації розміром 5 байт розраховується за формулою для системи з зворотним зв'язком 3.3:

$$V_{ñðýô} = V\left(1-\frac{P_o n_á}{1-P_o n_á}\right)\left(1-\frac{r+n_{ñë}}{n_á}\right)=1,8 \qquad (3.4)$$

Де V = 3 Мбіт/с - швидкість передачі даних по каналу зв'язку;

P_O = 10-6 - ймовірність спотворення переданого пакета.

Узагальнений алгоритм обміну даними за протоколом Modbus. Узагальнений алгоритм обміну даними між головним та підпорядкованим пристроєм можна зображення у вигляді, показаному на рис. 3.1.

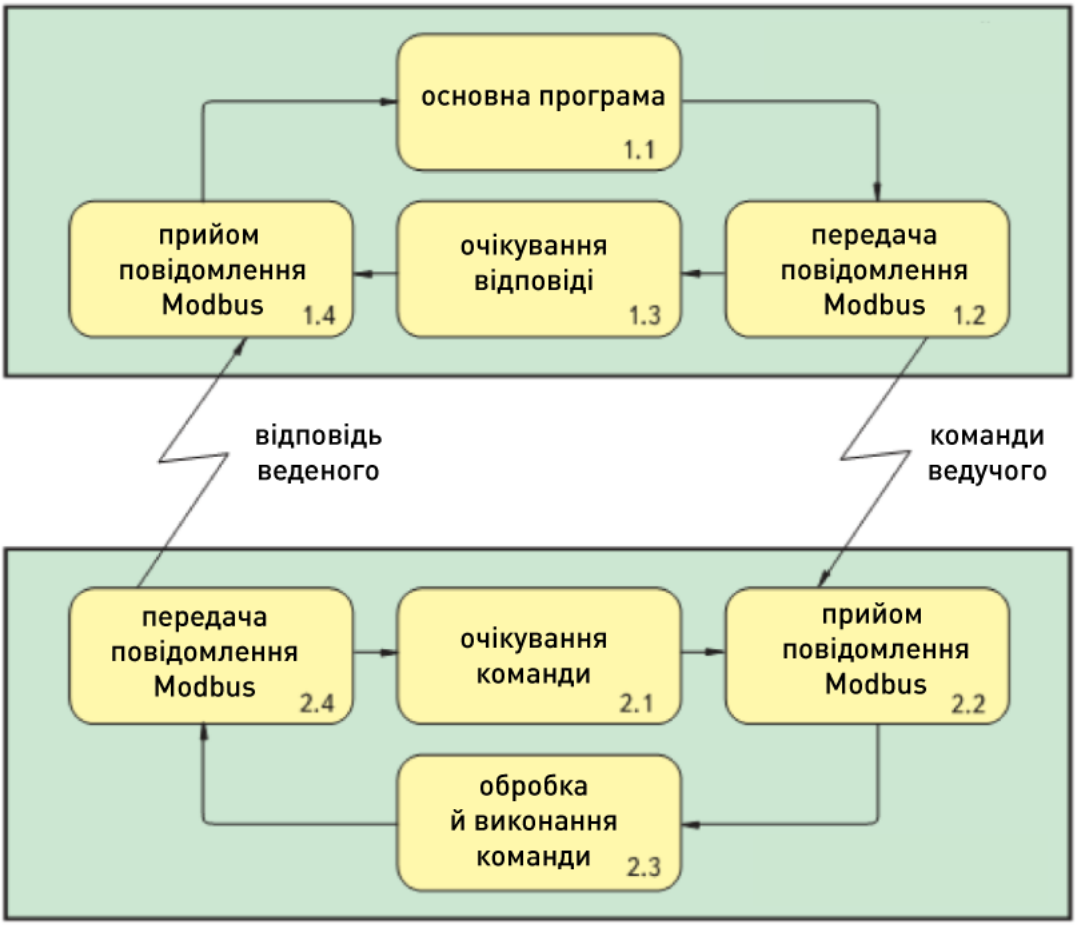

Рисунок 3.1 - Узагальнений алгоритм обміну даними між керуючим та керованим пристроями

Отже, керуючому пристрою під час виконання цільової програми (стан 1.1 на рис. 3.1) потрібно запитати дані віддаленого модуля або встановити виходи віддаленого модуля у певний стан. Керуючий пристрій (надалі - майстер) формує повідомлення Modbus, яке в загальному випадку містить: адресу підпорядкованого пристрою, код команди, додаткові дані та контрольну суму. Сформоване повідомлення передається через UART мікроконтролера та фізичну лінію зв'язку до підпорядкованого пристрою (стан 1.2). Після передачі повідомлення майстер переходить у стан очікування відповіді від підпорядкованого (стан 1.3). Підпорядкований пристрій, який перебуває у стані очікування команди від майстра (стан 2.1), при надходженні даних в буфер UART переходить у стан прийому (стан 2.2) і у разі успішного завершення прийому переходить до декодування та виконання команди (стан 2.3). Після виконання команди підпорядкований пристрій формує відповідне повідомлення, яке в загальному випадку містить адресу підпорядкованого пристрою (тобто власну адресу), код команди, додаткові дані та контрольну суму.

Сформоване повідомлення передається керуючому пристрою (стан 2.4), і після успішного прийому відповіді від підпорядкованого (стан 1.4), керуючий пристрій продовжує роботу з основною програмою, а підпорядкований переходить до очікування наступної команди. Тим самим транзакція Modbus завершується.

Передача та прийом повідомлення. Передача повідомлення в режимі RTU здійснюється найпростішим способом (рис. 3.2): передавальний пристрій формує повідомлення (стан 1), чекає, за потреби, звільнення буфера передавача UART (стан 2) та відправляє повідомлення (стан 3). Не передбачається жодних аварійних ситуацій та помилок.

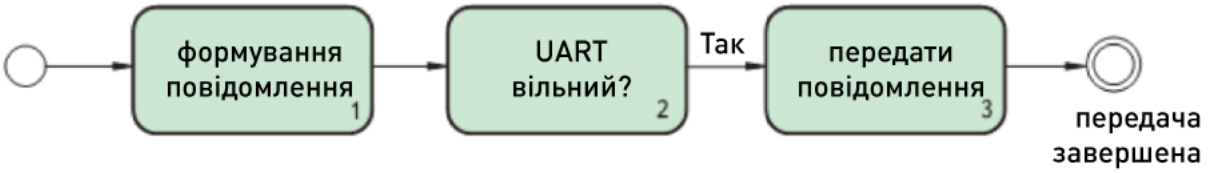

Рисунок 3.2 – Кінцевий автомат передачі повідомлення Modbus в режимі RTU

Кінцевий автомат, що здійснює прийом повідомлення Modbus в режимі RTU, представлений на рис. 3.3. Його робота проста. У стані 1 лічильник буфера I обнуляється. У стані 2 кожен отриманий символ записується в буфер Data. Перехід із стану 2 (поміщення повідомлення в буфер) до стану 3 (перевірка поля адреси) здійснюється за тайм-аутом, значення якого повинно бути рівним 3,5 тривалості передачі одного байта.[5]

У стані 3 відбувається перевірка поля адреси повідомлення. Для підпорядкованого пристрою це поле повинно співпадати з адресою пристрою. Якщо ж прийом здійснюється керуючим пристроєм, тобто повідомлення є відповіддю підпорядкованого на раніше відправлену команду, то у полі адреси повинна міститись адреса підпорядкованого, що надіслав відповідь. Якщо цього не відбувається, то повідомлення адресоване іншому пристрою і автомат знову переходить у стан 1 та очікує початку наступного повідомлення. Якщо ж адреса співпадає, автомат переходить у стан 4 "обчислення контрольної суми". Якщо контрольна сума співпадає з переданою в повідомленні, прийом повідомлення вважається успішно завершеним і автомат припиняє роботу. Автомат також здійснює обробку ряду аварійних ситуацій, які можуть виникнути під час прийому повідомлення. До таких ситуацій належать наступні:

- розмір буфера перевищив максимально допустимий;

- контрольна сума не співпадає.

Виникнення помилок такого типу при роботі пристрою свідчить про погану якість зв'язку (сильні електромагнітні перешкоди, погане екранування, надто велика довжина лінії зв'язку, невиконання правил монтажу та інше). Поява помилок також може свідчити про некоректну роботу передавального пристрою.

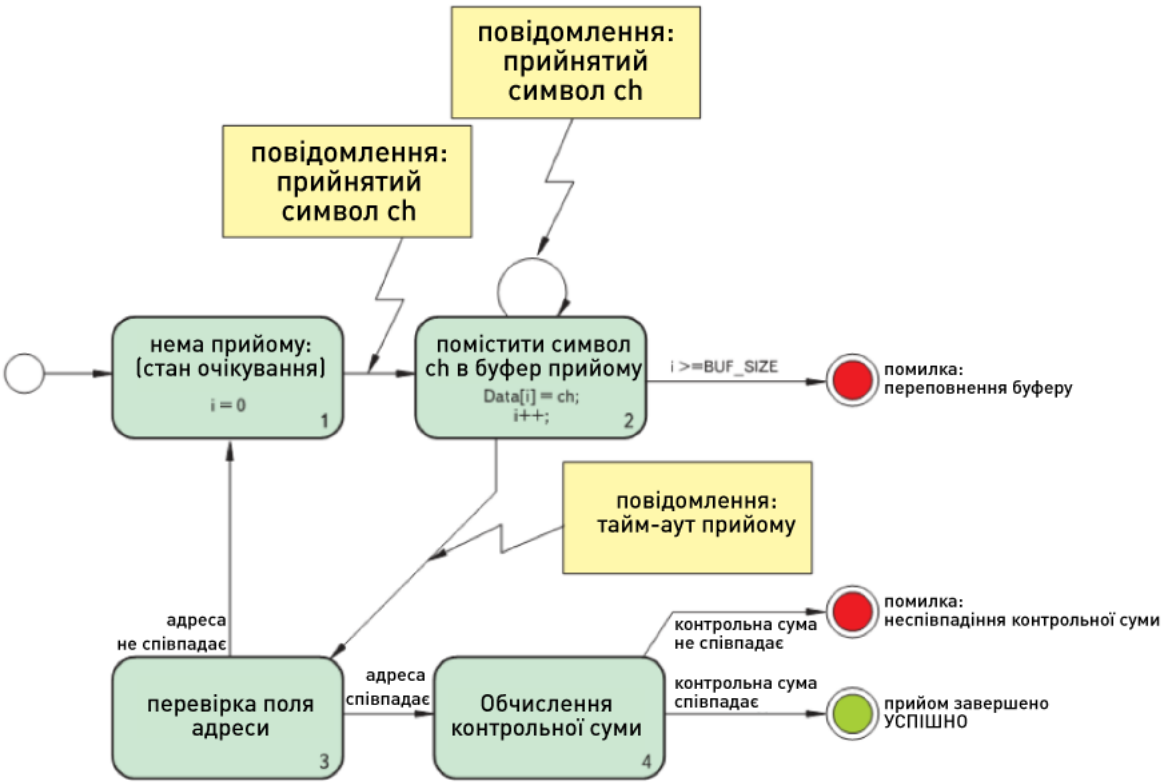

Рисунок 3.3 - Кінцевий алгоритм прийому повідомлення Modbus в режимі RTU

Узагальнений алгоритм роботи керуючого пристрою Modbus. Узагальнений алгоритм роботи керуючого пристрою Modbus зображений на рис 3.4. Робота автомата керується двома "зовнішніми" повідомленнями: "почати транзакцію" - повідомлення від основної програми (передбачається, що Modbus-пакет формується та розміщується в буфері передачі основною програмою) і "початок прийому" - повідомлення, що сигналізує про отримання символу в приймальному буфері UART і формується відповідним обробником переривань UART. Отже, після ініціалізації UART (стан 1) потік переходить у стан очікування 2. Як тільки основна програма видає команду на початок транзакції, автомат переходить у стан 3, де відбувається

передача повідомлення. Після передачі повідомлення контролер очікує відповіді підпорядкованого пристрою (стан 4).

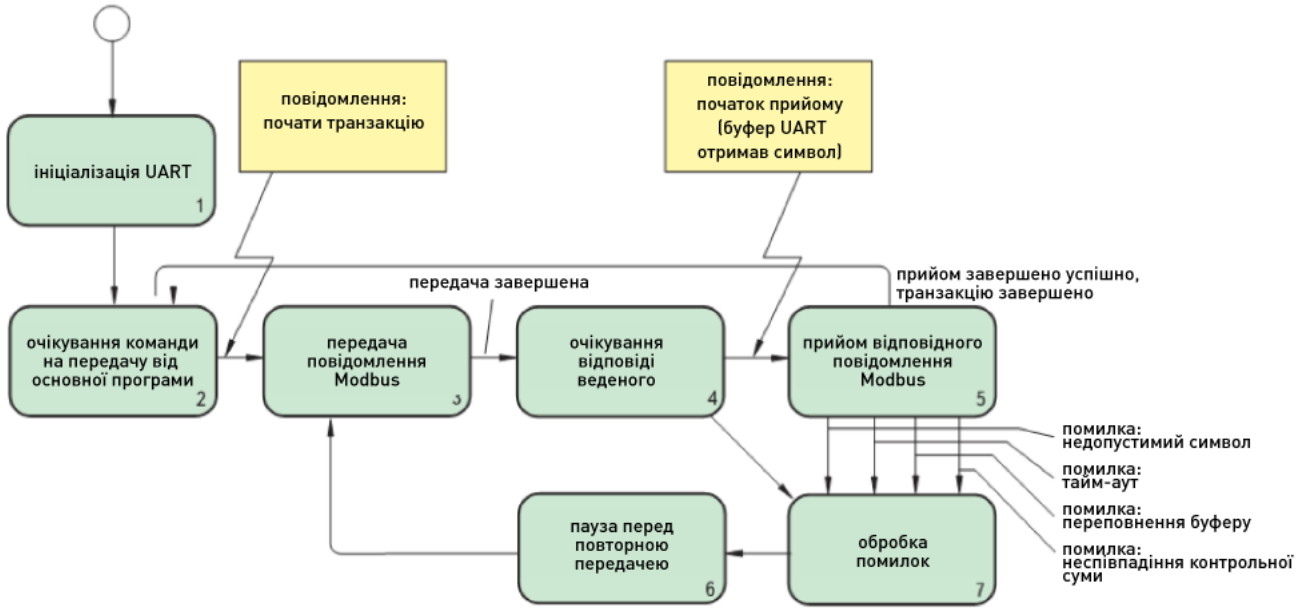

Рисунок 3.4 – Узагальнений алгоритм роботи керуючого пристрою Modbus

Очікування відповіді обмежене в часі і зазвичай не перевищує декількох секунд. При надходженні першого символу відповідного повідомлення в буфер UART автомат переходить у стан 5 (у цьому стані розміщений один з розглянутих вище алгоритмів прийому повідомлення). При успішному прийомі повідомлення транзакція вважається завершеною, і автомат переходить у стан 2, очікуючи наступної команди основної програми. При виникненні помилки під час прийому автомат затримується (стан 6) і повторює передачу повідомлення (стан 3). Зверніть увагу, що між станами 5 і 6 на рисунку зображений стан 7 "обробка помилок". У протоколі Modbus, в принципі, не передбачено жодної специфічної обробки помилок у прийнятих повідомленнях (у разі виникнення помилки при прийомі, керуючий пристрій просто повинен повторити передачу команди). Однак, ми можемо ввести в програму пристрою, наприклад, журнал помилок, з розподілом причин виникнення певної помилки (неспівпадіння контрольної суми, невідповідність адреси тощо).

Узагальнений алгоритм роботи веденого пристрою Modbus. Узагальнений алгоритм роботи веденого пристрою Modbus зображений на рисунку 3.5.

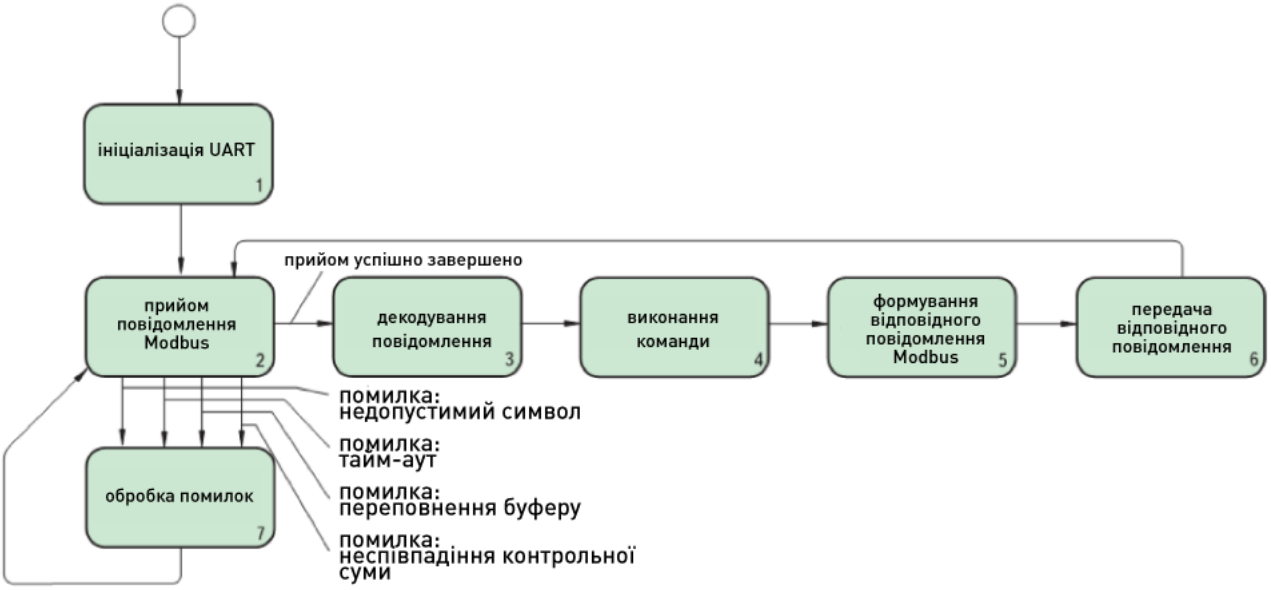

Рисунок 3.5 - Узагальнений алгоритм роботи веденого пристрою Modbus

Він схожий на відповідний алгоритм ведучого пристрою, з єдиною різницею, що операції відбуваються у зворотному порядку: спочатку приймається повідомлення (стан 2), потім після декодування команди, її виконання та створення відповідного повідомлення (стан 3-5), відбувається передача відповіді ведучому пристрою. Різниця полягає також у тому, що при виникненні помилки прийому ведений пристрій не зобов'язаний відправляти відповідне повідомлення, йому просто потрібно зачекати, поки мастер передасть команду повторно. Стан 7, як і у попередньому випадку, не є обов'язковим, але дуже корисним при налагодженні.

Кадр даних протоколу Ethernet. Усі дані передаються у формі пакета або кадру. Кадр складається з оболонки, що містить керуючу інформацію (таку як байти синхронізації та адреси), і фактичного повідомлення. Кожен вузол визначає адресу призначення та зчитує дані, якщо кадр призначений йому.[2]

Кадр даних Ethernet має структуру, представлену в таблиці 3.3.

Таблиця 3.3 – Формат кадру IEEE 802.3

Преамбула	Початковий роздільник	Адреса призначення	Адрес джерела	Індикатор даних	Дані	Контрольна сума CRC
7 байт	1 байт	2 або 6 байт	2 або 6 байт	2 байти	46 – 1500 байт	4 байти

Преамбула - поле складається з семи байтів, кожен з яких має двійкове значення 10101010. Метою цього поля є забезпечення синхронізації всіх приймачів контролера доступу до середовища з кадром.

Роздільник початку кадру (SFD) SFD позначає початок кадру і має двійкове значення 10101011.

Адреса призначення можуть складатися з 16 або 48 бітів - це залежить від конфігурації системи. На практиці адреса призначення майже завжди складаються з 48 бітів. Адреса повинні мати однакову довжину для всіх вузлів. Вузол з таким адресом читатиме повідомлення.

Адреса джерела можуть складатися з 16 або 48 бітів - це залежить від конфігурації системи. На практиці вони майже завжди складаються з 48 бітів. Це адреса вузла, який надсилає дані.

Індикатор довжини - це двобайтовий індикатор довжини, який визначає, скільки байтів містить поле даних.

Дані - це поле, яке містить реальні дані і може мати довжину від 46 до 1500 байтів. Мінімальна довжина поля даних визначається необхідністю виявлення конфлікту, а максимальна довжина обмежує доступний час для будь-якого вузла на 1,2 мілісекунди. Якщо довжина реальних даних менша за 46 байтів, то поле даних повинно бути доповнено до 46 байтів.

Контрольна послідовність кадру - це 32-розрядне значення, яке використовується для виявлення помилок за допомогою циклічного зайвого коду (CRC).

Послідовність передачі:

1. Контролер доступу до середовища формує кадр даних.
2. Контролер доступу до середовища перевіряє лінію на відсутність несучої.
3. Приймач-передавач вузла передає кадр у вигляді потоку бітів.

Приймач-передавач контролює лінію для виявлення можливого конфлікту.

У випадку виникнення конфлікту передавач посилає незначущу послідовність, припиняє передачу і повторює спробу після короткого проміжку часу, що визначається випадковим чином.

Послідовність прийому:

1. Контролер доступу до середовища виявляє сигнал, що надходить від приймача-передавача.
2. Увімкнений сигнал виявлення несучої, щоб заборонити будь-яку передачу контролером доступу до середовища.

3. Контролер доступу до середовища використовує отримані байти преамбули для забезпечення синхронізації.
4. Перевіряється адреса призначення, і якщо вона збігається з адресою вузла, здійснюється зчитування даних.
5. Дані зчитані, перевіряється відповідність індикатора довжини довжині повідомлення. Обчислюється FCS і порівнюється зі значенням поля FCS. Якщо обидва порівняння позитивні, повідомлення передається далі.

Побудова блок-схеми прийому даних ведучим пристроєм. Максимальний розмір пакета даних, переданих за протоколом передачі даних Modbus, не повинен перевищувати 255 байтів. При отриманні пакета даних, розмір якого перевищує встановлене значення, прийом даних припиняється і відправляється повідомлення про некоректну передачу повідомлення. Також повідомлення про некоректну передачу повідомлення відправляється у випадку невідповідності контрольної суми і неприпустимого значення функції.

Загальний алгоритм функціонування модуля розподіленого збору даних наведений на рисунку 3.6

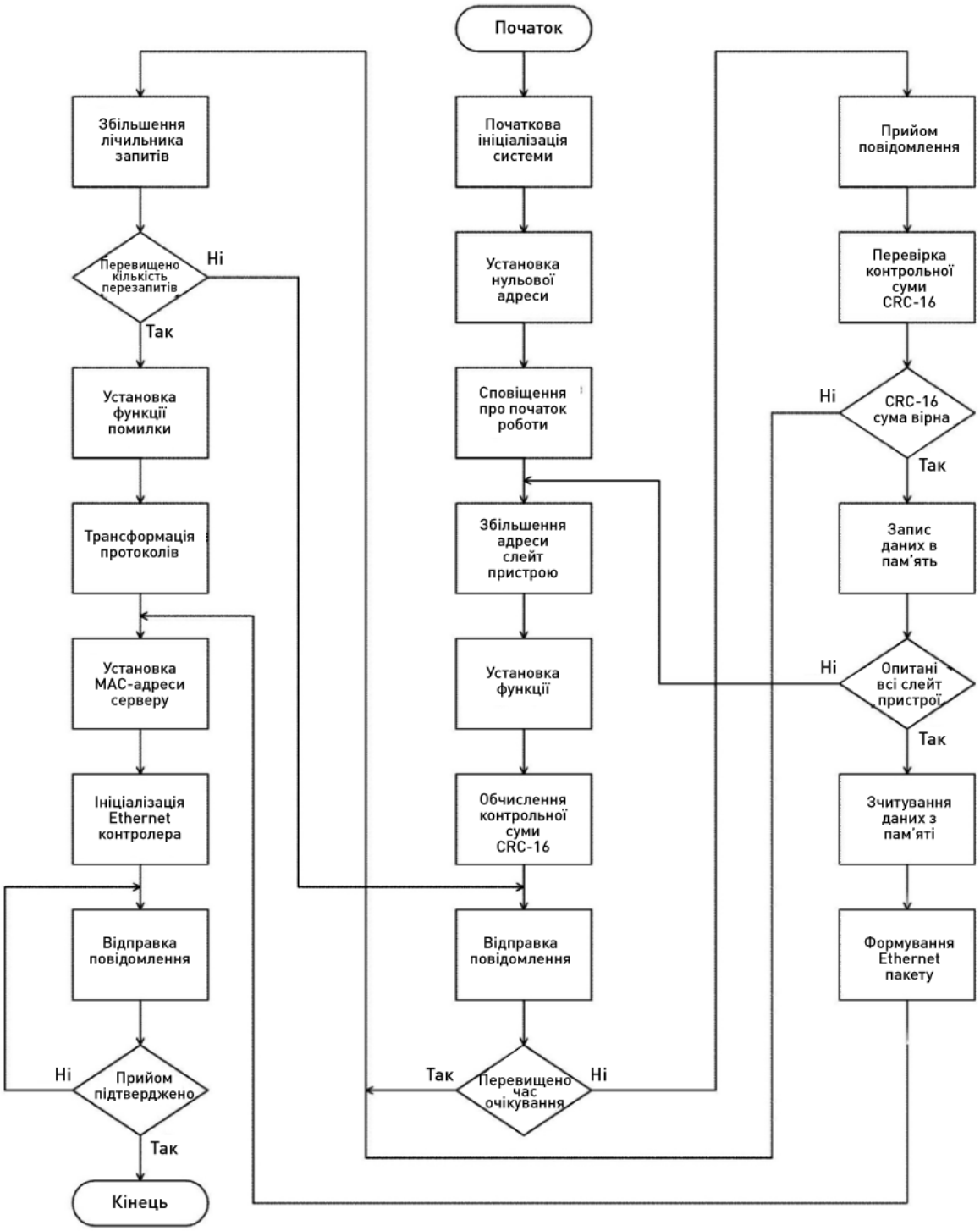

Рисунок 3.6 - Алгоритм функціонування модуля розподіленого збору даних

Структурна схема проєктованого модуля розподіленого збору даних приведена на рисунку 3.7.

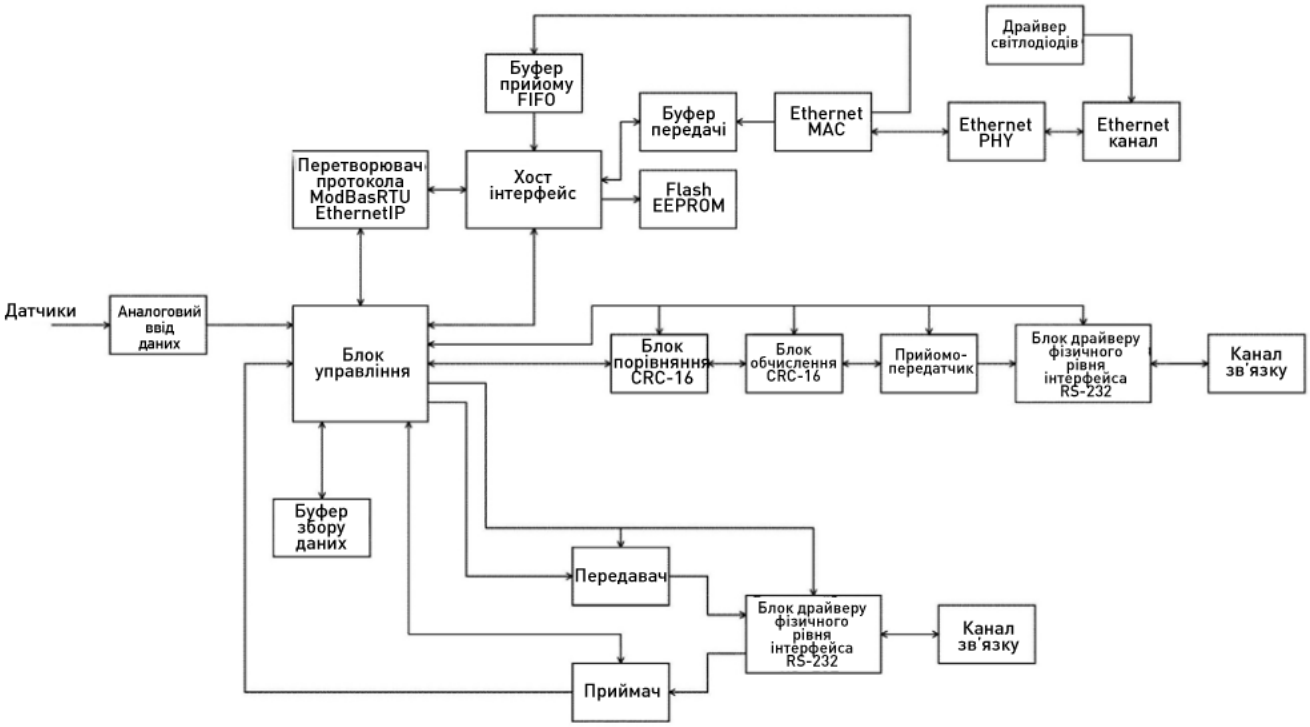

Рисунок 3.7 - Структурна схема проєктованого пристрою

Блок управління - блок призначений для управління розподіленого модуля і виконання основної програми технічного процесу.

Буфер збору даних призначений для зберігання інформації, зібраної з датчиків і переданої веденими пристроями через канал зв'язку з інтерфейсом RS-485.

Блок драйвера фізичного рівня інтерфейсу RS-485 призначений для перетворення сигналу і формування необхідної амплітуди для передачі даних в канал зв'язку.

Блок обчислення CRC16 - блок, призначений для обчислення контрольної суми надісланого і отриманого пакета за алгоритмом CRC-16 для забезпечення достовірності отриманої та надісланої інформації.

Блок порівняння CRC16 - призначений для порівняння розрахованої контрольної суми та суми, переданої через канал зв'язку.

Блок протоколу перетворення Modbus RTU/EthernetIP призначений для перетворення протоколу Modbus RTU в EthernetIP для передачі даних серверу через канал Ethernet. Інтерфейс хоста - блок, який забезпечує зв'язок між пристроєм керування та буфером прийому і передачі через паралельну 16-розрядну шину, де шина адреси та шина даних поєднані.

Буфер прийому призначений для зберігання даних, що отримуються з каналу даних, розмір буфера складає 4 кілобайти.

Буфер передачі призначений для зберігання даних перед їхньою відправкою в канал зв'язку, розмір буфера складає 4 кілобайти.

Драйвери світлодіодів - блок керування світлодіодами, які сигналізують про наявність каналу та передачу даних.

Ethernet MAC - блок збереження MAC-адреси та формування протоколу SPX.

Flash EEPROM - блок несприйнятливої до енергії пам'яті типу Flash, призначений для зберігання алгоритмів створення Ethernet-пакета та функціонування протоколів передачі даних через канал Ethernet, розмір пам'яті Flash складає 8 кілобайт.

3.2 Розробка схеми електричного функціонального проектування пристрою.

Введення даних з аналогових вхідних модулів здійснюється через 16-розрядну шину, яка з'єднується з двома 8-розрядними паралельними портами введення/виведення мікроконтролера. Сигнал готовності від зовнішнього пристрою надходить на вхід обробки зовнішнього переривання мікроконтролера, що дозволяє пристрою переключитись на зчитування даних від зовнішнього пристрою.

При передачі даних пристроєм, дані по бітах надсилаються на вивід MOSI (Master Out Slave In) до вхідного передавача DI драйвера RS-485. При цьому драйвер RS-485 має бути встановлений у режим передавача і вимкнений з режиму приймача, щоб уникнути колізій. Для цього на входах DE та RE драйвера RS-485 подається логічна одиниця (1), в іншому випадку вхід передавача знаходиться в стані високого імпедансу. Формат передачі даних з мікроконтролера та швидкість передачі даних визначаються програмно.

При прийманні даних пристроєм, дані по бітах отримуються на вхід MISO (Master In Slave Out) з виходу приймача RO драйвера RS-485. При цьому драйвер RS-485 має бути встановлений у режим приймача і вимкнений з режиму передавача. Для цього необхідно подати логічний нуль (0) на входах DE та RE драйвера RS-485.

Канал зв'язку інтерфейсу RS-485 представлений у вигляді двохпровідної лінії зв'язку в режимі півдуплексу.

Для реалізації послідовного інтерфейсу RS-232 використовуються драйвери інтерфейсу RS-232 в інтегральному виконанні. Зазвичай, драйвери інтерфейсу RS-232 мають послідовний вхід передавача та вихід приймача, які з'єднані з

мікроконтролером. Вихід передавача та вхід приймача підключаються до каналу зв'язку через COM-порт.

Обмін даними між мікроконтролером і драйвером інтерфейсу RS-232 здійснюється за допомогою універсального синхронного/асинхронного приймача-передавача (USART). Цей модуль приймача-передавача забезпечує повнодуплексний обмін даними по послідовному каналу, при цьому швидкість передачі даних може варіюватися у широких межах. Довжина передачі може становити від 5 до 9 бітів. У всіх модулях обов'язково присутні схеми контролю та формування біта парності.

Модулі USART, реалізовані в мікроконтролерах, можуть виявляти такі непередбачувані ситуації:

- переповнення буфера;
- помилка кадрування;
- невірний старт-біт.

Для зменшення ймовірності виникнення збоїв, у модулях реалізована функція фільтрації.

При передачі даних пристроєм, дані по бітах надходять через вихід TxDn на вхід T_IN драйвера інтерфейсу RS-232. При прийманні даних, дані по бітах надходять з виходу R_OUT приймача і подаються на вхід RxDn мікроконтролера.

Передаваний кадр повідомлення починається зі старт-біту, за яким слідує молодший біт слова. Після старшого біта слова, дані слідують одним або двома стоп-бітами. Якщо включено схему формування біта парності, він включається між старшим бітом слова даних та першим стоп-бітом. Формат кадру зображено на рисунку 3.8.

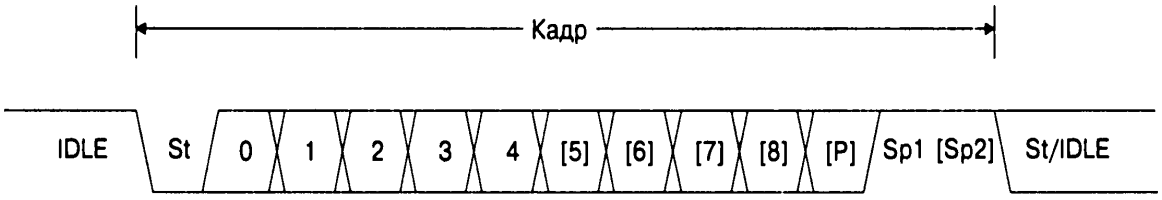

Рисунок 3.8 - Формат кадру

ST - старт-біт, завжди дорівнює 0;

[n] - біти слова даних;

P - біт парності;

Sp1 - стоп-біт, завжди дорівнює 1;

IDLE - немає обміну по лінії RxDn або TxDn, повинно бути 1.

Проектований пристрій підключається до зовнішнього пристрою за допомогою інтерфейсу RS-232 через комутаційний порт (COM-порт). У даному пристрої використовуються два 9-контактних роз'єми DB-9, які підключені в режимі мінімального нуль-модемного кабелю (рисунок 3.9) і повного нуль-модемного кабелю (рисунок 3.10).

Мінімальний нуль-модемний кабель використовується для з'єднання пристроїв по інтерфейсу RS-232, повний нуль-модемний кабель використовується для підключення пристрою до модему.

Сигнали 9-контактного роз'єму та їх описи наведені в таблиці 3.4.

Зв'язок модуля розподіленого збору даних з віддаленим сервером здійснюється через Ethernet-канал з системою 10Base-T. Система 10Base-T має менші витрати на установку порівняно з іншими системами і використовує скручену пару. Кожний вузол підключається до центрального хаба двома парами (одна пара для прийому, а друга для передачі).

Хаб може бути на відстані до 100 метрів.

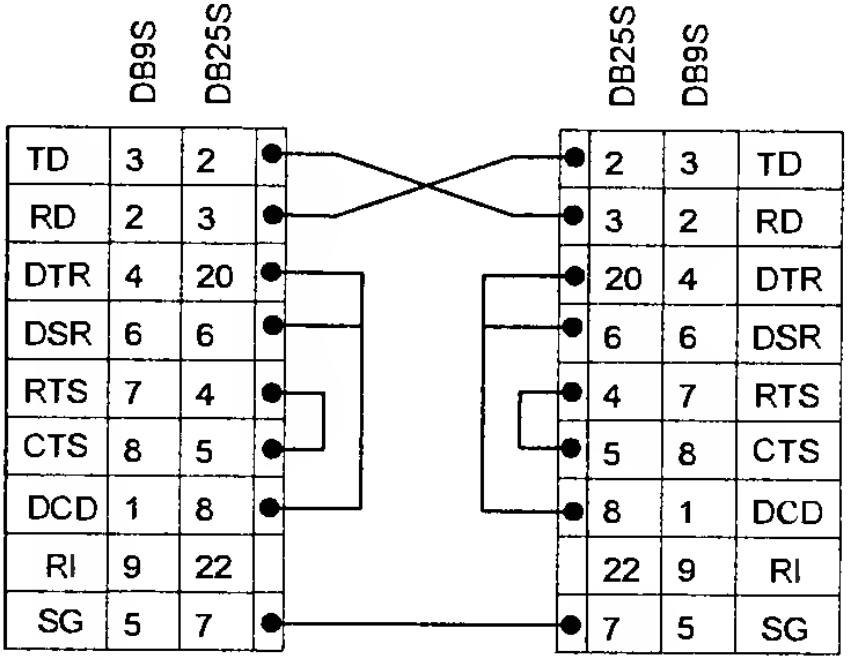

Рисунок 3.9 – Мінімальний нуль-модемний кабель

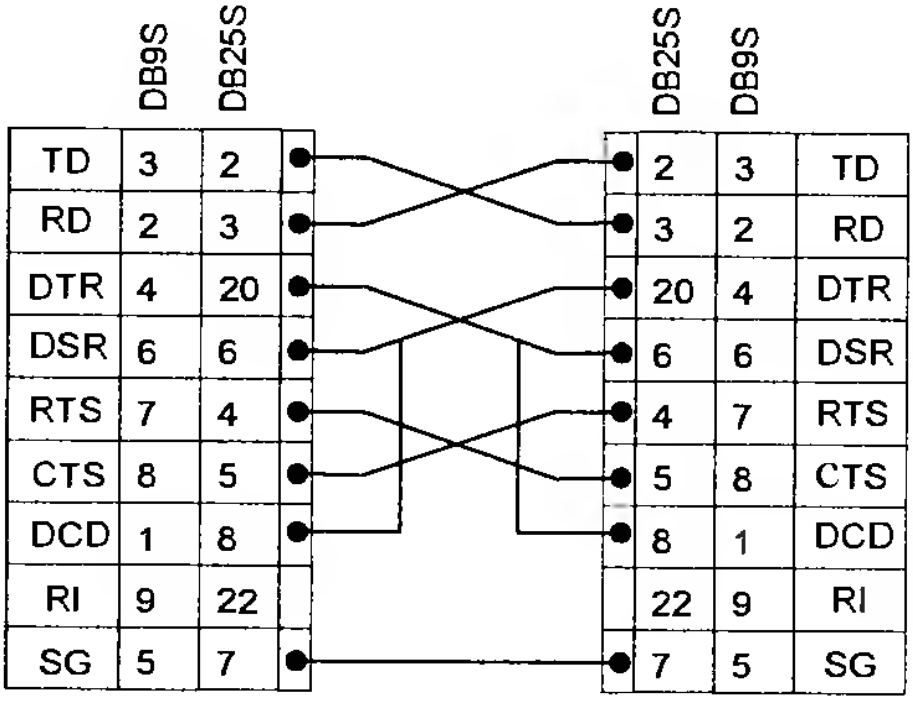

Рисунок 3.10 – Повний нуль-модемний кабель

Таблиця 3.4 – Сигнали 9-штекрного роз’єму та їх опис

Контакт DB-9	Позначення	Опис	Абревіатура	I/O
1	CF	Виявлення несучої	DCD	I
2	BB	Прийняті дані	RxD (RD)	I
3	BA	Передані дані	TxD (TD)	O
4	CD	Готовність	DTR	O
5	AB	Сигнальне заземлення	SG	I/O
6	CC	Готовність DCE	DSR	I/O
7	CACJ	Запит передачі готовність до прийому	RTS	O
8	CB	Готовність передачі	CTS	O
9	CE	Індикатор виклику	RI	I

Підключення до інтерфейсних плат здійснюється за допомогою модульних 8-контактних роз'ємів RJ-45, як показано на рис. 3.11.

Хаб забезпечує роботу шини між вузлами і може мати роз'єми AUI для товстого Ethernet, тонкого Ethernet або оптоволоконних передавачів-приймачів.

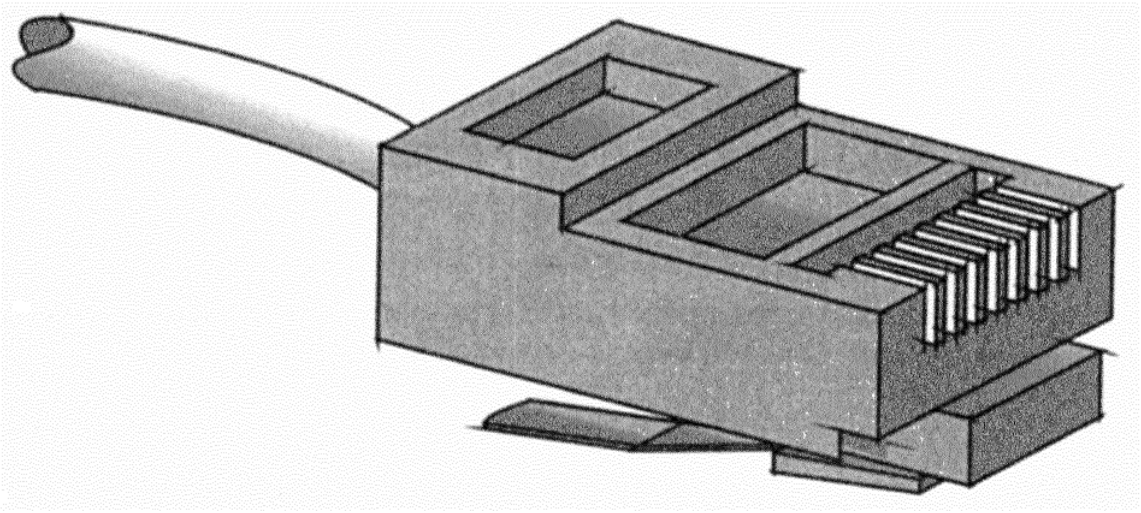

Рисунок 3.11 – Роз'єм RJ-45 для витих пар

3.3 Вибір елементної бази. Мікроконтролери AVR від фірми Atmel є 8-бітними мікроконтролерами, призначеними для використання в апаратурі зв'язку та вимірювальній апаратурі. Вони виготовлені за технологією малого споживання КМОП, яка в поєднанні з вдосконаленою RISC-архітектурою дозволяє досягти найкращого співвідношення вартості/швидкодії/енергоспоживання.

До особливостей мікроконтролерів сімейства Mega AVR можна віднести:

FLASH-пам'ять програм об'ємом від 8 до 256 Кбайт (кількість циклів стирання/запису не менше 10 000);

оперативна пам'ять (статичний ОЗП) об'ємом від 512 байт до 8 Кбайт;

пам'ять даних на основі ЕСППЗУ (EEPROM) об'ємом від 256 байт до 4 Кбайт (кількість циклів стирання/запису не менше 100 000);

можливість захисту від читання та модифікації пам'яті програм та даних;

можливість програмування безпосередньо в системі через послідовні інтерфейси SPI та JTAG;

можливість самопрограмування;

наявність декількох режимів зниженого енергоспоживання;

наявність детектора зниженого напруги живлення (Brown-Out Detector-BOD);

можливість програмного зниження частоти тактового генератора.

Для даного пристрою доцільно використовувати мікроконтролер моделі ATmega1281V-12AI/I. Даний мікроконтролер містить послідовний периферійний інтерфейс SPI, універсальний синхронний/асинхронний приймач-передавач (USART) та має достатню обчислювальну потужністю. Параметри даного мікроконтроллера наведені в табиці 3.5.[6]

Таблиця 3.5 – Параметри мікроконтроллера ATmega128

Модель	Пам'ять програм (FLASH) [Кбайт]	Пам'ять даних (ОЗУ) [байт]	Пам'ять даних (EEPROM) [байт]	Кількість контактів входов/виходів	Напруга живлення [В]	Тактова частота [МГц]
ATmega128	128	4K	4K	53	4,5…5,5	0…16
ATmega128L					2,7…5,5	0...8

Розташування виводів мікроконтролеру ATmega128 у корпусі TQFP-64 і MLF-64 наведено на рис. 3.12.

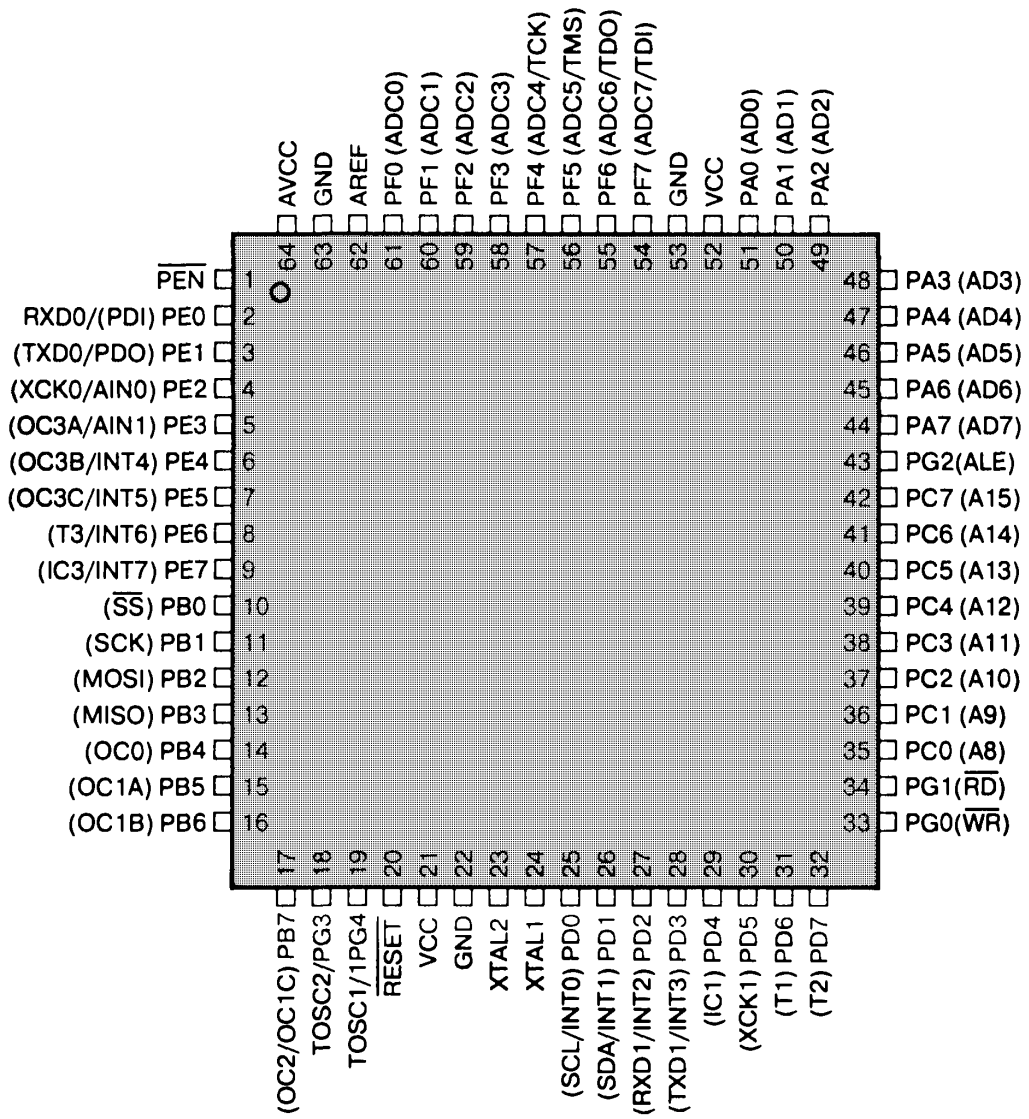

Рисунок 3.12 – Розташування моделі ATmega128

Призначення, тип і нумерація виводів мікроконтролера ATmega128 у корпусі TQFP-64 і MLF-64 наведені в таблицях 3.6.

У даному мікроконтролері реалізований послідовний периферійний інтерфейс SPI (Serial Peripheral Interface), призначений для обміну даними між мікроконтролером та різними периферійними пристроями, такими як цифрові потенціометри, ЦАП/АЦП, FLASH-ПЗУ та інші. Через цей інтерфейс також може здійснюватися обмін даними між кількома мікроконтролерами AVR.

Таблиця 3.6 – Опис виводів моделі ATmega128

Позначення	Номер виводу	Тип виводу	Опис
XTAL1	24	I	Вхід тактового генератора
XTAL2	23	O	Вихід тактового генератора
$\overline{RESET}$	20	I	Вхід скидання
Порт А. 8-розрядний двунапрямлений порт введення/виведення з внутрішніми підтягуючими резисторами.			
PAO	51	I/O	0-й біт порту А
PAl	50	I/O	1-й біт порту А
PA2	49	I/O	2-й біт порту А
PA3	48	I/O	3-й біт порту А
PA4	47	I/O	4-й біт порту А
PA5	46	I/O	5-й біт порту А
PA6	45	I/O	6-й біт порту А
PA7	44	I/O	7-й біт порту А
Порт С. 8-розрядний двунапрямлений порт введення/виведення з внутрішніми підтягуючими резисторами.			
PCO	35	I/O	0-й біт порту С
PCl	36	I/O	1-й біт порту С
PC2	37	I/O	2-й біт порту С
PC3	38	I/O	3-й біт порту С
PC4	39	I/O	4-й біт порту С
PC5	40	I/O	5-й біт порту С
PC6	41	I/O	6-й біт порту С
PC7	42	I/O	7-й біт порту С
Виводи послідовного периферійного інтерфейсу SPI.			
$\overline{SS}$	10	I/O	Вибір Slave-пристрою на шині
SCK	11	I/O	Вихід (Master) або вхід (Slave) тактового сигналу модуля SPI

Продовження таблиці 3.6

Позначення	Номер виводу	Тип виводу	Опис
MOSI	12	I/O	Вихід (Master) або вхід (Slave) даних модуля SPI
MISO	13	I/O	Вхід (Master) або вихід (Slave) даних модуля SPI
Універсальний синхронний/асинхронний приймач-передавач (USART).			
RXD1	27	I/O	Вхід USART1
TXD1	28	I/O	Вихід USART1
XCK1	30	I/O	Вхід/вихід зовнішнього тактового сигналу USART1
RXD0	2	I/O	Вхід USART0
TXD0	3	I/O	Вихід USART0
XCK0	4	I/O	Вхід/вихід зовнішнього тактового сигналу USART0
INT7	9	I/O	Вхід зовнішнього переривання
Порт F. 8-бітний двонапрямлений порт введення/виведення з внутрішніми підтягуючими резисторами.			
PFO	61	I/O	0-й біт порту F
PF1	60	I/O	1-й біт порту F
PF2	59	I/O	2-й біт порту F
PF3	58	I/O	3-й біт порту F
PF4	59	I/O	4-й біт порту F
PF5	60	I/O	5-й біт порту F
PF6	61	I/O	6-й біт порту F
PF7	62	I/O	7-й біт порту F

Структурна схема модуля SPI наведена на рисунку 3.11.

Часові діаграми передачі даних за допомогою послідовного периферійного інтерфейсу наведені на рисунку 3.14.

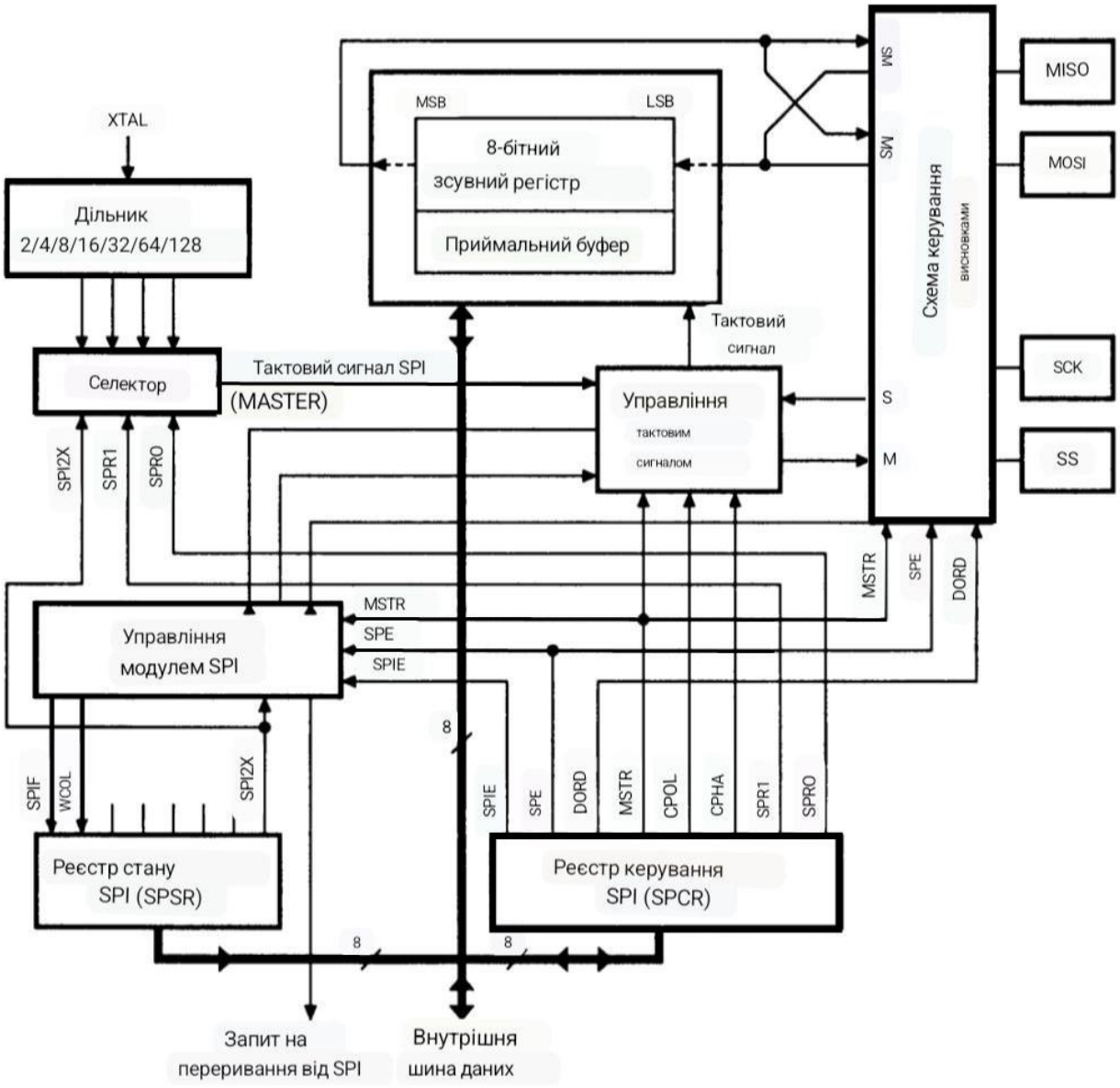

Рисунок 3.13 - Структурна схема модуля SPI

Частота тактового сигналу SCK і відповідно швидкість передачі даних по інтерфейсу визначаються програмно і встановлюються бітами SPR1, SPR0 регістру SPCR та бітом SPI2X регістру SPSR. Звичайно, йдеться про мікроконтролер, що працює в режимі Master, оскільки саме він є джерелом тактового сигналу. Задання тактової частоти SCK наведено в таблиці 3.7.

Отже, частота тактового сигналу SCK послідовного інтерфейсу SPI при тактовій частоті мікроконтролера 12 МГц змінюється від 93 КГц до 6 МГц.

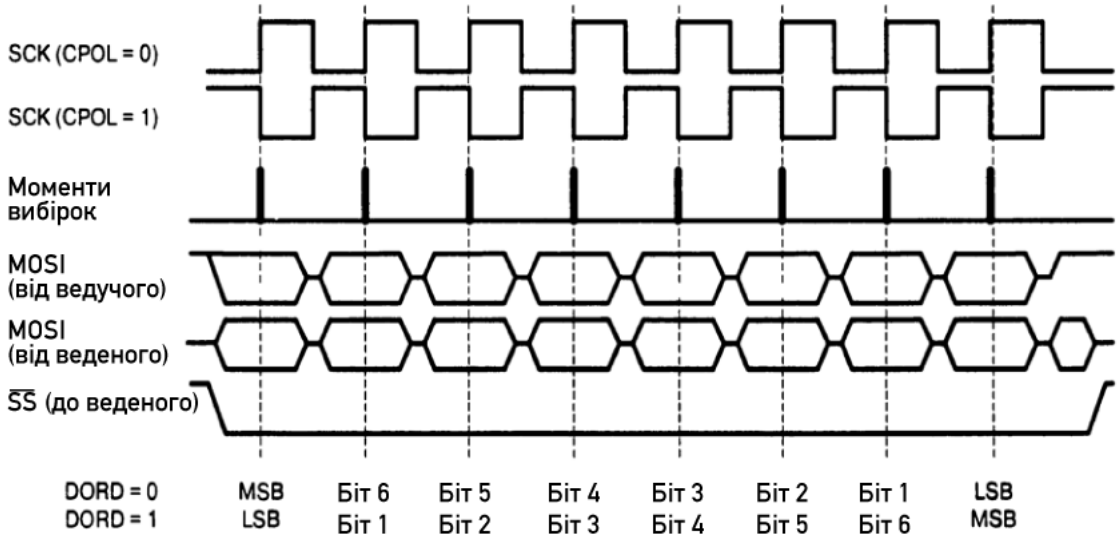

Рисунок 3.14 - Часові діаграми передачі даних за допомогою SPI

Таблиця 3.7 - Налаштування частоти тактового сигналу SCK

SPI2X	SPR1	SPR0	Частота сигналу SCK
0	0	0	$f_{CLK}/4$
0	0	1	$f_{CLK}/16$
0	1	0	$f_{CLK}/64$
0	1	1	$f_{CLK}/128$
1	0	0	$f_{CLK}/2$
1	0	1	$f_{CLK}/8$
1	1	0	$f_{CLK}/32$
1	1	1	$f_{CLK}/64$

де f_{CLK} – тактова частота мікроконтроллера.

У всіх мікроконтролерах сімейства Mega вбудований модуль універсального синхронного/асинхронного приймача-передавача (USART). Модель ATmega1281V-12AI/I включає два модулі універсального синхроннго/

асинхронного приймача-передавача, які забезпечують повнодуплексний обмін даними по послідовному каналу.

Структурна схема одного модуля USART наведена на рисунку 3.15.

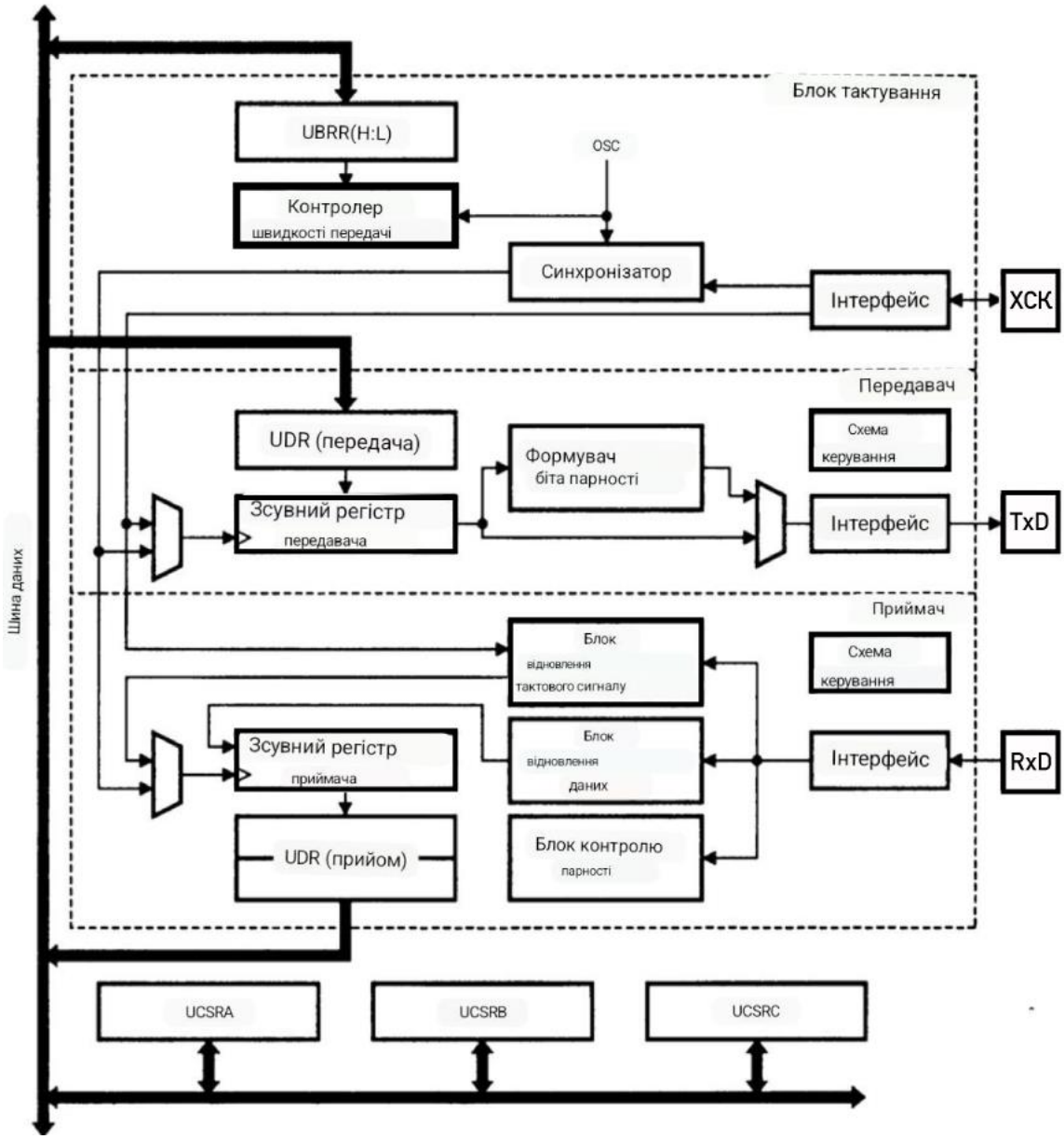

Рисунок 3.15 - Структурна схема модуля USART

Швидкість обміну даними визначається за формулою 3.5:

$$BAUD = \frac{fck}{16 \cdot (UBBR + 1)}, \quad (3.5)$$

де BAUD – швидкість передачі біт/с;

fck – тактова частота мікроконтролера;

UBBR – вміст регістру керування швидкістю передачі даних. Послідовний інтерфейс передачі даних RS-485 реалізується через драйвер інтерфейсу RS-485 моделі MAX3443E від компанії MAXIM. Схематичне зображення драйвера інтерфейсу RS-485 наведено на рисунку 3.16.

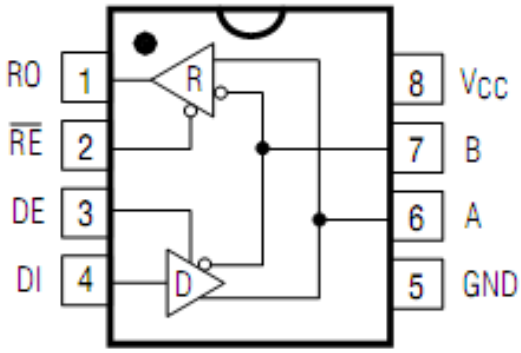

Рисунок 3.16 - Схематичне зображення MAX3443E

Драйвер інтерфейсу RS-485 моделі MAX3443E має вбудовану захист від перепадів напруги до 60 В та функцію гарячого підключення приймача і передавача до каналу зв'язку. Він гарантовано забезпечує передачу даних зі швидкістю до 10 Мбіт/с. Драйвери MAX3443E можуть бути підключені до загальної шини у кількості до 128 штук. Організація локальної мережі з використанням драйвера MAX3443E показана на рисунку 3.17.

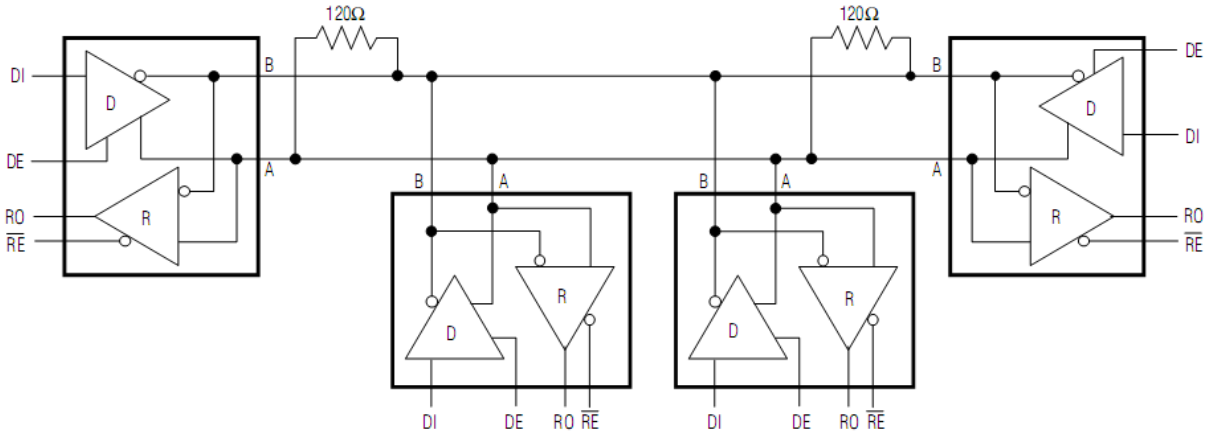

Рисунок 3.17 - Організація локальної мережі з використанням драйвера MAX3443E

Паралельно до драйверів MAX3443E підключаються два термінальних резистори з опором 120 Ом. Призначення, тип і нумерація виводів драйвера інтерфейсу RS-485 моделі MAX3443E наведені в таблиці 3.8.

Таблиця 3.8 - Призначення, тип і нумерація виводів драйвера інтерфейсу RS-485 MAX3443E

Позначення	Номер виводу	Опис
RO	1	Вихід приймача
$\overline{RE}$	2	Включення приймача
DE	3	Включення передавача
DI	4	Вхід передавача
GND	5	Заземлення
A	6	Неінвертуючий вихід приймача-передавача
B	7	Інвертуючий вихід приймача-передавача
Vcc	8	Живлення

Схема зв'язку драйвера інтерфейсу RS485с підключається за типовою схемою, яка наведена в описі розробника.

Для реалізації послідовного інтерфейсу передачі даних RS-232с використовуються драйвери інтерфейсу RS-232с марки MAX1336E від компанії MAXIM. Схематичне зображення драйвера інтерфейсу RS-485 наведено на рисунку 3.18.

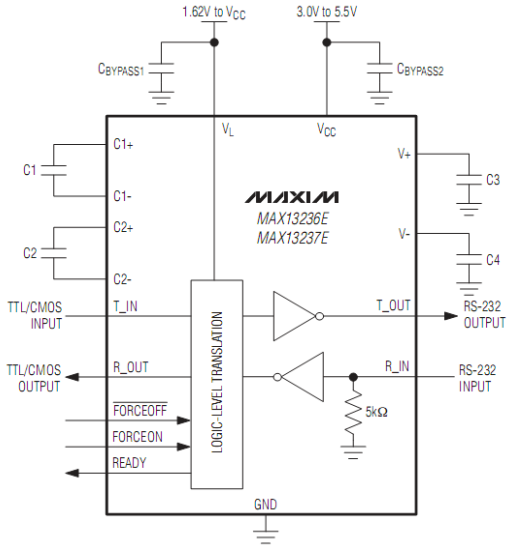

Рисунок 3.18 – Схематичне зображення MAX1326

Таблиця 3.9 – Призначення, тип і нумерація виводів драйвера інтерфейсу RS-485 MAX13236E

Позначення	Номер виводу	Опис
READY	14	Готовність до передачі даних
C1+	16	Позитивний вивід для конденсатора
V+	15	Позитивне живлення плати
C1-	1	Від'ємний вивід для конденсатора
C2+	2	Позитивне значення для конденсатора
C2-	3	Від'ємний вивід для конденсатора
V-	4	Від'ємне живлення плати
RIN	5	Вхід приймача RS-232c
ROUT	6	Вихід приймача
VL	7	Виводи живлення, що визначають логіку
TIN	8	Вхід передавача
FORCEON	9	Увімкнення захисту
TOUT	10	Вихід передавача RS-232c
GND	11	Заземлення
VCC	12	Живлення
$\overline{\text{FORCEOFF}}$	13	Вимкнення захисту

Драйвер інтерфейсу RS232c підключається за типовою схемою, наведеною в описі розробника[8].

Для реалізації каналу Ethernet використовується контролер мережі CP2200. CP220x - це компактні однокристальні контролери Ethernet IEEE802.3 10-BaseT, які призначені для інтеграції Ethernet-інтерфейсу в різноманітні системи. Ці IC реалізують функції стеку мережевих протоколів на рівні MAC і PHY та забезпечують доступ до Ethernet-мереж для будь-яких мікроконтролерів або хост-пристроїв, які мають принаймні 11 вільних вводу-виводу ліній. CP2200 має корпус TQFP з 48 виводами розміром 9x9 мм.

CP2201 відрізняється від CP2200 типом корпусу (28-виводний QFN розміром 5x5 мм), реалізацією драйвера світлодіодів і неможливістю підтримки немультиплексованого режиму взаємодії зовнішньою хост-системою. Структурна схема контролера CP2200 показана на рис. 3.19.

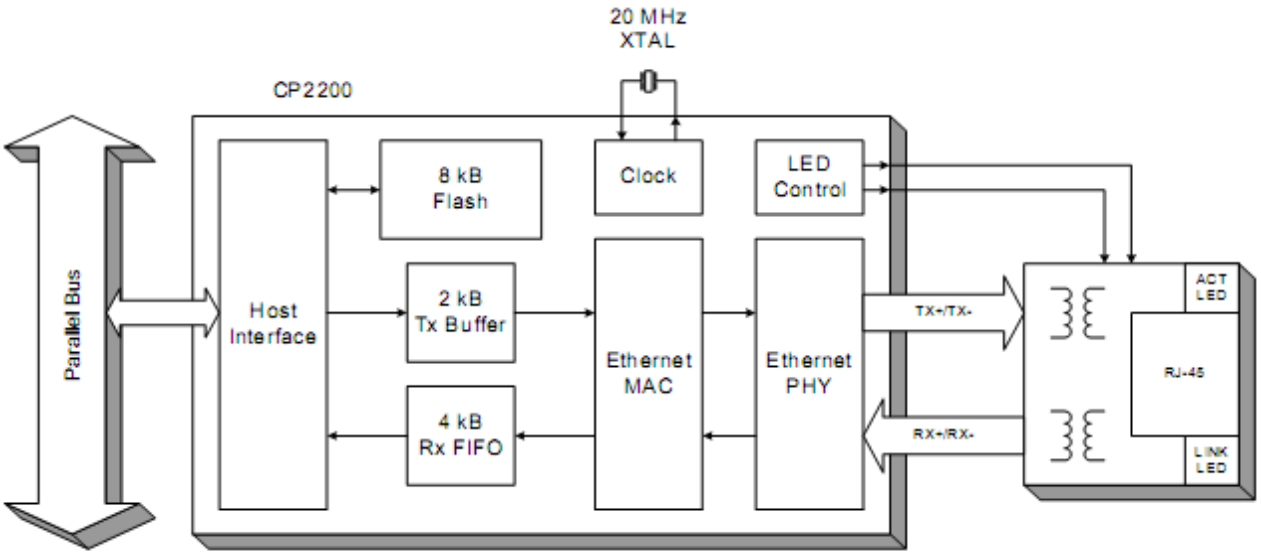

Рисунок 3.19 – Структурна схема CP2200

Як видно з рис. 1, CP2200 містить паралельний хост-інтерфейс, енергонезалежну пам'ять типу Flash EEPROM об'ємом 8 кбайт, 2-килобайтний буфер передачі, 4-килобайтний буфер прийому типу FIFO, контролери Ethernet на рівні MAC і PHY, а також тактовий генератор і драйвер світлодіодів. Модуль паралельного хост-інтерфейсу дозволяє передавати дані зі швидкістю до 30 Мбіт/с, працює з шинами формату Intel (Motorola) і підтримує 8-розрядні (мультиплексований і немультиплексований) режими взаємодії зовнішньою хост-системою.

Даний модуль забезпечує доступ до реєстрів спеціального призначення, через які здійснюється прийом і передача даних, а також керування функціонуванням і контроль стану всіх модулів CP2200. Крім того, хост-інтерфейс може генерувати сигнал переривання, який може "пробуджувати" хост-процесор з режиму "сну" при отриманні пакетів або при підключенні CP2200 до мережі (Wake-on-LAN).

Загалом підтримується 14 подій, які можуть викликати переривання. Якщо хост-процесор не має окремого вивода для підключення сигналу переривання, то він може періодично опитувати регістри стану переривань CP2200 для виявлення відповідних подій.

За допомогою хеш-таблиці можна встановити діапазон MAC-адрес, які будуть дозволені для запису в буфер прийому; пакети з адресами, які не потрапляють в встановлений діапазон, будуть відкидатися.

Серед інших функцій Ethernet-інтерфейсу, реалізованих вбудованими контролерами PHY і MAC на автоматичному рівні, варто виділити наступні:

- розширення коротких пакетів до мінімальної довжини;
- обчислення і перевірка контрольної суми (CRC);
- перевірка довжини пакета;
- повна сумісність зі стандартами 100/1000 Base-T;
- підтримка стандартів Plug-and-Play (auto-negotiations, auto-polarity);
- дуплексний і напівдуплексний режими роботи;
- виявлення колізій з автоматичним повтором передачі;
- тестування і налагодження в режимі зворотного зв'язку (можливі два режими: з замиканням каналу передачі на рівні PHY або на рівні MAC);
- перевірка наявності зв'язку з мережею;
- виявлення і приглушення шумів при прийомі;
- відсікання некоректних пакетів при передачі.

Також варто зазначити, що буфери прийому та передачі Ethernet-інтерфейсу відображені в одному і тому ж діапазоні адресного простору, і доступ до них здійснюється за допомогою одного й того ж вказівника адреси. Однак регістри даних (для операцій читання-запису) використовуються різні.

Крім буферів прийому та передачі в CP2200 є ще один блок пам'яті - вбудована флеш-пам'ять об'ємом 8 кБ. Вона розташована в окремому адресному просторі, і доступ до неї здійснюється за допомогою власних вказівника адреси та регістру даних. Ця пам'ять може використовуватись для зберігання користувацьких констант, веб-серверу, а також як незалежна від енергії пам'ять загального призначення. При виготовленні ІС у шість останніх комірок цієї пам'яті на заводі записується унікальна 48-розрядна MAC-адреса, що в більшості випадків дозволяє не присвоювати серійні номери розроблюваним пристроям.

Флеш-пам'ять стирається секторами розміром 512 байтів та програмується по одному байту. Управління операціями запису-стирання флеш-пам'яті здійснюється автоматично апаратними засобами і не перешкоджає роботі паралельного хост-інтерфейсу. Хост-система отримує повідомлення про завершення доступу до флеш-пам'яті за допомогою переривань.

Для тактування CP2200 можна використовувати або вбудований тактовий генератор (у цьому випадку потрібно підключити зовнішній кварцовий резонатор 20 МГц), або зовнішній тактовий сигнал. При відсутності тактових імпульсів з яких-небудь причин буде згенерований сигнал скиду тривалістю приблизно 1 мс, в тому числі на зовнішньому виводі /RST, який можна використовувати для скидання зовнішніх модулів.

Сигнал скиду також генерується при відсутності живлення або зниженні його порогового значення.

Типові схеми підключення Ethernet-контролерів CP2200, рекомендовані компанією Silicon Laboratories, наведені на рис. 3.20.

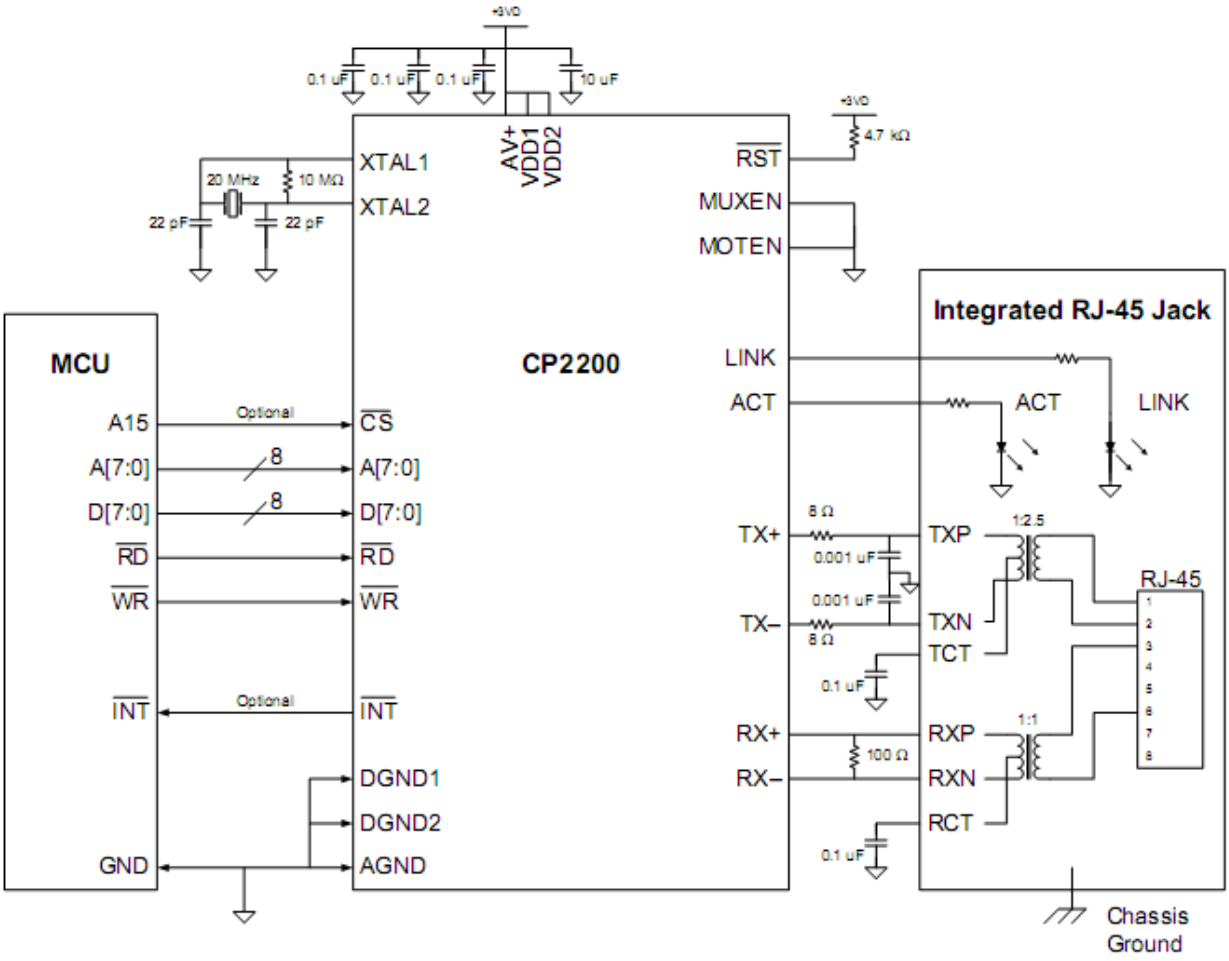

Рисунок 3.20 - Типова схема включення CP2200

Значення виводів мікросхеми CP2200 приведені в таб. 3.10.

Таблиця 3.10 – Значення виводів мікросхеми CP2200

Позначення	Номер виводу	Тип виводу	Опис
AV+	5	Живлення	Аналогове живлення +5В
AGND	4		Аналогова земля
VDD1	13	Живлення	Цифрове живлення +5В

DGND1	14		Цифрова земля
VDD2	30	Жив-лення	Цифрове живлення +5В
DGND2	31		Цифрова земля
RST	15	D I/O	Перезавантаження пристрою.
LINK	3	D Out	Сигнал з'єднання.
ACT	2	D Out	Сигнал активності
XTAL1	46	A In	Вхід тактового генератора
XTAL2	45	A Out	Вихід тактового генератора
TX+	9	A Out	10Base-T передатчик
TX–	10	A Out	10Base-T передатчик
RX+	7	A In	10Base-T приймач
RX–	6	A In	10Base-T приймач
MOTEN	43	D In	Включення шини моторола.
MUXEN	44	D In	Включення шини інтел
INT	42	D In	Сигнал переривання.
CS	41	D In	Вибір пристрою.
RD	39	D In	Читання.
WR	40	D In	Запис
ALE	32	D In	Вибір шини
A0-A7	27,28,29,32, 33,34,37,38	D In	Шина адреси
D0-D7	16-23	D I/O	Шина даних

3.4 Розробка та розрахунок принципових електричних схем вузлів та блоків пристрою.

Розрахунок продуктивності Ethernet. Припустимо, що в черзі на передачу пакета знаходяться N вузлів. Це число представляє повне навантаження системи і припускається, що воно більше 1.

У Ethernet-системі існує конфліктний інтервал, або час, протягом якого вузли намагаються провести передачу, перш ніж їм буде надано гарантію на управління шиною. Протягом цього часу, що дорівнює тривалості 512 бітів, можуть виникати конфлікти через затримку поширення сигналу по мережі.

Якщо припустити, що вузол має ймовірність передачі в будь-який проміжок часу, рівну 1/N, то ймовірність затримки складає [1 - 1/N].

Ймовірність (P) того, що рівно один вузол спробує передавати у цьому часовому інтервалі і успішно отримає контроль над шиною, дорівнює:

$$P = \left[1 - \left\{\frac{1}{N}\right\}\right]^{N-1}, \tag{1.8}$$

Де P - ймовірність того, що шина буде вільною;

N - кількість вузлів.

Тобто ймовірність того, що очікування не буде, перед тим як буде отримано контроль над шиною, дорівнює P.

Ймовірність очікування лише одного часового інтервалу становить $P(1-P)$, а ймовірність очікування N часових інтервалів - $P(1-P)^N$. Це геометрична послідовність з середнім значенням (1 - P)/P.

Отже, середня кількість часових інтервалів, які вузол повинен очікувати, перед тим як він отримає доступ до шини, складає:

$$S = \frac{(1-P)}{P}, \tag{3.6}$$

де S - середній час очікування інтервалів.

Для N=10 ймовірність вільної шини при використанні пристрою складає:

$$P = \left[1 - \left\{\frac{1}{N}\right\}\right]^{N-1} = \left[1 - \left\{\frac{1}{10}\right\}\right]^{10-1} = 0,387, \tag{3.7}$$

Тоді середня кількість очікуваних інтервалів:

$$S = \frac{(1-P)}{P} = \frac{(1-0,387)}{0,387} = 1,58, \tag{3. 8}$$

Ефективність Е визначається відношенням відсотку часу, коли мережа передає корисні дані. Кожен пакет має максимум 192 біта (витрати), які складаються з:

7 байтів преамбули,

1 байту SFD,

6 байтів адреси призначення,

6 байтів адреси джерела,

2 байтів індикатора довжини,

4 байтів CRC.

Максимальний розмір пакета становить 1526 байтів або 12208 бітів, а мінімальний розмір пакета - 72 байти або 576 бітів. Між кожним кадром є проміжок, тривалість якого дорівнює 96 бітам, що при швидкості 20 Мбіт/с складає 4,8 мікросекунди.

Ефективність Е для мінімального і максимального пакета розраховується за формулою 3.9:

$$Å = \frac{Dp}{(Dp + O + T)}, \qquad (3.9)$$

де Dp - корисні дані;

O - очікування;

T - час затримки.

Для кадра максимального розміру:

$$Å_{MAX} = \frac{6104}{(6104 + 256 + 48)} = 95,2\% , \qquad (3.10)$$

Для мінімального розміру:

$$Å_{MIN} = \frac{288}{(288 + 256 + 48)} = 48,6\%, \qquad (3.11)$$

Зменшення ефективності викликається зростаючою частиною часу фіксованих витрат та інтервалу між кадрами.

3.5 Розробка програмного забезпечення проектованого пристрою.

Розробка алгоритму обчислення контрольної суми. Для обчислення контрольної суми методом циклічного надлишкового коду передавач виконує алгоритм:

- бере повідомлення, яке складається з послідовності бітів;

- додає 16 нулів з правого боку повідомлення;

- ділить за модулем 2 на наступне число, дільник (або породжуючий поліном), наприклад, 11000000000000101 (CRC-16), і отримує результат у вигляді залишку CRC;

- додає залишок у формі 16-розрядного числа до послідовності початкового повідомлення (тобто замінює 16 нулів на 16-розрядний залишок) і передає це на приймач.

Після отримання пакету даних приймач виконує такі операції:

- бере повне повідомлення плюс біти контрольної суми CRC і ділить на те саме число, яке використовував передавач;

- якщо помилок немає, то результатом залишок буде складатися з нулів (згідно з рівнянням 1.2 розділу 1);

- якщо є помилки, то залишок не буде дорівнювати нулю.

Загальний алгоритм обчислення контрольної суми CRC16 для передавача показано на рисунку 3.21.[7]

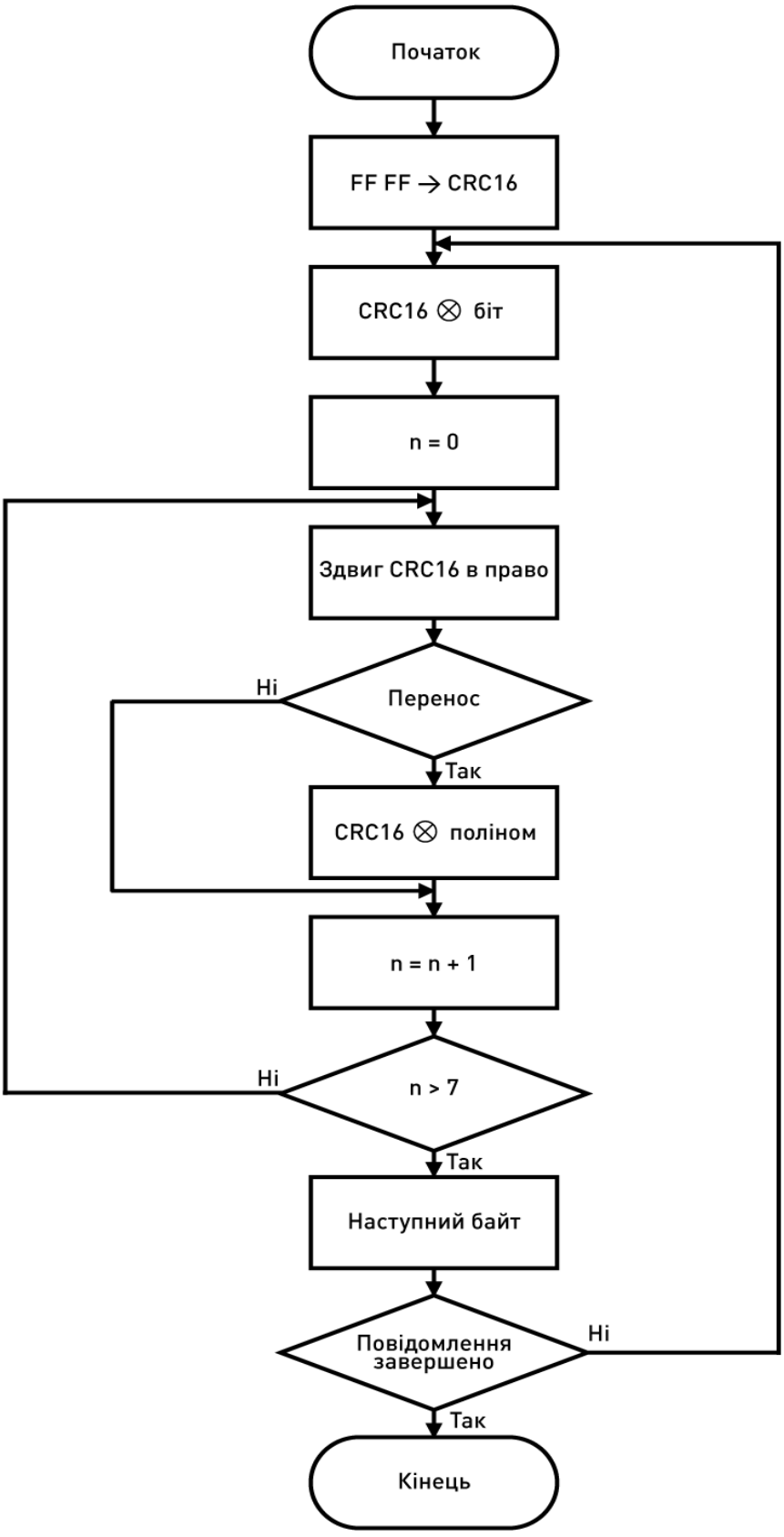

Рисунок 3.21 - Алгоритм обчислення контрольної суми

ВИСНОВКИ

У рамках даної кваліфікаційної роботи було досліджено та проаналізовано пристрій розподілу інформації на основі послідовних інтерфейсів. Головною метою роботи було розкрити принципи роботи та можливості таких пристроїв, а також дослідити їх застосування у різних галузях.

У процесі дослідження було з'ясовано, що послідовні інтерфейси є одними з найпоширеніших та найстаріших способів передачі даних між пристроями. Найбільш відомі послідовні інтерфейси включають UART (Універсальний асинхронний приймач-передавач), SPI (Послідовний інтерфейс зі зсувним регістром) та I2C (Шина з двома проводами).

Дослідження показало, що пристрої розподілу інформації на основі послідовних інтерфейсів використовуються в різних сферах, включаючи мікроконтролерні системи, комп'ютерні мережі, інформаційні технології та інші. Вони дозволяють ефективно передавати дані між пристроями, забезпечуючи надійність і швидкість передачі.

Крім того, в процесі дослідження було виявлено деякі виклики та обмеження, пов'язані з використанням послідовних інтерфейсів, такі як обмежена швидкість передачі даних та обмежена довжина кабелю. Однак, зрозуміння цих обмежень дозволяє розробникам ефективно використовувати ці пристрої та враховувати їх у своїх проектах.

Отже, на основі проведеного дослідження можна зробити висновок, що пристрої розподілу інформації на основі послідовних інтерфейсів є важливими компонентами сучасних технологій. Вони забезпечують надійну та швидку передачу даних і використовуються у багатьох галузях. Подальше розвиток та вдосконалення цих пристроїв сприятимуть розширенню їх застосування і покращенню ефективності взаємодії між пристроями.

СПИСОК ЛІТЕРАТУРИ

1. Ян Аксельсон USB Complete: The Developer's Guide, Медісон, Вісконсін 2009, 535 с.
2. Макс Рей The RS232 standard. A Tutorial with Signal Names and Definitions. The RS232 standard. A Tutorial with Signal Names and Definitions. URL: https://www.camiresearch.com/Data_Com_Basics/RS232_standard.html 2008, 311 с.
3. О.Ю.Гусєв, Г.Ф.Конахович, В.І.Корнієнко, Г.В.Кузнецов, О.Ю.Пузиренко Теорія Електричного Зв’язку URL: https://tks.nau.edu.ua/wp-content/uploads/2016/10/TEORIYA-ELEKTRYCHNOGO-ZVYAZKU.pdf 2016, 331 с.
4. Конахович Г.Ф. Експлуатація телекомунікаційних систем, Київ, 2019 372 с.
5. Величко О.М. Гордієнко Т.Б. Інтелектуальні інформаційні системи: структура і застосування 2022, 728 с.
6. Й. Й. Білинський, К. В. Огороднік, М. Й. Юкиш. Електронні системи: навчальний посібник, Вінниця: ВНТУ, 2011. — 208 с.
7. Жуйков В.Я., Терещенко Т.О., Петергеря Ю.С. Електронний підручник «Мікропроцесори і мікроконтролери» - 2009
8. Терещенко Т. О., Тодоренко В.А., Батрак Л.М.,Ямненко Ю. С. Мікропроцесорні пристрої. Навчальний посібник для аспірантівспеціальності «Електроніка». - К.: НТУУ «КПІ ім. Ігоря Сікорського», 2017. 244с.