

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра електроніки і комп'ютерної техніки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи бакалавра на тему:

«Мережа передачі даних на основі інтерфейсу RS-485»

Завідувач кафедри:

Опанасюк А.С.

Керівник

дипломного проекту:

Знаменщиков Я.В.

Виконав студент

гр. ТК-71:

Зоренко Д.Ю.

Суми 2021 р.

РЕФЕРАТ

Пояснювальна записка містить: 50 сторінок тексту, 6 таблиць і 26 рисунків, 14 джерел посилання. Графічна частина роботи містить алгоритм роботи, структурну, функціональну і принципову електричну схеми.

У даній роботі пропонується мережа передачі даних на основі інтерфейсу RS-485.

У першому розділі проведений огляд літературних джерел промислових мереж передач даних, мережі RS-485, протоколу передачі даних Modbus.

Другий розділ містить розробку та обґрунтування алгоритму роботи, структурної, функціональної, принципової електричної схем. Також виконаний відбір елементної бази.

У третьому розділі проведено розробку протоколу передачі даних та лістинг коду програми.

					ЕЛІТ 6.172. 352 ПЗ			
Зм.	Лист	№ докум.	Підпис	Дата				
Розроб.		Зоренко			Мережа передачі даних на основі інтерфейсу RS-485 Пояснювальна записка	Літ.	Аркуш	Аркушів
Перевір.		Знаменщиков					3	
Т. Контр.						СумДУ, зр. ТК-71		
Н. Контр.		Гапич В.М.						
Затверд.		Опанасюк А.С.						

ЗМІСТ

ВСТУП.....	5
1 ЛІТЕРАТУРНИЙ ОГЛЯД.....	6
1.1 Промислові мережі передачі даних.....	6
1.2 Перешкоди, що виникають в дротових каналах зв'язку.....	8
1.3 Інтерфейс RS-485, його характеристики.....	11
1.4 Організація мереж на основі інтерфейсу RS-485.....	15
1.5 Організація протоколу передачі даних по мережі RS-485.....	20
1.6 Постановка задачі проектування.....	29
2 РОЗРОБЛЕННЯ ТА РОЗРАХУНКИ ЕЛЕКТРИЧНИХ СХЕМ ВУЗЛІВ І БЛОКІВ МЕРЕЖІ ПЕРЕДАЧІ ДАНИХ.....	31
2.1 Розробка алгоритму роботи.....	31
2.2 Розробка структурної схеми.....	33
2.3 Розробка функціональної схеми.....	34
2.4 Вибір елементної бази.....	35
2.5 Розробка схеми електричної принципової.....	39
3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	42
3.1 Протокол передачі даних.....	42
3.2 Лістинг коду програми.....	43
ВИСНОВОК.....	52
СПИСОК ЛІТЕРАТУРИ.....	53

ВСТУП

Організація мереж передачі даних є невід'ємною частиною створення систем передачі даних в галузі промислової автоматизації, безпеки, керування тощо. Розробники промислових систем стикаються зі складними завданнями щодо забезпечення їх надійної експлуатації в електромагнітній обстановці, здатній вивести з ладу обладнання або порушити роботу цифрових систем передачі даних. Одним із прикладів подібних систем є автоматичне керування технологічним обладнанням на автоматизованому промисловому підприємстві. Контролер, керуючий процесом, вимірює його параметри, а також параметри навколишнього середовища, і передає команди виконавчим пристроям або формує аварійні оповіщення. Промислові контролери являють собою, як правило, мікропроцесорні пристрої, архітектура яких оптимізована для вирішення завдань даного промислового підприємства. Лінії передачі даних топології «точка-точка» в таких системах схильні до сильних електромагнітних перешкод від впливу навколишнього середовища.

Вибір доцільності використання того чи іншого типу мережі проводиться в залежності від поставлених задач, вимог до кількості пристроїв підключених до мережі, максимальної швидкості передачі даних, фінансових витрат потрібних на організацію мережі. Деякі типи мереж, наприклад Ethernet, потребують складного комунікаційного обладнання, для роботи інших, таких як CAN та RS-485, достатньо однієї мікросхеми-перетворювача інтерфейсу чи вбудованої в мікроконтролер апаратної підтримки інтерфейсу.

Серед всіх існуючих інтерфейсів RS-485 є найбільш поширеним інтерфейсом фізичного рівня для реалізації мереж з послідовною передачею даних, призначених для жорстких умов експлуатації в промислових цілях і в системах автоматизованого управління будівлями. Даний стандарт послідовного інтерфейсу забезпечує обмін даними з високою швидкістю на порівняно велику відстань по одній диференційній лінії (витій парі).

На сьогоднішній день існує велика кількість професійних рішень в галузі побудови мереж та систем передачі даних, а також відкритих проєктів на Інтернет-ресурсах присвячених їх розробці. Проте важливою проблемою залишається проєктування мереж передачі даних, орієнтованих на вузьку спеціалізацію з урахуванням індивідуальних потреб користувача і специфіки використання, тому тематика даної роботи є актуальною, потребує детального вивчення, нових ідей і рішень.

					ЕЛТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		5

1 ЛІТЕРАТУРНИЙ ОГЛЯД

1.1 Промислові мережі передачі даних

Обмін інформацією між пристроями, що входять до складу автоматизованої системи (комп'ютерами, контролерами, датчиками, виконувачами пристроями), відбувається в загальному випадку через промислову мережу (Fieldbus, "польову шину").

Промисловою мережею називають комплекс обладнання і програмного забезпечення, які забезпечують обмін інформацією (комунікацію) між декількома пристроями. Промислова мережа є основою для побудови розподілених систем збору даних і управління.

Оскільки в промисловій автоматизації мережеві інтерфейси можуть бути невід'ємною частиною пристроїв, що з'єднуються, а мережеве програмне забезпечення прикладного рівня моделі OSI виконується на основному процесорі промислового контролера, то відокремити мережеву частину від пристроїв, що об'єднуються в мережу, іноді фізично неможливо. З іншого боку, зміну однієї мережі на іншу часто можна виконати за допомогою заміни електричного кабелю ПО і мережевого адаптера або введенням перетворювача інтерфейсу, тому часто один і той же тип ПЛК може використовуватися в мережах різних типів.

З'єднання промислової мережі з її компонентами (пристроями, вузлами мережі) виконується за допомогою інтерфейсів. Мережевим інтерфейсом називають логічну і (або) фізичну межу між пристроєм і середовищем передачі інформації. Зазвичай цією межею є набір електронних компонентів і пов'язаного з ними програмного забезпечення. При істотних модифікаціях внутрішньої структури пристрою або програмного забезпечення інтерфейс залишається без змін, що є однією з ознак, що дозволяють виділити інтерфейс в складі обладнання.

Найбільш важливими параметрами інтерфейсу є пропускна здатність і максимальна довжина кабелю, що підключається. Промислові інтерфейси зазвичай забезпечують гальванічну розв'язку між пристроями, що сполучаються. Найбільш поширені в промисловій автоматизації послідовні інтерфейси RS-485, RS-232, RS-422, Ethernet, CAN, HART, AS-інтерфейс.

Для обміну інформацією взаємодіючі пристрої повинні мати однаковий протокол обміну. У простій формі протокол - це набір правил, які керують

					ЕЛТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

обміном інформацією. Він визначає синтаксис і семантику повідомлень, операції управління, синхронізацію і стану при комунікації. Протокол може бути реалізований апаратно, програмно або програмно-апаратно. Назва мережі зазвичай збігається з назвою протоколу, що пояснюється його визначальною роллю при створення мережі.

Зазвичай мережа використовує кілька протоколів, що утворюють стек протоколів - набір пов'язаних комунікаційних протоколів, які функціонують спільно і використовують деякі або всі сім рівнів моделі OSI. Для більшості мереж стек протоколів реалізований за допомогою спеціалізованих мережевих мікросхем або вбудований в універсальний мікропроцесор.

Взаємодія пристроїв в промислових мережах виконується відповідно до моделей клієнт-сервер або видавець-передплатник (виробник-споживач). У моделі клієнт-сервер взаємодіють два об'єкти. Сервером є об'єкт, який надає сервіс, тобто який виконує деякі дії за запитом клієнта. Мережа може містити кілька серверів і кілька клієнтів. Кожен клієнт може посилати запити до більш ніж одного сервера, кожен сервер може відповідати на запити декількох клієнтів. Ця модель зручна для передачі даних, які з'являються періодично або в заздалегідь визначений час, як, наприклад, значення температури в періодичному технологічному процесі. Однак ця модель незручна для передачі подій, що випадково виникають, наприклад, події, що складається в випадкового спрацьовування датчика рівня, оскільки для отримання цієї події клієнт повинен періодично, з високою частотою, запитувати стан датчика і аналізувати його, перевантажуючи мережу марним трафіком.

У моделі взаємодії видавець-передплатник є один видавець і безліч передплатників. Передплатники повідомляють видавцеві список тегів, значення яких вони хочуть отримувати за визначеним розкладом або по мірі появи нових даних. Кожен клієнт може підписатися на свій набір тегів. Відповідно до встановленого розкладу видавець розсилає передплатникам запитану інформацію.

У будь-якої моделі взаємодії можна виділити пристрій, який управляє іншим (підлеглим) пристроєм. Пристрій, що проявив ініціативу в обміні, називають провідним, головним або майстром (Master). Пристрій, який відповідає на запити майстра, називають веденим, підлеглим або слейв (Slave). Підлеглий пристрій ніколи не починає комунікацію першим. Він чекає запиту від ведучого і тільки відповідає на запити. Наприклад, в моделі клієнт-сервер

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

клієнт є майстром, сервер - підлеглим. У моделі видавець-передплатник на етапі підписки майстром є клієнт, а на етапі розсилки публікацій - сервер.

У мережі може бути один або декілька провідних пристроїв. Такі мережі називається, відповідно, одномайстерними або багатомайстерними. У багатомайстерної мережі виникає проблема вирішення конфліктів між пристроями, що намагаються одночасно отримати доступ до середовища передачі інформації. Конфлікти можуть бути вирішені шляхом передачі маркера, як, наприклад, в мережі Profibus, методом побітного порівняння ідентифікатора (використовується в CAN), методом прослуховування мережі (використовується в Ethernet) і методом запобігання колізій (використовується в бездротових мережах).

У всіх мережах застосовується "широкомовлення" без певної адреси, тобто всім учасникам мережі. Такий режим використовується зазвичай для синхронізації процесів в мережі, наприклад, для одночасного запуску процесу введення даних усіма пристроями введення або для синхронізації годин.

Деякі мережі використовують Багатоабонентський режим, коли одне й те саме повідомлення надсилається кільком пристроям.

Передача інформації в мережі виконується через канал між передавальним і приймальним пристроєм. Канал є поняттям теорії інформації і включає в себе лінію зв'язку і приймально-передавальні пристрої. У загальному випадку замість терміна "лінія зв'язку" використовують термін "середовище передачі", в якості якого може виступати, наприклад, оптоволокно, ефір або вита пара проводів.

1.2 Перешкоди, що виникають в дротових каналах зв'язку

У процесі проходження по реальним каналам зв'язку сигнали піддаються спотворенням, тому одержані повідомлення відтворюються з деякими помилками. Ці помилки обумовлені характеристиками тракту передачі, а також перешкодами, які впливають на сигнал. Зміна характеристик тракту, як правило, має регулярний характер, і тому їх можна в більшості випадків усунути за допомогою відповідної корекції. Перешкоди ж, що впливають на сигнал, мають випадковий характер, тобто вони заздалегідь невідомі і тому їх вплив не можна повністю усунути. «На заваді» прийнято називати будь-яке випадковий вплив на сигнал, що знижує вірогідність відтворення переданих повідомлень. Існуючі перешкоди дуже різні за своєю природою і фізичного впливу.

					ЕЛТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

У каналах дротового зв'язку основними видами перешкод є імпульсні шуми і переривання зв'язку. Імпульсні шуми виникають при автоматичній комутації та внаслідок перехресних наведень. «Перериванням зв'язку» називається явище, при якому сигнал або різко згасає, або зовсім пропадає, наприклад через порушення контактів при з'єднанні. Всі зазначені перешкоди відносяться до «зовнішніх» перешкод, однак є і «внутрішні» перешкоди, що виникають в апаратурі, наприклад в підсилювачах і перетворювачах частоти. Внутрішні перешкоди обумовлені, головним чином, наявністю теплових шумів - хаотичного руху носіїв заряду (електронів) в провідниках. Ці перешкоди принципово непереборні. У загальному випадку вплив перешкод на корисний сигнал можна представити у вигляді оператора

$$Z(t) = L[S(t), n(t)]. \quad (1.1)$$

Перешкоди, що діють в каналі передачі даних, в залежності від характеру впливу поділяються на адитивні перешкоди $n(t)$ підсумовується з корисним сигналом і мультиплікативні перешкоди $\mu(t)$.

Аддитивна перешкода $n(t)$:

$$Z(t) = S(t) + n(t), \quad (1.2)$$

де $S(t)$ - переданий сигнал, $n(t)$ - адитивна перешкода;

Мультиплікативна перешкода $\mu(t)$:

$$Z(t) = S(t) \times \mu(t), \quad (1.4)$$

де $\mu(t)$ - деяка випадкова функція, що відображає зміну в часі коефіцієнта передачі каналу зв'язку.

У реальних системах зв'язку часто діють як адитивна, так і мультиплікативна перешкоди:

$$Z(t) = \mu(t) \times S(t) + n(t). \quad (1.5)$$

Схема дії перешкод в лінії зв'язку показана на рис. 1.1.

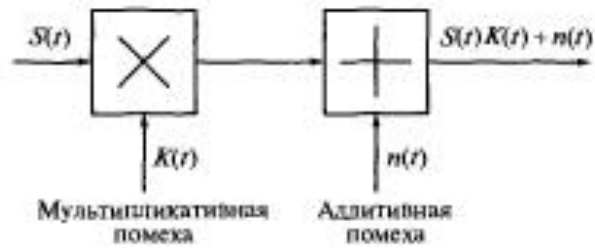


Рисунок 1.1 - Схема дії перешкод в лінії зв'язку

Аддитивні перешкоди містять три складові: флуктуаційну, гармонійну, і імпульсну.

Флуктуаційна перешкода в більшості випадків має нормальний розподіл. Причинами флуктуаційних перешкод є не тільки власні шуми активних і пасивних елементів каналоутворюючої апаратури, а й шуми ланцюгів кабельних ліній, а також шуми, що виникають за рахунок зовнішніх електромагнітних впливів.

Однією з найбільш поширених видів гармонічних перешкод, що з'являються в лініях системи, є перешкода на частоті змінного струму 50 Гц і її вищі гармоніки, а також синусоїдальні перешкоди більш високих частот, що потрапляють в канал за рахунок передачі будь-якої інформації по сусідніх лініях зв'язку.

Імпульсні перешкоди, як флуктуаційні, так і гармонічні, найчастіше проникають в канал через лінійний тракт системи передачі. Ці перешкоди на виході каналу представляють собою реакцію каналу на ударну дію короткого імпульсу тривалістю близько 75 мкс.

Одним з основних параметрів імпульсної перешкоди є її амплітуда, тобто величина максимального викиду напруги. Для оцінки розподілу амплітуд імпульсних перешкод використовують ймовірність сумарного часу перевищення імпульсною завадою певного порогового значення, або ймовірність перевищення амплітудою перешкоди певного рівня. Важливим параметром імпульсної перешкоди є її інтенсивність $V_{ін}$, яка визначається кількістю імпульсних перешкод, які перевищили заданий поріг на годину, що перевищили задані пороги.

Мультиплікативні перешкоди в каналах зв'язку виражаються в основному в зміні залишкового загасання, що приводить до змін рівня сигналу. Розрізняють плавні і стрибкоподібні зміни рівня сигналу. Різке зниження рівня щодо номінального називають перервою сигналу.

При передачі сигналу по каналу зв'язку виникають паразитні індуктивні і ємнісні зв'язки, які викликають перешкоди і відповідно призводять до спотворення переданого сигналу.

Паразитні індуктивні і ємнісні зв'язки являють собою фізичні фактори, що характеризують вплив електричних і магнітних полів, що виникають ланцюгах будь-якого функціонуючого радіоелектронного засобу.

Паразитний індуктивний зв'язок проявляється в ході наступних фізичних процесів. У просторі, що оточує будь-який ланцюг, по якому протікає електричний струм I , виникає магнітне поле, постійне або змінне. У сусідніх провідниках, що знаходяться в змінному магнітному полі, порушуються змінні ЕРС $E = I\omega M$, де M - взаємна індуктивність. Величина M пропорційно індуктивності впливають один на одного елементів ланцюгів і обернено пропорційна відстані між ними. Наприклад, взаємоіндуктивність двох мідних прямих паралельних провідників довжиною 100 мм і товщиною 0.02 мм при інтервалі між ними 2 мм складає 0.07 мкГн, а при інтервалі 10 мм - 0.04 мкГн.

Ємнісний паразитний зв'язок виникає між будь-якими елементами схеми, перш за все, між паралельно розташованими дротами, а також точками схеми і корпусом (шасі). Ємнісний зв'язок залежить від геометричних розмірів елементів ланцюгів і відстані між ними. Наприклад, ємність між двома паралельними дротами довжиною 100 мм і діаметром 0.1 мм зменшується з 0.75 пф до 0.04 пф при збільшенні відстані між ними з 2 до 50 мм. Для дротів діаметром 2 мм ця ємність при тих же умовах знижується з 5 пф до 0.07 Пф.

Через паразитні індуктивні і ємнісні зв'язки виникають паразитні наведення - небажані електромагнітні сигнали від прокладених в джгутах суміжних кабелів. Наведення можуть привести до неприпустимого рівня перешкод в каналі зв'язку і, отже, втрати інформації.

1.3 Інтерфейс RS-485, його характеристики

RS-485 (EIA / TIA-485) - це стандарт, який визначає електричні характеристики приймачів і передавачів інформації для використання в балансних цифрових багатоточкових системах. Інтерфейс RS-485 є одним з найбільш поширених стандартів фізичного рівня в сучасних засобах промислової автоматизації.

Електричні характеристики інтерфейсу наведені в таблиці 1.1.

					ЕЛТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

Таблиця 1.1 - Електричні характеристики інтерфейсу RS-485

Параметр	Значення
Максимальне число приймачів / передавачів	32/32
Максимальна довжина кабелю, м	1200
Максимальна швидкість передачі даних, Мбіт / с	10
Рівень логічної «1» передавача, В	+ 1,5 ... + 6
Рівень логічного «0» передавача, В	-1,5 ... -6
Діапазон синфазної напруги передавача, В	-1 ... + 3
Максимальний струм короткого замикання передавача, мА	250
Допустимий опір навантаження передавача, Ом	54
Поріг чутливості приймача, мВ	± 200
Допустимий діапазон напруг приймача, В	-7 ... + 12
Рівень логічної «1» приймача, мВ	більше +200
Рівень логічного «0» приймача, мВ	менше -200
Вхідний опір приймача, кОм	12

В основі інтерфейсу RS-485 лежить спосіб диференціальної (балансної) передачі даних. Суть даного методу полягає в наступному: по одному дроту (умовно лінія А) передається нормальний сигнал, а по другому проводу (умовно лінія В) передається інвертований сигнал, таким чином, між двома проводами витої пари завжди існує різниця потенціалів (рисунок 1.2). Для випадку логічної "одиниці" різниця потенціалів позитивна, для логічного «нуля» - негативна.

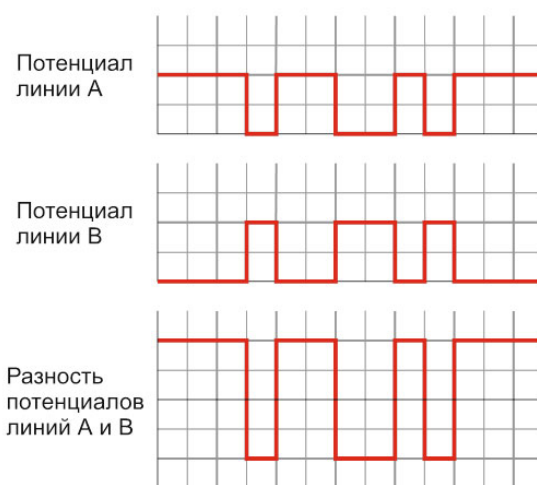


Рисунок 1.2 - Діаграма диференційної (балансної) передачі даних

Перевагою диференціальної (балансної) передачі даних є висока стійкість до синфазних перешкод. Синфазна перешкода - перешкода, яка діє на обидві лінії зв'язку однаково. Найчастіше лінії зв'язку прокладаються в місцях, схильних до впливу неоднорідних електромагнітних полів, електромагнітна хвиля, проходячи через ділянку лінії зв'язку, буде наводити в обох проводах потенціал. У разі RS-232 інтерфейсу корисний сигнал, який передається потенціалом щодо загального «земляного» дроту був би загублений. При диференціальній передачі не відбувається спотворення сигналу враховуючи те, що перешкода однаково діє на обидва провідника і наводить в них однаковий потенціал, в результаті чого різниця потенціалів (корисний сигнал) залишається незмінною. З цієї причини лінії зв'язку інтерфейсу RS-485 являють собою два скручених між собою провідника і називаються витоб парою. Прямі виходи «А» підключаються до одного дроту, а інверсні «В» до другого проводу (рис 1.3). У разі неправильного підключення виходів до ліній приймачі не вийдуть з ладу, але при цьому правильно функціонувати вони не будуть.

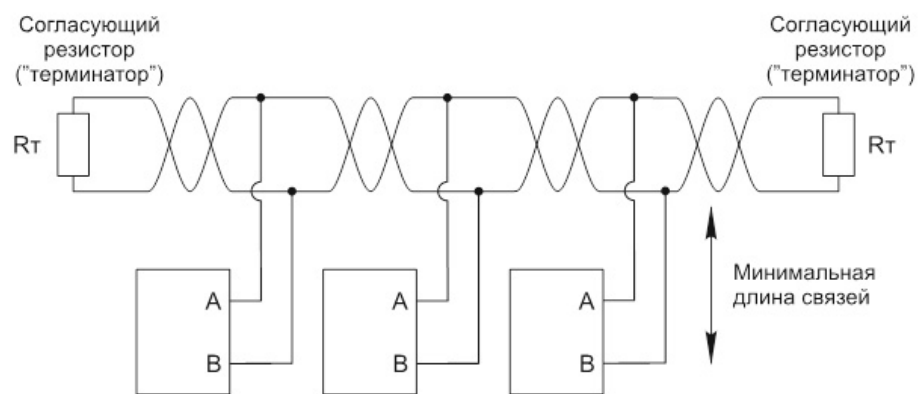


Рисунок 1.3 - Конфігурація мережі RS-485

Апаратна реалізація інтерфейсу - мікросхеми приймачів з диференціальними входами / виходами (до лінії) і цифровими портами (до портів UART контролера). Існують два варіанти такого інтерфейсу: RS-422 і RS-485.

RS-422 - повнодуплексний інтерфейс. Прийом і передача йдуть по двом окремим парам проводів. На кожній парі проводів може бути тільки по одному передавачу.

RS-485 - напівдуплексний інтерфейс. Прийом і передача йдуть по одній парі проводів з поділом за часом. У мережі може бути багато передавачів, так як вони можуть відключатися в режимі прийому.

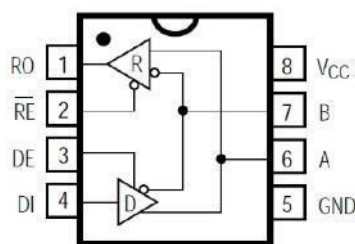


Рисунок 1.4 - Прийомопередавач інтерфейсу RS-485

- D (driver) - передавач;
- R (receiver) - приймач;
- DI (driver input) - цифровий вхід передавача;
- RO (receiver output) - цифровий вихід приймача;
- DE (driver enable) - дозвіл роботи передавача;
- RE (receiver enable) - дозвіл роботи приймача;
- A - прямий диференціальний вхід / вихід;
- B - інверсний диференціальний вхід / вихід.

Цифровий вихід приймача (RO) підключається до порту приймача UART (RX). Цифровий вхід передавача (DI) до порту передавача UART (TX). Оскільки на диференціальній стороні приймач і передавач з'єднані, то під час прийому потрібно відключати передавач, а під час передачі - приймач. Для цього служать керуючі входи - дозвіл приймача (RE) і дозвіл передавача (DE). Так як вхід RE інверсний, то його можна з'єднати з DE і переключати приймач і передавач одним сигналом з будь-якого порту контролера. При рівні "0" - робота на прийом, при "1" - на передачу.

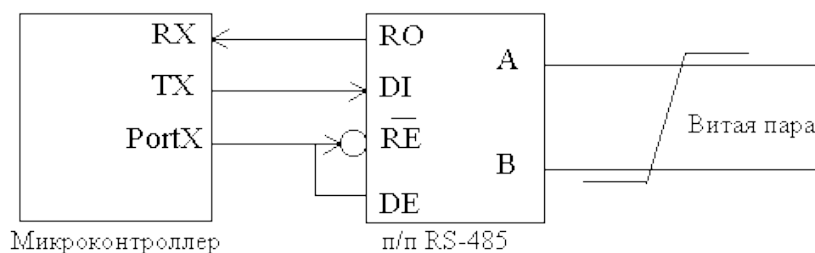


Рисунок 1.5 – Підключення прийомо-передавача інтерфейсу RS-485 до мікроконтролера

Всі пристрої підключаються до однієї витієї пари однаково: прямі виходи (A) до одного проводу, інверсні (B) - до іншого.

Вхідний опір приймача з боку лінії (R_{AB}) зазвичай становить 12 кОм. Так як потужність передавача не безмежна, це створює обмеження на кількість приймачів, підключених до лінії. Згідно зі специфікацією RS-485 з урахуванням узгоджувальних резисторів передавач може вести до 32 приймачів. Однак є ряд мікросхем з підвищеним вхідним опором, що дозволяє підключити до лінії значно більше 32 пристроїв.

1.4 Організація мереж на основі інтерфейсу RS-485

Конфігурація мережі являє собою послідовне приєднання приймачів до витой пари (топология «шина»), при цьому мережа не повинна містити довгих відгалужень при підключенні пристроїв, так як довгі відгалуження викликають неузгодженості і відображення сигналу (рисунок 1.6).

Стандарт передбачає, що пристрої підключаються безпосередньо до шини. При цьому скручування і зрощування кабелю не допускаються. При збільшенні довжини ліній зв'язку при високій швидкості передачі даних має місце так званий ефект довгих ліній. Він полягає в тому, що швидкість поширення електромагнітних хвиль в провідниках обмежена, для прикладу у провідника з поліетиленовою ізоляцією вона обмежена на рівні близько 206 мм / нс. Крім цього електричний сигнал має властивість відбиватися від кінців провідника і його відгалужень. Для коротких ліній подібні процеси протікають швидко і не впливають на роботу мережі, однак при значних відстанях в сотні метрів відбита від кінців провідників хвиля може спотворити корисний сигнал, що призведе до помилок і збоїв.

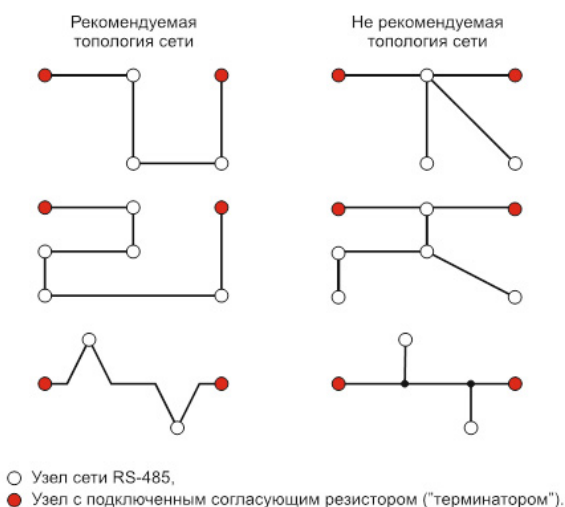


Рисунок 1.6 – Приклади топологій мережі RS-485

Проблему відображень сигналу в інтерфейсі RS-485 вирішують за допомогою узгоджувальних резисторів - «термінаторів», які встановлюються безпосередньо біля виходів двох приймачів максимально віддалених один від одного. Слід зазначити, що в більшості випадків «термінатори» вже змонтовані в споживаючих пристроях і підключаються до мережі за допомогою відповідних перемичок на корпусі пристрою. Номінал «термінатора» відповідає хвильовому опору кабелю, при цьому потрібно пам'ятати, що хвильовий опір кабелю залежить від його характеристик і не залежить від його довжини. Наприклад, для витої пари UTP-5, що використовується для прокладки Ethernet хвильовий опір становить 100 ± 15 Ом. Спеціалізований кабель Belden 9841 ... 9844 для прокладки мереж RS-485 має хвильовий опір 120 Ом, тому розрахунками резистора - «термінатора» можна знехтувати і використовувати 120 Ом.

Екрановані виті пари (наприклад, кабелі Belden 9841, 3106A) рекомендується застосовувати в особливо відповідальних системах, а також при швидкостях обміну понад 500 Кбіт / сек.

Потрібно відзначити, що для забезпечення відмовостійкості і завадозахищеності зі збільшенням довжини ліній зв'язку швидкість передачі бажано зменшити. Залежність швидкості обміну від довжини ліній представлена на рис. 1.7. Дана залежність може відрізнитися за інших умов і носить скоріше рекомендаційний характер.

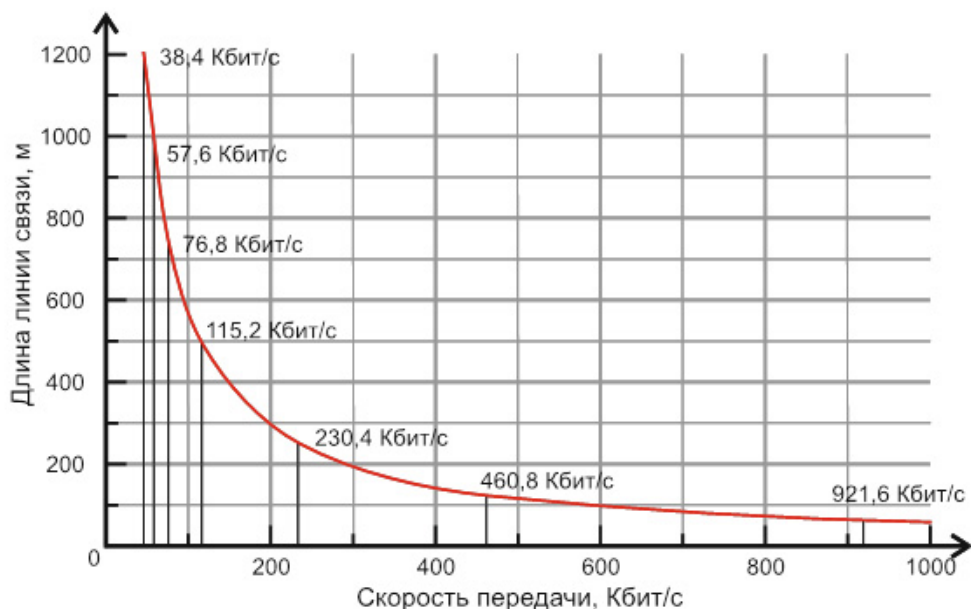


Рисунок 1.7 – Залежність швидкості обміну від довжини лінії зв'язку

Поріг чутливості приймача становить ± 200 мВ, тобто при різниці потенціалів на вході приймача в діапазоні від мінус 200 мВ до плюс 200 мВ його вихідний стан буде перебувати в стані невизначеності. Різницю потенціалів понад 200 мВ приймач приймає як логічну «1», а різницю потенціалів менш мінус 200 мВ приймач приймає як логічний «0». Стан невизначеності може виникнути, коли жоден з передавачів не активний, відключений від мережі, або знаходиться в «третьому стані», або всі пристрої мережі знаходяться в режимі прийому інформації. Стан невизначеності вкрай небажаний, так як він викликає помилкові спрацьовування приймача через несинфазні перешкоди.

Використання захисного зміщення дозволяє виключити можливість виникнення невизначеного стану в мережі. Для цього лінію А необхідно підтягнути резистором до живлення (pullup), а лінію В резистором - до «землі» (pulldown). В результаті, з урахуванням «термінаторів», вийде резистивний діляк напруги. Для надійної роботи мережі необхідно забезпечити зсув близько 250 ... 300 мВ (рис. 1.8).

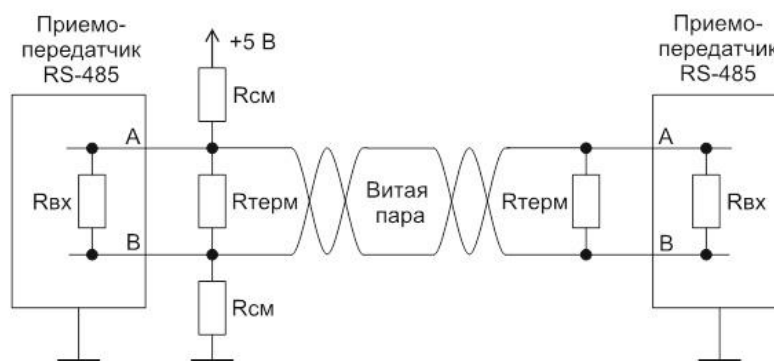


Рисунок 1.8 – Захисне заземлення

Розглянемо ситуацію, коли в мережі знаходяться два пристрої. Нам необхідно отримати зміщення 250мВ, при цьому в мережі підключені два термінальних резистора по 120Ом, і є джерело напруги + 5В, обидва приймача володіють одиничним навантаженням - їх опір становить 12 кОм.

З огляду на те, що термінальні резистори по 120Ом і обидва приймача по 12кОм включені паралельно, то їх загальний опір дорівнює:

$$R_{\text{мережі}} = (R_{\text{заг.терм}} \cdot R_{\text{заг.вх}}) / (R_{\text{заг.терм}} + R_{\text{заг.вх}}) = (60 \cdot 6000) / (60 + 6000) = 60 \text{ Ом.}$$

Розрахуємо струм в ланцюзі зміщення:

$$I_{\text{зм}} = U_{\text{зм}} / R_{\text{мережі}} = 0,250 \text{ В} / 60 \text{ Ом} = 0,0042 \text{ А}.$$

При цьому послідовний опір ланцюга зміщення складе:

$$R_{\text{зм.заг}} = U_{\text{живл}} / I_{\text{зм}} = 5 \text{ В} / 0,0042 \text{ А} = 1140 \text{ Ом}.$$

Отримуємо опір резисторів зміщення:

$$R_{\text{зм}} = 1140 / 2 = 570 \text{ Ом}.$$

Вибираємо найближчий номінал 560 Ом.

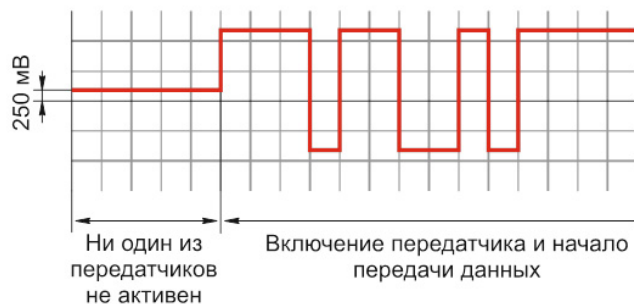


Рисунок 1.9 – Діаграма передачі даних при використанні захисного зміщення

Виходячи з розрахунку захисного зміщення можна помітити, що через дільник напруги постійно протікає струм (для випадку вище це 4,2 мА), що може бути неприпустимим в системах з малим енергоспоживанням. Це є серйозним недоліком захисного зміщення.

Знизити втрати можна збільшенням номіналу резисторів узгодження до 1,1 кОм і вище, але в даному випадку доведеться шукати компроміс між енергоспоживанням і надійністю мережі.

Для гальванічно розв'язаної лінії резистори зміщення слід підтягувати до «землі» і живлення з боку ізольованої лінії.

Для захисту від перешкод екран витой пари слід заземлювати в одній точці, при цьому стандарт не визначає в який, тому часто екран кабелю заземлюється на одному з його кінців. Іноді причиною виникнення помилок при передачі сигналу є працюючий поблизу радіопередавач. Щоб усунути вплив радіосигналу на передавальний кабель досить встановити високочастотний конденсатор малої ємності між екраном кабелю і заземленням електричної мережі близько 1 ... 10нФ.

Якщо прилади, об'єднані в одну мережу, живляться від різних джерел або знаходяться на значній відстані один від одного, то необхідно додатковим дренажним проводом об'єднати «землі» всіх пристроїв. Це правило виходить з того, що різниця потенціалів між лінією і «землею» за стандартом не повинна перевищувати від мінус 7 до плюс 12 В. В разі, коли пристрої знаходяться на

значній відстані один від одного, або живляться від різних джерел різниця потенціалів на вході приймача може перевищити в кілька разів допустимий діапазон, що призведе до виходу з ладу приймача. При цьому слід враховувати, що з'єднання в мережі RS-485 потрібно починати саме з дренажного дроту, а здійснюючи відключення пристрою в останню чергу від'єднувати дренажний дріт. Для обмеження «блукаючих» струмів в дренажному дроті його слід підключати до кожної сигнальної землі через резистор номіналом 100 Ом потужністю 0,5 Вт, крім цього необхідно через такий же резистор 100 Ом 0,5 Вт підключити дренажний дріт до захисного заземлення. Рекомендується здійснювати захисне заземлення дренажного дроту в одній точці, щоб виключити постійне протікання «блукаючого» струму через нього в порівнянні з випадком, коли дренажний провід заземлюється в кожному пристрої. Не слід використовувати екран кабелю в якості дренажного дроту.

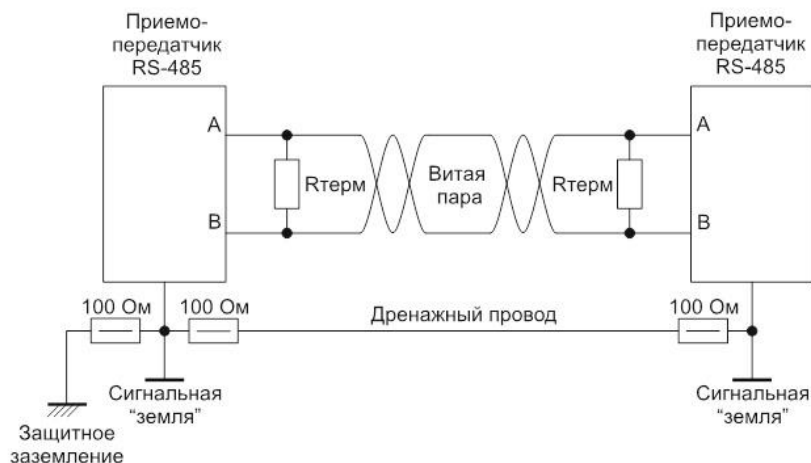


Рисунок 1.10 – Використання дренажного дроту для зрівнювання потенціалу «земель»

Для захисту мереж від синфазних перенапруг і імпульсних перешкод менше 2 кВ варто використовувати приймачі з гальванічною розв'язкою. Якщо ж високий потенціал буде прикладено диференційно, тобто до одного провіднику лінії, то приймач буде пошкоджений, так як різниця потенціалів між провідниками повинна знаходитися в діапазоні від мінус 7 до плюс 12 В.

Захист пристроїв мережі RS-485 від диференціальних перенапруг, що складають десятки кіловольт, наприклад, попадання розряду блискавки в лінію, здійснюється за рахунок використання спеціальних захисних пристроїв. У найпростішому випадку всі провідники лінії шунтуються обмежувачами напруги на «землю» (рисунок 1.11). Якщо заземлення лінії неможливо, то провідники

лінії шунтуються обмежувачами між собою (рис. 1.11 б). Захист, організований на варисторах, супрессорах, газорозрядних трубках, здатний витримувати лише короточасні сплески напруги. Додатковий захист від струмів короткого замикання в лініях можна забезпечити за допомогою установки в лінію плавких запобіжників.

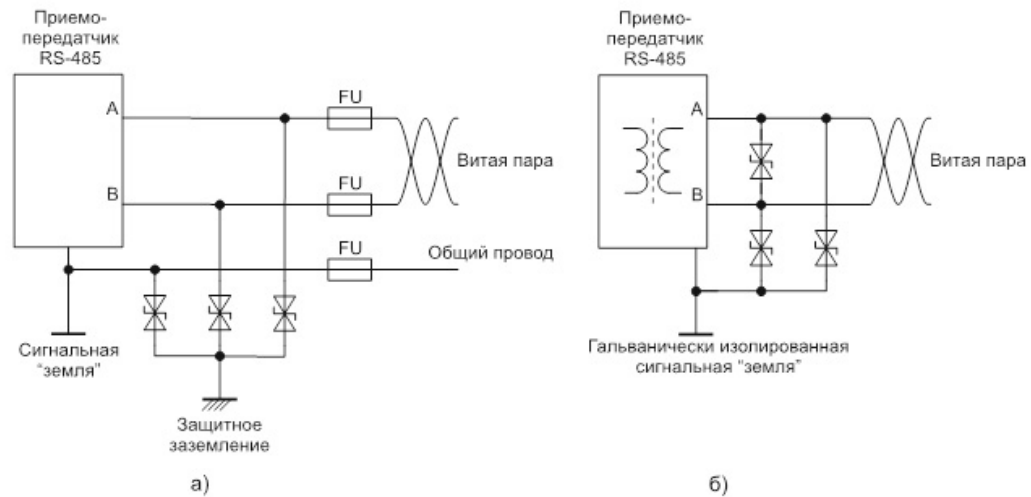


Рисунок 1.11 – Варіанти захисту мережі RS-485 від перенапруг і імпульсних перешкод

Як правило, пристрої, що працюють в мережах RS-485 крім «термінаторів» мають вбудований захист від перенапруг і імпульсних перешкод. Крім цього на ринку існує безліч готових пристроїв придушення імпульсних перешкод, принцип дії яких також заснований на застосуванні варисторів і газорозрядних трубок. Варто лише пам'ятати, що кожний додатковий пристрій захисту, встановлений в мережі, вносить додаткову ємність, еквівалентну ємності кабелю довжиною 120 ... 130 м.

1.5 Організація протоколу передачі даних по мережі RS-485

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

У централізованій мережі один пристрій завжди провідний (майстер). Він генерує запити і команди іншим (веденим) пристроям. Ведені пристрої можуть

передавати тільки по команді ведучого. Як правило, обмін між веденими йде тільки через ведучого, хоча для прискорення обміну можна організувати передачу даних від одного веденого до іншого по команді ведучого.

У децентралізованій мережі роль ведучого може передаватися від пристрою до пристрою або за певним алгоритмом черговості, або по команді поточного ведучого до наступного (передача маркера ведучого). При цьому ведений пристрій може в своїй відповіді ведучому передати запит на перехід в режим ведучого і очікувати дозволу або заборони.

Послідовний канал за мірками контролера є досить повільним. На швидкості 9600 бод передача одного символу займає більше мілісекунди. Тому, коли контролер щільно завантажений обчисленнями і не повинен їх зупиняти на час обміну по UART, потрібно використовувати переривання по завершенню прийому і передачі символу. Можна виділити місце в пам'яті для формування посилки на передачу і збереження прийнятої посилки (буфер посилки), а також вказівники на позицію поточного символу. Переривання по завершенню прийому або передачі символу викликають відповідні підпрограми, які передають або зберігають черговий символ із зсувом вказівника і перевіркою ознаки кінця повідомлення, після чого повертають управління основній програмі до наступного переривання. По завершенню надсилання і прийняття всієї посилки або формується користувальницький прапорець, що відпрацьовується в основному циклі програми, або відразу викликається підпрограма обробки повідомлення.

У загальному випадку посилка по послідовному каналу складається з керуючих байтів (синхронізація посилки, адрес відправника і одержувача, контрольної суми та ін.) і власне байтів даних.

Одним зі стандартних протоколів послідовної передачі є MODBUS, його підтримку забезпечують багато виробників промислових контролерів. Але якщо потрібно здійснити просту передачу кількох байтів повідомлення, можливі варіанти організації простих протоколів.

Основне завдання в організації протоколу - змусити всі пристрої розрізняти керуючі байти і байти даних. Наприклад, ведений пристрій, отримуючи по лінії потік байтів, має розуміти, де початок посилки, де кінець і кому вона адресована.

1) Часто зустрічаються протоколи на основі ASCII-коду. Керуючі символи і дані передаються у вигляді звичайних ASCII символів. Посилка може виглядати так:

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		21

В HEX вигляді: 3Ah 31h 32h 52h 53h 34h 38h 35h 0Dh

В ASCII вигляді: ":" "1" "2" "R" "S" "4" "8" "5" / ПС /

На початку керуючий символ початку посилки ":", наступні дві цифри - адреса одержувача (12), потім символи даних (RS485) і в кінці - керуючий символ кінця посилки 0Dh (переведення рядка). Всі пристрої на лінії, прийнявши символ ":", починають записувати в пам'ять посилку до символу кінця рядка 0Dh. Потім порівнюють адресу з посилки зі своєю адресою. Пристрій з адресою, що співпадає, обробляє дані посилки, решта - ігнорують посилку. Дані можуть містити будь-які символи, крім керуючих (":", 0Dh).

Перевага цього протоколу в зручності налагодження системи і простоті синхронізації посилки. Можна через перетворювач RS485-RS232 підключити лінію до COM-порту комп'ютера і в будь-якому терміналі побачити всю інформацію, що проходить. Недоліки - занадто великий розмір посилки при передачі великої кількості двійкової інформації, адже на передачу кожного байта потрібно два ASCII символи (7Fh - "7", "F"). Крім того, треба перетворювати дані з двійкового виду в ASCII і назад.

2) Можна організувати протокол з безпосередньою передачею двійкових даних. При цьому керуючі символи і байти даних розрізняються за допомогою налаштування додаткового дев'ятого біта в UART. Для керуючих символів цей біт встановлюється в "1". Першим в посилці передається керуючий символ з одиничним дев'ятим бітом - інші його "нормальні" біти можуть містити адресу пристрою-одержувача, ознаку початку / кінця посилки та іншу службову інформацію. Потім передаються байти даних з нульовим дев'ятим бітом. Всі приймаючі пристрої дізнаються по дев'ятому біту керуючий символ і за змістом його інших бітів визначають, кому адресовані наступні дані. Пристрій, що адресується приймає дані, а всі інші ігнорують їх до наступного керуючого символу.

UART деяких контролерів, наприклад C167 (Infineon), може в особливому режимі (wakeur) автоматично розпізнавати в отриманому байті дев'ятий біт і генерувати переривання при отриманні тільки керуючого символу. Пристрій, що адресується при цьому потрібно переключити в режим звичайного прийому до наступного керуючого символу. Це дозволяє іншим пристроям заощадити час на обробці переривань при отриманні байтів даних, адресованих не ним.

Якщо потрібно сполучення системи і комп'ютера з Windows, такий протокол краще не застосовувати, тому що у Windows можуть бути проблеми з розпізнаванням дев'ятого біта в UART.

					ЕлІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		22

3) Протокол може бути "чисто" двійковим, тобто без виділення спеціальних керуючих символів. Синхронізація посилки в цьому випадку може здійснюватися за рахунок відстеження паузи між прийнятими байтами. Приймаючий пристрій відраховує час з моменту останнього прийому байта до наступного, і якщо ця пауза виявляється більше якоїсь величини (наприклад, 1.5 - 3.5 байта), робиться висновок про втрату попередньої посилки і початку нової. Навіть якщо попередня посилка була незакінчена - приймальний буфер скидається. Можна також синхронізувати посилки за унікальною стартовою послідовністю байтів (по аналогії зі стартовим символом в ASCII протоколі). У таких протоколах треба вживати особливих заходів для захисту від прийому помилкової посилки, розпочатої через перешкоди.

Протокол Modbus. Протокол Modbus і мережа Modbus [Modbus - Modicon] є найпоширенішими в світі. Незважаючи на свій вік (стандартом де-факто Modbus став ще в 1979 році), Modbus не тільки не застарів, але, навпаки, істотно зросла кількість нових розробок і обсяг організаційної підтримки цього протоколу. Мільйони Modbus-пристроїв по всьому світу продовжують успішно працювати, а остання версія опису протоколу з'явилася в грудні 2006 р [Modbus].

Одним з переваг Modbus є відсутність необхідності в спеціальних інтерфейсних контролерах (Profibus і CAN вимагають для своєї реалізації замовні мікросхеми), простота програмної реалізації і елегантність принципів функціонування. Все це знижує витрати на освоєння стандарту як системними інтеграторами, так і розробниками контролерних обладнань. Високий ступінь відкритості протоколу забезпечується також повністю безкоштовними текстами стандартів, які можна скачати з сайту www.modbus.org.

Протокол Modbus за поширеністю на території СНД конкурує тільки з Profibus. Популярність протоколу в даний час пояснюється, перш за все, сумісністю з великою кількістю обладнання, яке має протокол Modbus. Крім того, Modbus має високу достовірність передачі даних, пов'язану із застосуванням надійного методу контролю помилок. Modbus дозволяє уніфікувати команди обміну завдяки стандартизації номерів (адрес) регістрів і функцій їх читання-запису.

Основним недоліком Modbus є мережевий обмін по типу "ведучий / ведений", що не дозволяє веденим пристроям передавати дані по мірі їх появи і тому вимагає інтенсивного опитування ведених пристроїв провідним.

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23

Різновидами Modbus є протоколи Modbus Plus [Modicon] - багатомайстерний протокол з кільцевої передачею маркера і Modbus TCP [Modbus], розрахований на використання в мережах Ethernet і інтернет.

Протокол Modbus має два режими передачі: RTU (Remote Terminal Unit - «віддалений термінальний пристрій») і ASCII. Стандарт передбачає, що режим RTU в протоколі Modbus повинен бути присутнім обов'язково, а режим ASCII є опційним. Користувач може вибирати будь-який з них, але всі модулі, включені в мережу Modbus, повинні мати один і той же режим передачі.

Розглянемо більш детально протокол Modbus RTU. Відзначимо, що Modbus ASCII не можна плутати з приватно-фірмовим протоколом DCON, який використовується в модулях фірм Advantech і ICP DAS і не відповідає стандарту Modbus.

Стандарт Modbus передбачає застосування фізичного інтерфейсу RS-485, RS-422 або RS-232. Найбільш поширеним для організації промислової мережі є 2-провідний інтерфейс RS-485. Для з'єднань точка-точка може бути використаний інтерфейс RS-232 або RS-422.

Таблиця 1.2 Модель OSI для Modbus

Номер рівня	Назва рівня	Реалізація
7	Прикладний	MODBUS Application Protocol
6	Рівень представлення	Ні
5	Сеансовий	Ні
4	Транспортний	Ні
3	Мережевий	Ні
2	Канальний (передачі даних)	Протокол "ведучий/ведений" Режими RTU і ASCII
1	Фізичний	RS-485 або RS-232

У стандарті Modbus є обов'язкові вимоги, що рекомендуються і опційні (необов'язкові). Існує три ступені відповідності стандарту: «повністю відповідає» - коли протокол відповідає всім обов'язковим і всім рекомендованим вимогам, «умовно відповідає» - коли протокол відповідає тільки обов'язковим вимогам і не відповідає рекомендованим, і «не відповідає».

Модель OSI протоколу Modbus містить три рівні: фізичний, канальний і прикладної.

Фізичний рівень. У нових розробках на основі Modbus стандарт рекомендує використовувати інтерфейс RS-485 з двухпроводною лінією передачі, але допускається застосування чотирьох ліній і інтерфейсу RS-232.

Modbus-шина повинна складатися з одного магістрального кабелю, від якого можуть бути зроблені відводи. Магістральний кабель Modbus повинен містити 3 провідники в загальному екрані, два з яких представляють собою виту пару, а третій з'єднує загальні ("земляні") виводи всіх інтерфейсів RS-485 в мережі. Загальний провід і екран повинні бути заземлені в одній точці, бажано близько провідного пристрою.

Пристрої можуть підключатися до кабелю трьома способами:

- безпосередньо до магістрального кабелю;
- через пасивний перехідник (трійник);
- через активний перехідник (містить розв'язуючий повторювач інтерфейсу).

В документації на пристрій і на трійник повинні бути вказані найменування ланцюгів, що підключаються.

На кожному кінці магістрального кабелю повинні бути встановлені резистори для узгодження лінії передачі, як це потрібно для інтерфейсу RS-485. На відміну від фізичного інтерфейсу RS-485, в якому термінальні резистори на низьких швидкостях обміну можна не використовувати, стандарт на протокол Modbus формально вимагає застосування термінальних резисторів для всіх швидкостей обміну. Їх номінал може бути рівним 150 Ом і потужність 0,5 Вт. Стандарт вимагає, щоб в керівництві з експлуатації пристроїв Modbus було сказано, чи є зазначені резистори всередині пристрою, або їх необхідно встановлювати при монтажі мережі. Якщо потрібні зовнішні резистори, то вони повинні мати номінал в інтервалі від 450 до 650 Ом і бути встановлені тільки в одному місці в межах кожного сегмента мережі (сегментами вважаються частини мережі між повторювачами інтерфейсу).

Modbus-пристрій обов'язково має підтримувати швидкості обміну 9600 біт / с і 19200 біт / с, з них 19200 біт / с встановлюється "за замовчуванням". Допускаються також швидкості 1200, 2400, 4800, ..., 38400 біт / с, 65 кбіт / с, 115 кбіт / с,

Швидкість передачі має бути витримана в передавачі з похибкою не гірше 1%, а приймач повинен приймати дані при відхиленні швидкості передачі до 2%.

Сегмент мережі, що не містить повторювачів інтерфейсу, повинен допускати підключення до 32 пристроїв, проте їх кількість може бути збільшена,

					ЕЛТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		25

якщо це допустимо виходячи з навантажувальної здатності передавачів і вхідного опору приймачів, які повинні бути приведені в документації на інтерфейси. Зазначення цих параметрів в документації є обов'язковою вимогою стандарту.

Максимальна довжина магістрального кабелю при швидкості передачі 9600 біт / с і перетині жив більш 0,13 кв. мм (AWG26) становить 1 км. Відводи від магістрального кабелю не повинні бути довше 20 м. При використанні багатопортового пасивного розгалужувача з N відведеннями довжина кожного відведення не повинна перевищувати значення 40 м / N.

Modbus не встановлює конкретних типів роз'ємів, але якщо використовуються роз'єми RJ45, mini-DIN або D-Shell, вони повинні бути екранованими, а цоколівки повинні відповідати стандарту.

Для мінімізації помилок при монтажі рекомендується використовувати дроти наступних кольорів: жовтий - для позитивного виводу RS-485 (на якому встановлюється логічна "1", коли через інтерфейс виводиться логічна "1"); коричневий - для другого виводу інтерфейсу RS-485; сірий - для загального проводу.

Типовим перетином кабелю є AWG 24 (0,2 кв. Мм, діаметр дроту 0,51 мм). При використанні кабелю категорії 5 його довжина не повинна перевищувати 600 м. Хвильовий опір кабелю бажано вибирати більш 100 Ом, особливо для швидкості обміну більш 19200 біт / с.

Протокол Modbus передбачає, що тільки один провідний пристрій (контролер) і до 247 ведених (модулів введення-виведення) можуть бути об'єднані в промислову мережу. Обмін даними завжди ініціюється провідним. Ведені пристрої ніколи не починають передачу даних, поки не отримають запит від ведучого. Ведені пристрої також не можуть обмінюватися даними один з одним. Тому в будь-який момент часу в мережі Modbus може відбуватися тільки один акт обміну.

Адреси з 1 по 247 є адресами Modbus пристроїв в мережі, а з 248 по 255 зарезервовані. Провідний пристрій не повинен мати адреси і в мережі не повинно бути двох пристроїв з однаковими адресами.

Провідний пристрій може посилати запити до всіх пристроїв одночасно ("широкомовний режим") або тільки одному. Для широкомовного режиму зарезервованій адресу "0" (при використанні в команді цієї адреси вона приймається всіма пристроями мережі).

					ЕЛТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		26

Опис кадру (фрейму) протоколу Modbus. У протоколі Modbus RTU повідомлення починає сприйматися як нове після паузи (тиші) на шині тривалістю не менше 3,5 символів (14 біт), тобто величина паузи в секундах залежить від швидкості передачі.



Рисунок 1.12 - Формат кадру протоколу Modbus RTU; PDU - "Protocol Data Unit" - "елемент даних протоколу"; ADU - "Application Data Unit" - "елемент даних програми"

Формат кадру показаний на рис. 1.12. Поле адреси завжди містить тільки адресу веденого пристрою, навіть у відповідях на команду, надіслану провідним. Завдяки цьому ведучий пристрій знає, від якого модуля прийшла відповідь.

Поле «Код функції» говорить модулю про те, яку дію потрібно виконати.

Поле «Дані» може містити довільну кількість байт. У ньому може міститися інформація про параметри, які використовуються в запитах контролера або відповідях модуля.

Поле «Контрольна сума» містить контрольну суму CRC довжиною 2 байта.

Структура даних в режимі RTU. У режимі RTU дані передаються молодшими розрядами вперед (рис. 1.13).

За замовчуванням в RTU режимі біт паритету встановлюють рівним 1, якщо кількість двійкових одиниць в байті непарна, і рівним 0, якщо вона парна. Такий паритет називають парних (even parity) і метод контролю називають контролем парності.

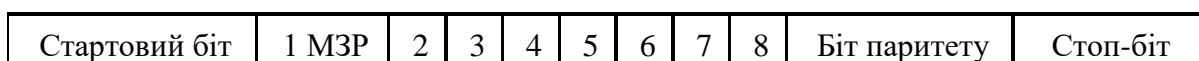


Рисунок 1.13 - Послідовність бітів в режимі RTU; МЗР - молодший значущий розряд. При відсутності біта паритету на його місце записується другий стоп-біт

При парній кількості двійкових одиниць в байті біт паритету може бути рівним 1. У цьому випадку говорять, що паритет є непарний (odd parity).

Контроль парності може бути відсутнім взагалі. У цьому випадку замість біта паритету повинен використовуватися другий стоповий біт. Для забезпечення максимальної сумісності з іншими продуктами рекомендується використовувати можливість заміни біта паритету на другий стоповий біт.

Ведені пристрої можуть сприймати будь-який з варіантів: парний, непарний паритет або його відсутність.

Структура Modbus RTU повідомлення. Повідомлення Modbus RTU передаються у вигляді кадрів, для кожного з яких відомо початок і кінець. Ознакою початку кадру є пауза (тиша) тривалістю не менше 3,5 шістнадцятирічних символів (14 біт). Кадр повинен передаватися безперервно. Якщо при передачі кадру виявляється пауза тривалістю більше 1,5 шістнадцятирічних символи (6 біт), то вважається, що кадр містить помилку і повинен бути відхилений приймаючим модулем. Ці величини пауз повинні строго дотримуватися при швидкостях нижче 19200 біт / с, однак при більш високих швидкостях рекомендується використовувати фіксовані значення паузи, 1,75 мс і 750 мкс відповідно.

Контроль помилок. У режимі RTU є два рівня контролю помилок в повідомленні:

- контроль паритету для кожного байта (опційно);
- контроль кадру в цілому за допомогою CRC методу.

CRC метод використовується незалежно від перевірки паритету. Значення CRC встановлюється в провідному пристрої перед передачею. Після отримання повідомлення обчислюється CRC для всього повідомлення і порівнюється з його значенням, зазначеним у поле CRC кадру. Якщо обидва значення збігаються, вважається, що повідомлення не містить помилки.

Стартові, стопові біти і біт паритету в обчисленні CRC не беруть участь.

Прикладний рівень. Прикладний рівень Modbus RTU версії 1.1a описаний в Modbus. Він забезпечує комунікацію між пристроями типу "ведучий / ведений". Прикладний рівень є незалежним від фізичного і канального, зокрема, він може використовувати протоколи Ethernet TCP / IP (Modbus TCP / IP), Modbus Plus (многомастерная мережу з передачею маркера), інтерфейси RS-232, RS-422, RS-485, оптоволоконні, радіоканали і інші фізичні середовища для передачі сигналів.

Прикладний рівень Modbus заснований на запитах за допомогою кодів функцій. Код функції вказує відомому пристрою, яку операцію він повинен виконати.

При використанні протоколу прикладного рівня з різними протоколами транспортного і канального рівня зберігається незмінним основний блок Modbus-повідомлення, разом із кодом функції і дані (цей блок називається PDU - "Protocol

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		28

Data Unit" - "елемент даних протоколу"). До блоку PDU можуть додаватися додаткові поля при використанні його в різних промислових мережах і тоді він називається "ADU" - "Application Data Unit" - "елемент даних програми".

Коди функцій. Стандартом Modbus передбачені три категорії кодів функцій: встановлені стандартом, що задаються користувачем і зарезервовані.

Коди функцій є числами в діапазоні від 1 до 127. Коди в діапазоні від 65 до 72 і від 100 до 110 відносяться до заданим користувачем функцій, в діапазоні від 128 до 255 зарезервовані для пересилання кодів помилок у відповідному повідомленні. Код «0» не використовується.

Коди помилок використовуються веденим пристроєм, щоб визначити, яку дію зробити для їх обробки. Значення кодів і їхній зміст описані в стандарті на Modbus RTU.

Поле даних (рис. 1.12) в повідомленні, надісланому від ведучого пристрою відомому, містить додаткову інформацію, яку ведене використовує, щоб виконати функцію, зазначену в поле «код функції». Поле даних може містити значення станів дискретних входів / виходів, адреси регістрів, з яких треба зчитувати (записувати) дані, кількість байт даних, посилання на змінні, кількість змінних, код підфункцій і т. д.

Якщо ведений нормально виконав прийняту від провідного функцію, то у відповіді поле «код функції» містить ту ж інформацію, що і в запиті. В іншому випадку ведений видає код помилки. У разі помилки код функції у відповіді дорівнює коду функції в запиті, збільшеному на 128.

Зміст поля даних. У повідомленні ведучого пристрою відомому поле даних містить додаткову інформацію, необхідну для виконання зазначеної функції. Наприклад, якщо код функції вказує, що необхідно зчитати дані з групи регістрів пристрою введення (код функції 03 hex), то поле даних містить адресу початкового регістра і кількість регістрів. Якщо ведучий пристрій посилає команду запису даних в групу регістрів (код функції 10 hex), то поле даних повинне містити адресу початкового регістра, кількість регістрів, кількість байтів даних і дані для запису в регістр.

Конкретний зміст поля даних встановлюється стандартом для кожної функції окремо. У деяких повідомленнях поле даних може мати нульову довжину.

1.6 Постановка задачі проектування

Метою даного проекту є розробка апаратної частини мережі передачі даних на основі інтерфейсу RS-485 та протоколу передачі даних.

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм..	Лист	№ докум.	Підпис	Дата		29

У конкретному випадку, задача полягає в проектуванні системи передачі даних, що виконує наступні функції:

- Збір даних з датчиків, розташованих в промисловому приміщенні віддалено від вузла збору даних;
- Передача даних на вузол збору даних по інтерфейсу RS-485;
- Передача даних від вузла збору даних на комп'ютер користувача з метою збору та моніторингу показань датчиків.

Для виконання поставлених задач необхідно:

- Розробити алгоритм роботи системи передачі даних.
- Розробити структурну та функціональну схеми системи передачі даних;
- Провести вибір елементної бази та розробити принципову електричну схему блоків системи передачі даних;
- Розробити протокол передачі даних та програмне забезпечення.

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		30

2 РОЗРОБЛЕННЯ ТА РОЗРАХУНКИ ЕЛЕКТРИЧНИХ СХЕМ ВУЗЛІВ І БЛОКІВ МЕРЕЖІ ПЕРЕДАЧІ ДАНИХ

2.1 Розробка алгоритму роботи

Задачею системи передачі даних є збір даних з датчиків, що розташовані в промисловому приміщенні та їх подальшою передачею на комп'ютер користувача. Відповідно до принципів роботи мереж передачі даних на основі інтерфейсу RS-485, що були розглянуті в літературному огляді, система передачі даних має складатися з ведучого «Master» пристрою, в конкретному випадку це вузол збору даних, та ведених «Slave» пристроїв-датчиків, що здійснюють опитування сенсорів фізичних величин і пересилають інформацію на вузол збору даних. Відповідно до поставленої задачі було розроблено алгоритми роботи вузла збору даних та датчиків.

Робота вузла збору даних полягає в циклічному опитуванні датчиків і пересилці одержаних показань через комунікаційний порт на комп'ютер. На початку роботи відбувається ініціалізація пристрою, початкова установка параметрів, після чого починається опитування датчиків. Пристрій вузла збору даних по лінії даних інтерфейсу RS-485 надсилає повідомлення, в якому міститься адреса датчика, що має у відповідь надіслати показання сенсора фізичної величини (напруга, температура, тиск, тощо). Після надсилання запиту витримується пауза протягом якої очікується відповідь від датчика. Якщо відповідь надійшла, то проводиться аналіз отриманого пакету даних, проводиться перевірка на цілісність одержаних даних – розрахунок контрольної суми. Одержані дані зберігаються в пам'ять вузла збору даних, і проводиться опитування наступного датчика. Якщо під час паузи відповідь від датчика не надійшла, то інформаційний пакет вважається втраченим, і робиться відповідна відмітка в пам'яті вузла збору даних. Після завершення опитування всіх датчиків вузол збору даних формує єдиний інформаційний пакет з показаннями всіх датчиків за поточний цикл опитування і надсилає його через інтерфейс на комп'ютер користувача. Після цього проводиться очистка буфера пам'яті, що містить показання датчиків, і починається новий цикл опитування датчиків.

Алгоритм роботи пристрою-датчика наступний. Після увімкнення проводиться ініціалізація пристрою, починається циклічне зняття показань сенсора фізичної величини і проводиться прослуховування лінії даних RS-485. Якщо від вузла збору даних надійшло повідомлення, що містить адресу датчика,

					ЕлТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		31

то датчик формує пакет даних і надсилає його у відповідь центральному вузлу збору даних. Інформаційний пакет, що надсилається, містить адресу датчика, показання сенсора, контрольну суму.

Схему алгоритму роботи вузла збору даних проектованої системи передачі даних представлено на рисунку 2.1.

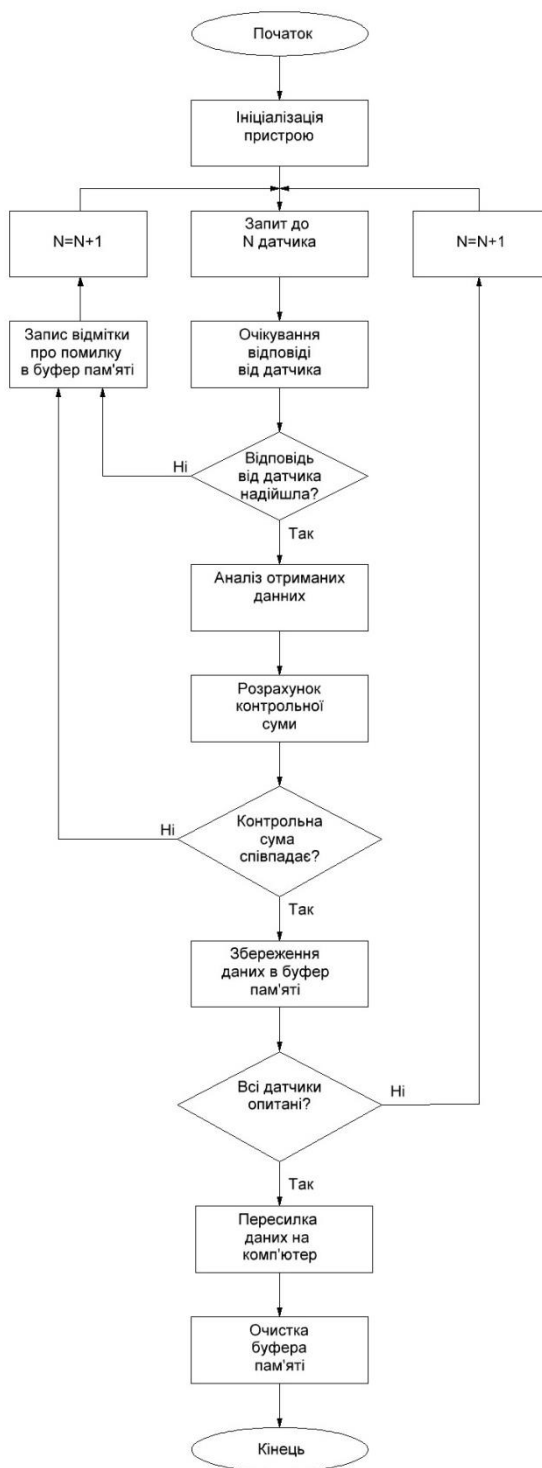


Рисунок 2.1 – Блок-схема алгоритму роботи вузла збору даних

2.2 Розробка структурної схеми

Систему передачі даних можна розділити на дві частини: 1) центральний вузол збору даних (ведучий пристрій) та 2) група датчиків (ведені пристрої). Кількість датчиків може варіюватися в діапазоні від 1 до 32, що задається специфікацією інтерфейсу. Структурна схема системи передачі даних представлена на рис. 2.2. На схемі показаний центральний вузол збору даних (A1) та один датчик (A2).

Центральний вузол збору даних являється ведучим пристроєм системи. Центральний вузол збору даних складається з наступних структурних блоків: перетворювач інтерфейсу RS-485/UART, пристрій керування – мікроконтролер та перетворювач інтерфейсу UART/USB. Пристрій керування, побудований на базі мікроконтролера, виконує функції прийому пакетів даних, що надходять по лінії даних RS-485, проводить перевірку даних на пошкодження, зберігає дані в буфер пам'яті та передає їх на комп'ютер по завершенню цикла опитування датчиків. Перетворювач інтерфейсу RS-485/UART виконує з'єднання мікроконтролера з лінією даних RS-485. Перетворювач інтерфейсу UART/USB призначений для передачі даних від вузла збору даних до комп'ютера.

Ведений пристрій – датчик має наступні структурні блоки: сенсор (датчик) фізичної величини, наприклад температури, тиску, струму, напруги, тощо, пристрій керування – мікроконтролер та перетворювач інтерфейсу UART/RS-485.

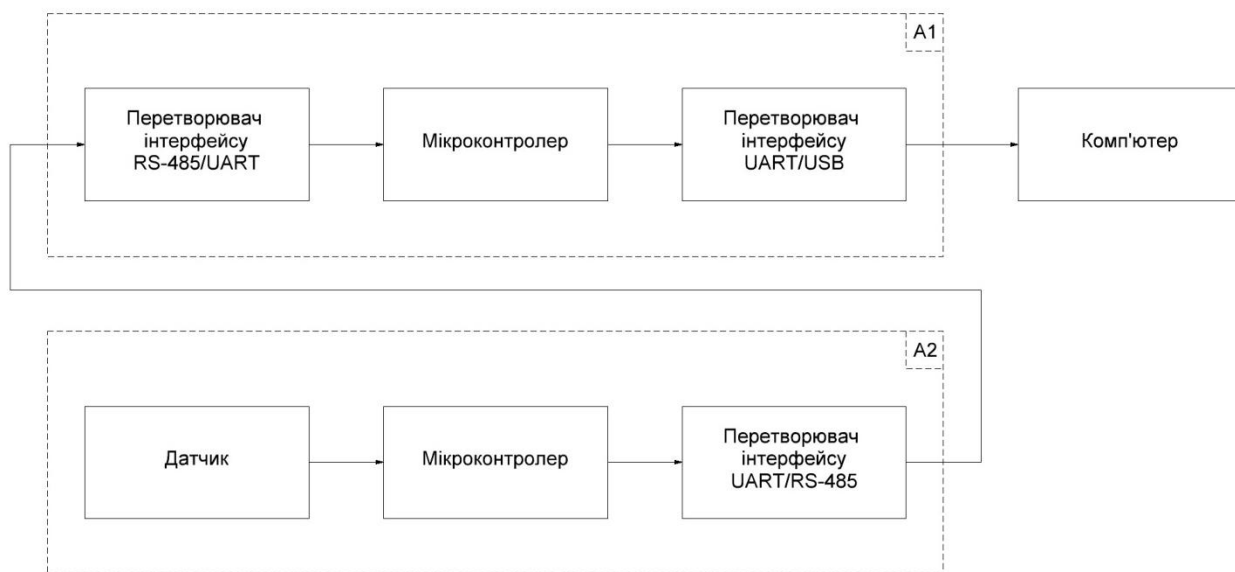


Рисунок 2.2 – Структурна схема системи передачі даних

2.3 Розробка функціональної схеми

Відповідно до структурної схеми рис.2.2 було розроблено функціональну електричну схему системи передачі даних, що представлена на рис.2.3.

Центральний вузол збору даних побудований на базі мікроконтролера. Одержання даних з лінії RS-485 та передача даних на комп'ютер відбувається за допомогою UART інтерфейсу, таким чином для реалізації функцій вузла збору даних мікроконтролер повинен бути обладнаний двома апаратними UART-портами. Оскільки для захисту даних від помилок при передачі по каналу зв'язку передбачається застосування контрольної суми CRC, то важливою функцією, яку має містити мікроконтролер, є апаратний блок розрахунку CRC. Зв'язок мікроконтролера з комп'ютером користувача реалізовано за допомогою USB інтерфейсу з використанням спеціалізованої мікросхеми-перетворювача інтерфейсу USB/UART. Зв'язок мікроконтролера центрального вузла збору даних по інтерфейсу RS-485 фізично реалізується з використанням мікросхеми перетворювача інтерфейсів UART/RS-485. Обмін даними з веденими пристроями – датчиками по інтерфейсу RS-485 відбувається по двом лініям даних – А та В.

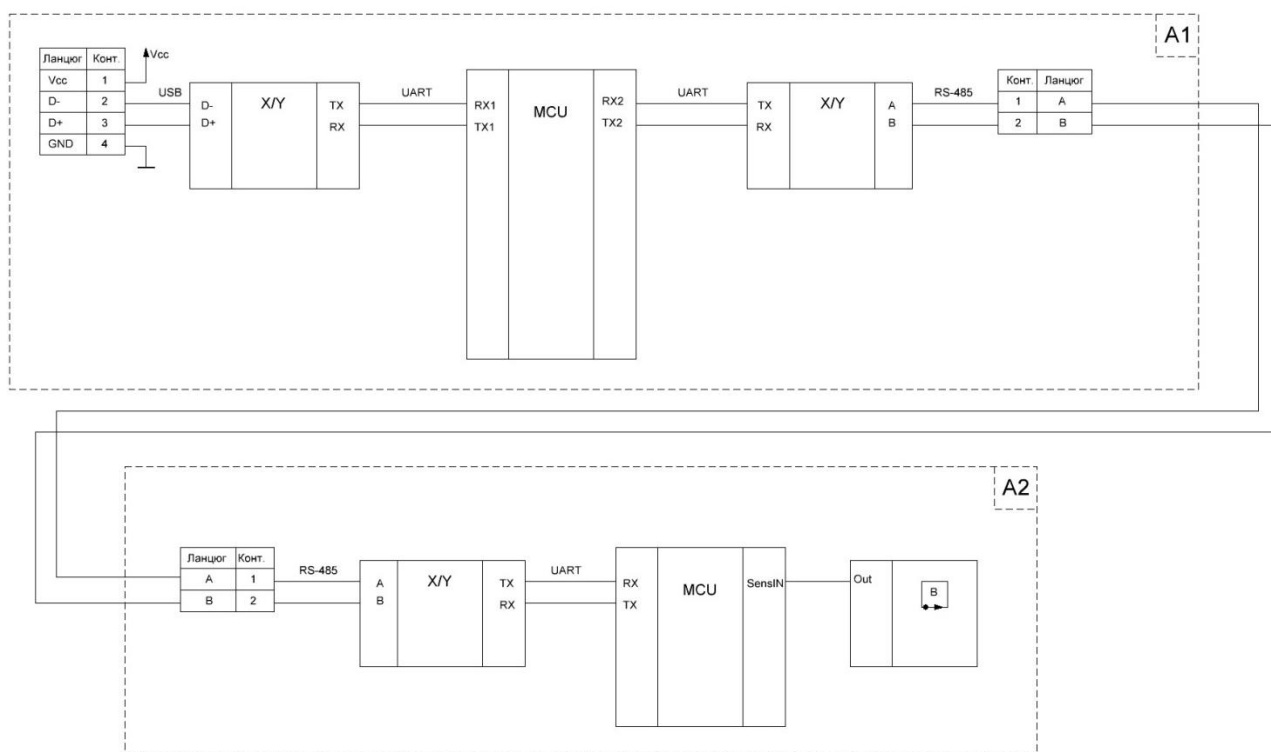


Рисунок 2.3 – Функціональна схема системи передачі даних

Основою датчика - підлеглого пристрою системи передачі даних є мікроконтролер, що здійснює опитування сенсора та по запиту від центрального вузла збору даних формує інформаційний пакет та надсилає по лінії RS-485. Аналогічно до ведучого пристрою, мікроконтролер датчика передає дані по UART з використанням перетворювача інтерфейсів UART/RS-485. Пристрій-датчик в системі призначений для вимірювання певної фізичної величини, при чому в залежності від потреб користувача ця фізична величина може бути різною. Тому система передбачає можливість використання широкого діапазону сенсорів з аналоговим та цифровим виходом, що можуть бути підключені до мікроконтролера, наприклад сенсори температури, вологості, струму, напруги, освітлення, цифрові акселерометри, гіроскопи, магнітометри, кінцеві вимикачі, тощо. У даному проекті як приклад розглядається підключення до мікроконтролера сенсора температури з цифровим виходом.

2.4 Вибір елементної бази

Вибір елементної бази для розробки системи передачі даних проводився з урахуванням наступних критеріїв: невисока ціна компонентів, доступність на ринку, сумісність з платформою Arduino.

Вибір мікроконтролера (центрального вузла збору даних). Мікроконтролер вузла збору даних повинен мати високу тактову частоту для опитування датчиків та обробки даних, два апаратних інтерфейси UART для обміну даними з комп'ютером та датчиками, апаратний блок розрахунку контрольної суми CRC. Переліченим вимогам в відповідає мікроконтролер STM32F103C8T6. Призначення виводів мікроконтролера показане на рис. 2.4.

Таблиця 2.1 - Характеристики мікроконтролера STM32F103C8T6:

Ядро	32-бітове Cortex-M3
Flash-пам'ять	128 КБ
RAM	8 КБ
Комунікаційні інтерфейси	2xSPI, 2xI2C, 3xUSART
АЦП	16-канальний 12-бітний
ЦАП	двоканальний 12-розрядний
Напруга живлення	2,0 ... 3,6 В

STM32F103C8T6

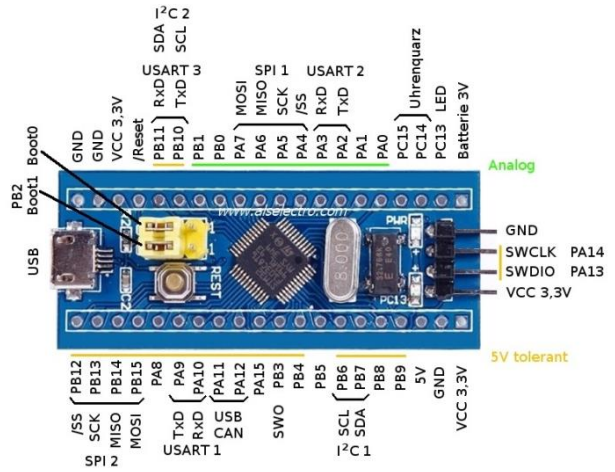
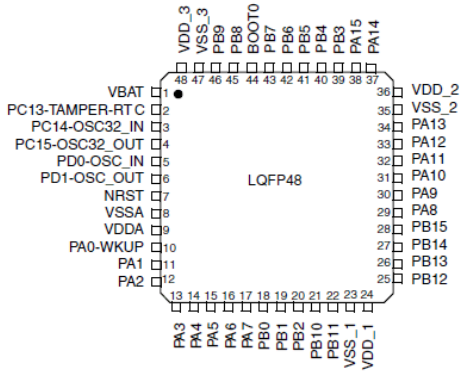


Рисунок 2.4 – Призначення виводів мікроконтролера STM32F103C8T6 (зліва) та вигляд плати мікроконтролера (справа)

Вибір мікроконтролера (датчик). Вимоги до мікроконтролера, що виконує функції опитування сенсора та передачі даних в датчику нижчі порівняно з мікроконтролером блока збору даних, тому для використання обрано мікроконтролер STM32F030F4P6.

STM32F030

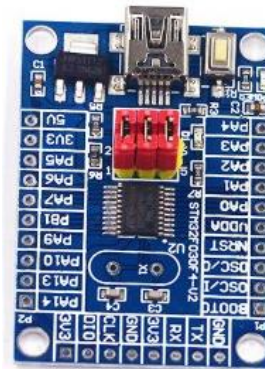
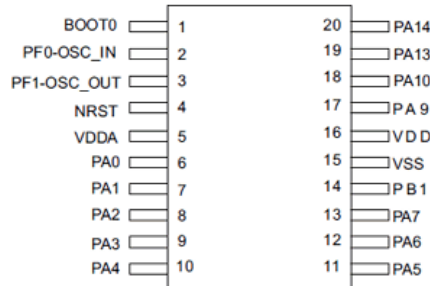


Рисунок 2.5 – Призначення виводів мікроконтролера STM32F030F4P6 (зліва) та вигляд плати мікроконтролера (справа).

Таблиця 2.2 - Характеристики мікроконтролера STM32F030F4P6:

Ядро	32-біт ARM Cortex-M0
Тактова частота	48 МГц
Обсяг пам'яті програм	16 КБ
Обсяг RAM	4k x 8
Вбудовані інтерфейси	I2C, SPI, UART
Вбудована периферія	DMA, POR, PWM, WDT

Вибір мікросхеми перетворювача інтерфейсу UART/RS-485. Для забезпечення передачі даних по інтерфейсу RS-485 використовується мікросхема-перетворювач інтерфейсу SP485. Зовнішній вигляд і призначення виводів мікросхеми показані на рис. 3.6.

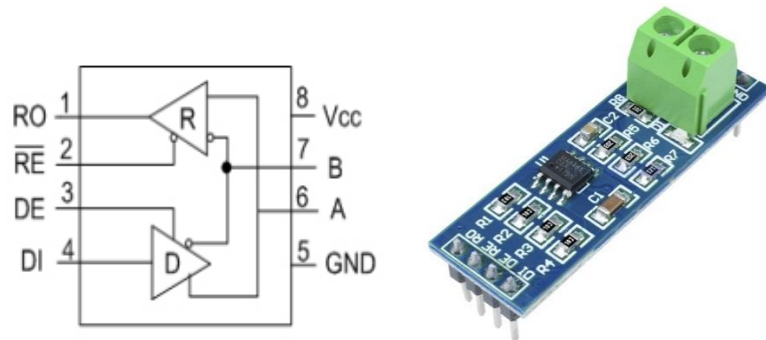


Рисунок 2.6 – Призначення виводів мікросхеми SP485 (зліва) та модуль інтерфейсу RS-485 (справа).

Таблиця 2.3 - Характеристики мікросхеми SP485:

Швидкість (макс.), Мбіт / с	5
Інтерфейс	RS-485
Пристроїв на шині	32
Rx / Tx Enable	Так
V _{CC} , В	від 4,75 до 5,25
I _{CC} , мА	0,9

Вибір схеми перетворювача інтерфейсу UART/USB. Для підключення мікроконтролера до комп'ютера через інтерфейс USB використовується мікросхема CP2102 (рис. 2.7) - перетворювач інтерфейсу USB в UART.

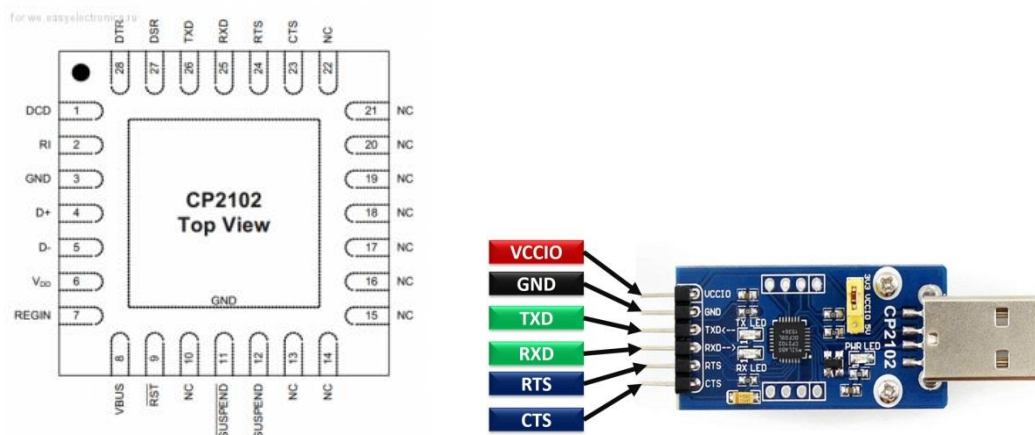
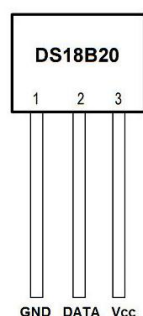


Рисунок 2.7 – Призначення виводів мікросхеми CP2102 (зліва) та модуль на основі мікросхеми (справа)

Характеристики мікросхеми CP2102:

- Чіп CP2102 від Silicon Labs
- Швидкість обміну даними по UART 300Біт / сек - 1 Мбіт / сек
- Буфер читання 576 байт, записи 640 байт
- Підтримка USB 2.0 12Мбіт / сек
- Підтримка режиму SUSPENDED USB
- Вбудований стабілізатор живлення 3.3В 100мА
- EEPROM з конфігураційними параметрами 1024 байт

Вибір датчика температури. В якості сенсора температури використовується мікросхема DS18B20 - цифровий вимірювач температури, з роздільною здатністю 9 - 12 розрядів. Параметри контролю можуть бути задані користувачем і збережені в енергонезалежній пам'яті датчика. DS18B20 обмінюється даними з мікроконтролером по однопровідній лінії зв'язку, використовуючи протокол інтерфейсу 1-Wire. Живлення датчик може отримувати безпосередньо від лінії даних, без використання зовнішнього джерела.



DS18B20



Рисунок 2.8 – Призначення виводів мікросхеми DS18B20 (зліва) та плата сенсора на основі мікросхеми (справа)

Характеристики мікросхеми DS18B20:

- Напруга живлення: 3V-5.5V;
- Протокол обміну даними: 1-Wire;
- Спосіб підключення: прямий / по одній лінії з паразитним живленням;
- Роздільна здатність перетворення температури: 9 біт - 12 біт;
- Діапазон вимірювання температури: від -55 до +125 °C;
- Період вимірювання температури при точності 12 біт: 750 мС;
- Тип індексації на лінії 1-Wire: унікальна 64-бітова адреса.

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		38

2.5 Розробка схеми електричної принципової

Підключення мікроконтролера центрального вузла збору даних.

Схема підключення мікроконтролера STM32F103C8T6 приведена на рис. 3.11.

Мікроконтролер розміщено в корпусі LQFP48. Напруга живлення складає 3,3В, до виводів живлення контролера підключені конденсатори ємністю 0,1мкФ, зашунтовані на «землю» для усунення перешкод на лінії живлення. Тактовий сигнал частотою 8 МГц подається на виводи OCS1 та OSC2 від кварцового резонатора Z1. Виводи TXD2 та RXD2 (12 та 13) підключені до перетворювача інтерфейсів UART/USB. Виводи TXD1 та RXD1 (20 та 21) підключені до перетворювача інтерфейсів UART/RS-485. Вивід PA11 призначений для вибору режиму роботи прийом/передача мікросхеми SP485.

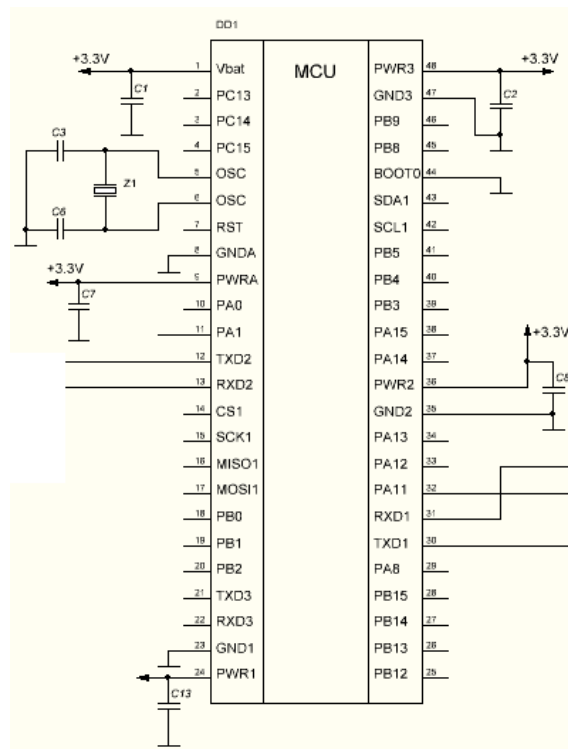


Рисунок 2.9 – Схема підключення мікроконтролера STM32F103C8T6

Підключення мікроконтролера датчика.

Мікроконтролер STM32F030F4P6 (рис. 2.10) розміщено в корпусі TSSOP20. Обмін даними з мікроконтролером центрального вузла збору даних виконується по інтерфейсу UART, виводи TXD та RXD (17 та 18) підключені до перетворювача інтерфейсів UART/RS-485, контроль режиму роботи прийом/передача мікросхеми SP485 виконується на допомогою сигналу на виводі PA13.

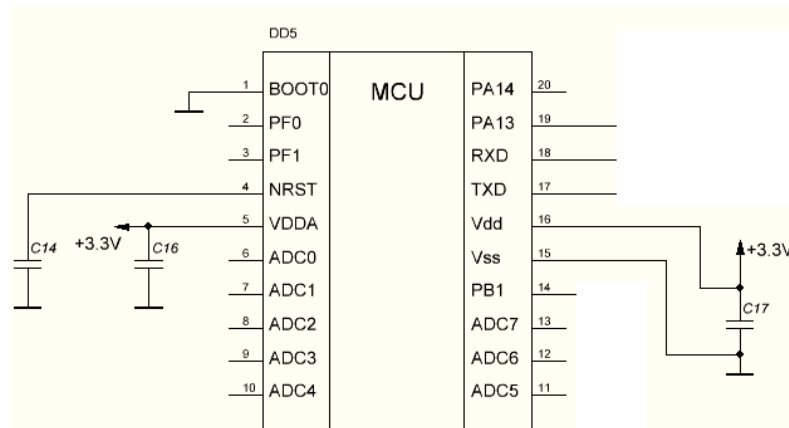


Рисунок 2.10 – Схема підключення мікроконтролера STM32F030F4P6

У випадку підключення до мікроконтролера сенсора температури DS18B20 використовується один вивід мікроконтролера PB1 (14). Оскільки пристрій датчика передбачає використання різних сенсорів, то в інших випадках можуть бути задіяні виводи мікроконтролера, що призначені для роботи аналогово-цифрового перетворювача, інтерфейсів SPI та I2C.

Підключення мікросхеми перетворювача інтерфейсу RS485. Схема підключення мікросхеми SP485 представлена на рис. 2.11. Живлення мікросхеми складає +5В. Обмін даними з мікроконтролером відбувається по інтерфейсу UART. Вибір режим роботи прийом/передача здійснюється за допомогою подачі керуючого сигналу на входи RE та DE, це зумовлено тим, що скільки інтерфейс RS-485 є напівдуплексним.

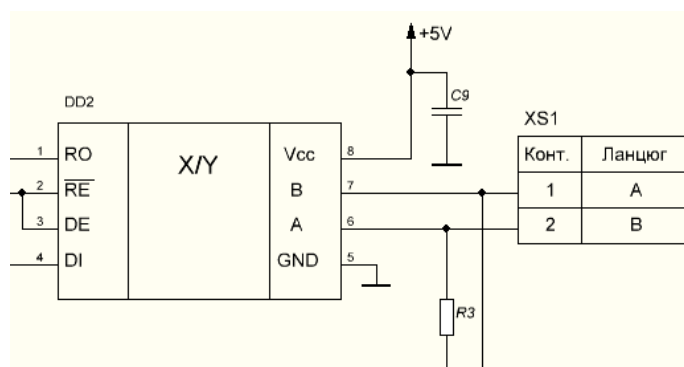


Рисунок 2.11 – Схема підключення мікросхеми SP485

Підключення мікросхеми перетворювача інтерфейсу UART/USB. Типова схема підключення мікросхеми CP2102 представлена на рис. 2.12. Живлення мікросхеми складає 5 В. Виводи D+ та D- є виводами обміну даними і підключаються до USB роз'єму. Виводи TXD та RXD підключені до відповідних виводів інтерфейсу UART мікроконтролера.

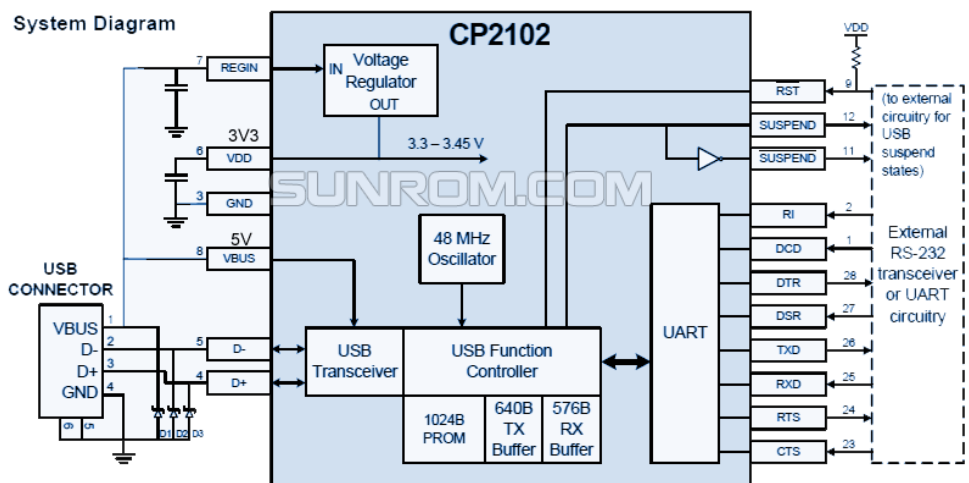


Рисунок 2.12 – Схема підключення мікросхеми CP2102

Підключення сенсора температури DS18B20. Схема підключення сенсора температури DS18B20 представлена на рис. 2.13. Живлення мікросхеми складає 3,3 В. У мікросхемі DS18B20 для обміну даними використовують спеціалізований протокол 1-Wire корпорації Dallas. Особливість використання такого протоколу полягає в тому, що для лінії зв'язку потрібен резистор-підтяжка 4,7 кОм, тому всі пристрої фізично підключені до однієї загальної шини і використовують вихід з трьома станами або вихід типу відкритий стік.

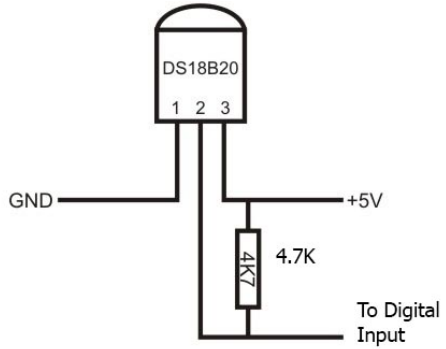


Рисунок 2.13 – Схема підключення сенсора DS18B20

Підключення регулятора напруги. Враховуючи те, що напруги живлення компонентів схеми відрізняються і складають +5 В та +3,3 В, то необхідним є застосування мікросхеми LD1117-3.3 – лінійного регулятора напруги з вихідною напругою 3,3 В. Мікросхема має три виводи – вхід, загальний вивід, вихід. На вході мікросхеми підключений електролітичний конденсатор ємністю 100 мкФ для фільтрації перепадів напруги живлення.

3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Протокол передачі даних

При побудові мережі використовується принцип організації ведений-ведений (master-slave). У мережі може бути присутнім тільки один ведучий вузол і кілька ведених вузлів. В якості ведучого вузла виступає персональний центральний вузол збору даних, підключений до комп'ютера користувача, ведені пристрої - датчики. При даній організації ініціатором циклів обміну може виступати виключно ведучий вузол.

Запити ведучого вузла - індивідуальні (адресуються до конкретного вузла). Ведені вузли здійснюють передачу, відповідаючи на індивідуальні запити ведучого вузла. При виявленні помилок в отриманні запитів, або неможливості виконання отриманої команди, ведений вузол, як відповідь, генерує повідомлення про помилку.

Формати повідомлень. Протокол обміну має чітко визначені формати повідомлень. Нижче описується формат байт і формат кадрів. Дотримання форматів забезпечує правильність і стійкість функціонування мережі.

1) Формат байта. Прилад налаштований на роботу в форматі 8N1 - 8 біт даних, без контролю паритету, 1 стоп біт.

Передача байт здійснюється на швидкостях, кратних 1200 біт / с - 1200, 2400, 4800, 9600, 19200.

2) Формат кадру. Довжина кадру не повинна перевищувати 8 байт. Контроль закінчення кадру здійснюється за допомогою інтервалу мовчання, довжиною не менше часу передачі 3,5 байт. Формат кадру наведено в табл. 3.1.

Кадр повинен передаватися як безперервний потік байт. Правильність прийняття кадру додатково контролюється перевіркою контрольної суми.

Таблиця 3.1 – Формат кадру

Інтервал мовчання $\geq 3,5$ байт	
Адреса	1 байт
Функція	1 байт
Дані	до 4 байт
Контрольна сума	2 байта
Інтервал мовчання $\geq 3,5$ байт	

3) Генерація і перевірка контрольної суми. Контрольна сума CRC16 є циклічним кодом перевірки на основі неприводимого полінома A001h. Передаючий пристрій формує контрольну суму для всіх байт переданого повідомлення. Приймаючий пристрій аналогічним чином формує контрольну суму для всіх байт прийнятого повідомлення і порівнює її з контрольною сумою, прийнятої від передавального пристрою. При розбіжності сформованої і прийнятої контрольних сум генерується повідомлення про помилку. Поле контрольної суми займає два байти. Контрольна сума в повідомленні передається молодшим байтом вперед. Нижче наводиться опис алгоритмічного способу формування CRC16.

Контрольна сума формується за наступним алгоритмом:

1. Завантаження CRC регістра (16 біт) одиницями (0xFFFF).
2. Виключне АБО з першими 8 бітами байта повідомлення і вмістом CRC регістра.
3. Зсув результату на один біт вправо.
4. Якщо зсунутий біт = 1, що виключне АБО вмісту регістра зі значенням 0xA001.
5. Якщо зсунутий біт = 0, повторити крок 3.
6. Повторювати кроки 3, 4, 5 поки не будуть виконані 8 зсувів.
7. Виключне АБО з наступними 8 бітами байта повідомлення і вмістом CRC регістра.
8. Повторювати кроки 3 - 7 поки всі байти повідомлення не будуть оброблені.
9. Кінцевий вміст регістра буде містити контрольну суму.

3.2 Лістинг коду програми

Оголошення змінних. Підключення бібліотек:

```

//*****
#define OBJ_SZ 123 //кількість об'єктів
#define SETUP 4 //кількість даних в масиві
//PARAMETERS ARRAY 0 PARAMETER = MODBUS ADDRESS
unsigned char SET_PAR[SETUP]; //0-елемент це адреса

//OBJECT ARRAY WHERE READING AND WRITING OCCURS
int res_table[OBJ_SZ]; //масив з об'єктами, місце для зчитування і запису

//buffer uart

```

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		43

```
#define BUF_SZ 256 //розмір буфера
#define MODBUS_WRD_SZ (BUF_SZ-5)/2 //максимальна кількість регістрів у відповіді
```

```
//uart structure
typedef struct {
unsigned char buffer[BUF_SZ]; //буфер
unsigned int rxtimer; //часова затримка
unsigned char rxcnt; //кількість прийнятих символів
unsigned char txcnt; //кількість переданих символів
unsigned char txlen; //довжина посилки на відправку
unsigned char rxgar; //кінець прийому
unsigned char protocol; //тип протоколу
unsigned char delay; //затримка
} UART_DATA;
```

```
UART_DATA uart3,uart1; //структури для відповідних USART
```

```
/**
*****
*****/
```

Ініціалізація таймера для формування часової затримки:

```
void SetupTIM6()
{
    NVIC_InitTypeDef NVIC_InitStructure;
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;

    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM6, ENABLE);
    TIM_DeInit(TIM6);

    //0.0001 sec setup APB=36Mhz/(36*100)
    TIM_TimeBaseStructure.TIM_Prescaler= 36; //на цю величину ділиться частота шини
    TIM_TimeBaseStructure.TIM_ClockDivision=TIM_CKD_DIV1; //частота без ділення складає 36Мгц
```

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		44

```
TIM_TimeBaseStructure.TIM_CounterMode=TIM_CounterMode_Up;//підрахунок  
вгору
```

```
TIM_TimeBaseStructure.TIM_Period=100;//значення, до якого рахує таймер  
TIM_TimeBaseInit(TIM6, &TIM_TimeBaseStructure);  
TIM_ClearFlag(TIM6, TIM_FLAG_Update);  
TIM_ITConfig(TIM6, TIM_IT_Update, ENABLE);  
TIM_Cmd(TIM6, ENABLE);
```

```
// Настрналаштування переривань  
NVIC_InitStructure.NVIC_IRQChannel = TIM6_IRQn;  
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;  
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;  
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;  
NVIC_Init(&NVIC_InitStructure);
```

```
TIM_ITConfig(TIM6, TIM_IT_Update, ENABLE);
```

```
}
```

```
//*****
```

Функції переривання для передачі даних по UART:

```
void USART3_IRQHandler(void)
```

```
{
```

```
    //Receive Data register not empty interrupt
```

```
    if(USART_GetITStatus(USART3, USART_IT_RXNE) != RESET)
```

```
    {
```

```
        USART_ClearITPendingBit(USART3, USART_IT_RXNE)//очистка
```

ознаки переривання

```
        uart3.rxtimer=0;
```

```
        if(uart3.rxcnt>(BUF_SZ-2))
```

```
        uart3.rxcnt=0;
```

```
        uart3.buffer[uart3.rxcnt++]=USART_ReceiveData (USART3);
```

```
}
```

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		45

```

//Transmission complete interrupt
if(USART_GetITStatus(USART3, USART_IT_TC) != RESET)
{
    USART_ClearITPendingBit(USART3, USART_IT_TC);//очистка ознаки
переривання
    if(uart3.txcnt<uart3.txlen)
    {
        USART_SendData(USART3,uart3.buffer[uart3.txcnt++]);//передача
    }
    else
    {
        //передача закінчилася, знаття високого рівня з RS485 TXE
        uart3.txlen=0;
        GPIO_WriteBit(GPIOB,GPIO_Pin_2,Bit_RESET);
        GPIO_WriteBit(GPIOD,GPIO_Pin_10,Bit_RESET);
        USART_ITConfig(USART3, USART_IT_RXNE, ENABLE);
        USART_ITConfig(USART3, USART_IT_TC, DISABLE);
        TIM_ITConfig(TIM6,TIM_IT_Update,ENABLE);
    }
}

}

void net_tx3(UART_DATA *uart)
{
    if((uart->txlen>0)&(uart->txcnt==0))
    {
        USART_ITConfig(USART3, USART_IT_RXNE, DISABLE);//вихід з
переривання на прийом
        USART_ITConfig(USART3, USART_IT_TC, ENABLE);//вмикаємо на
закінчення передачі

        // вмикаємо rs485 на передачу
        GPIO_WriteBit(GPIOB,GPIO_Pin_2,Bit_SET);
    }
}

```

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		46

```

GPIO_WriteBit(GPIOD,GPIO_Pin_10,Bit_SET);

USART_SendData(USART3, uart->buffer[uart->txcnt++]);
}
}
//*****

```

Робота основного циклу програми:

```

//Main loop
while(1)
{

if(uart3.rxgap==1)//часова затримка
{
MODBUS_SLAVE(&uart3);//перевірка, якщо пакет за адресою і не
пошкоджений, то формується відповідь
net_tx3(&uart3);//якщо є ознака готовності посилки, по ініціюємо передачу
}

if(uart1.rxgap==1)
{
MODBUS_SLAVE(&uart1);
net_tx1(&uart1);
}

}

//*****

```

Обмін даними з датчиком температури по шині 1-Wire.

Ініціалізація UART для роботи в режимі 1-wire:

```

void OW_Init(USART_TypeDef* USARTx) {
GPIO_InitTypeDef GPIO_InitStructure;
USART_InitTypeDef USART_InitStructure;

if (USARTx == USART2) {

```

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA |  
RCC_APB2Periph_AFIO, ENABLE);
```

```
GPIO_InitStruct.GPIO_Pin = GPIO_Pin_2;  
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_AF_PP;  
GPIO_InitStruct.GPIO_Speed = GPIO_Speed_50MHz;
```

```
GPIO_Init(GPIOA, &GPIO_InitStruct);
```

```
GPIO_InitStruct.GPIO_Pin = GPIO_Pin_3;  
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_IN_FLOATING;  
GPIO_InitStruct.GPIO_Speed = GPIO_Speed_50MHz;
```

```
GPIO_Init(GPIOA, &GPIO_InitStruct);
```

```
RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2, ENABLE);  
}
```

```
USART_InitStructure.USART_BaudRate = 115200;  
USART_InitStructure.USART_WordLength = USART_WordLength_8b;  
USART_InitStructure.USART_StopBits = USART_StopBits_1;  
USART_InitStructure.USART_Parity = USART_Parity_No;  
USART_InitStructure.USART_HardwareFlowControl =  
    USART_HardwareFlowControl_None;  
USART_InitStructure.USART_Mode = USART_Mode_Tx;
```

```
USART_Init(USARTx, &USART_InitStructure);  
USART_Cmd(USARTx, ENABLE);
```

```
RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA1, ENABLE);  
}
```

```
uint8_t OW_Reset(USART_TypeDef* USARTx) {  
    uint8_t ow_presence;  
    USART_InitTypeDef USART_InitStructure;
```

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		48

```

    USART_DMACmd(USARTx, USART_DMAREq_Tx | USART_DMAREq_Rx,
DISABLE);
    USART_InitStructure.USART_BaudRate = 9600;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_HardwareFlowControl =
        USART_HardwareFlowControl_None;
    USART_InitStructure.USART_Mode = USART_Mode_Tx |
USART_Mode_Rx;
    USART_Init(USARTx, &USART_InitStructure);
    USART_ClearFlag(USARTx, USART_FLAG_TC);
    USART_SendData(USARTx, 0xf0);
    while (USART_GetFlagStatus(USARTx, USART_FLAG_TC) == RESET);
    ow_presence = USART_ReceiveData(USARTx);

```

```

    USART_InitStructure.USART_BaudRate = 115200;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_HardwareFlowControl =
        USART_HardwareFlowControl_None;
    USART_InitStructure.USART_Mode = USART_Mode_Tx |
USART_Mode_Rx;
    USART_Init(USARTx, &USART_InitStructure);
    if (ow_presence != 0xf0) {
        return 1;
    }
    return 0;
}

```

//**

Відправка команди і зчитування даних по шині 1-wire:

```

void OW_SendCommand(USART_TypeDef* USARTx, const uint8_t *command,
uint16_t len) {
    DMA_InitTypeDef DMA_InitStructure;

```

						ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм..	Лист	№ докум.	Підпис	Дата			49


```

DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t) &(USART2->DR);
DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t) command;
DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralDST;
DMA_InitStructure.DMA_BufferSize = len;
DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
DMA_InitStructure.DMA_PeripheralDataSize =
    DMA_PeripheralDataSize_Byte;
DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_Byte;
DMA_InitStructure.DMA_Mode = DMA_Mode_Normal;
DMA_InitStructure.DMA_Priority = DMA_Priority_Low;
DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
DMA_Init(DMA1_Channel7, &DMA_InitStructure);
USART_DMACmd(USART2, USART_DMAReq_Tx, ENABLE);
DMA_Cmd(DMA1_Channel7, ENABLE);
}

```

```

const uint8_t read_scratch[] = {
    OW_0, OW_0, OW_1, OW_1, OW_0, OW_0, OW_1, OW_1, // 0xcc
SKIP ROM
    OW_0, OW_1, OW_1, OW_1, OW_1, OW_1, OW_0, OW_1, // 0xbe
READ SCRATCH
    OW_R, OW_R, OW_R, OW_R, OW_R, OW_R, OW_R, OW_R,
    OW_R, OW_R, OW_R, OW_R, OW_R, OW_R, OW_R, OW_R
};
uint8_t scratch[sizeof(read_scratch)];

```

```

void OW_ReadData(USART_TypeDef* USARTx, const uint8_t *command, uint8_t
*buf, uint16_t len) {
    DMA_InitTypeDef DMA_InitStructure;
    DMA_DeInit(DMA1_Channel6);
    DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t) &(USART2->DR);
    DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t) buf;
    DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralSRC;
    DMA_InitStructure.DMA_BufferSize = len;
    DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;

```

```

DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
DMA_InitStructure.DMA_PeripheralDataSize =
    DMA_PeripheralDataSize_Byte;
DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_Byte;
DMA_InitStructure.DMA_Mode = DMA_Mode_Normal;
DMA_InitStructure.DMA_Priority = DMA_Priority_Low;
DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
DMA_Init(DMA1_Channel6, &DMA_InitStructure);
DMA_DeInit(DMA1_Channel7);
DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t) &(USART2->DR);
DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t) command;
DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralDST;
DMA_InitStructure.DMA_BufferSize = len;
DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
DMA_InitStructure.DMA_PeripheralDataSize =
    DMA_PeripheralDataSize_Byte;
DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_Byte;
DMA_InitStructure.DMA_Mode = DMA_Mode_Normal;
DMA_InitStructure.DMA_Priority = DMA_Priority_Low;
DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
DMA_Init(DMA1_Channel7, &DMA_InitStructure);
USART_DMACmd(USARTx, USART_DMAREq_Tx | USART_DMAREq_Rx,
ENABLE);
DMA_Cmd(DMA1_Channel6, ENABLE);
DMA_Cmd(DMA1_Channel7, ENABLE);

while (DMA_GetFlagStatus(DMA1_FLAG_TC6) == RESET);
}

```

ВИСНОВОК

У даному дипломному проекті розроблена мережа передачі даних на основі інтерфейсу RS-485. На основі такої мережі реалізовано систему передачі даних, що може бути використана в промисловому приміщенні на виробництві, в приватному господарстві, тощо для збору показань датчиків з їх подальшою передачею користувачеві.

Перевагами мереж на основі інтерфейсу RS-485 є висока захищеність від перешкод в каналі зв'язку, простота підключення за допомогою двохдротової лінії, довжина лінії передачі даних близько 1 км, максимальна кількість пристроїв на лінії – 32.

У першому розділі роботи було проведено літературний огляд типів та призначення промислових мереж передачі даних, здійснено аналіз перешкод при передачі даних в дротових каналах зв'язку, розглянуто принцип роботи та характеристики інтерфейсу RS-485, а також особливості побудови мереж передачі даних на його основі. Також були розглянуті особливості організації протоколів передачі даних, що використовуються в мережах на основі інтерфейсу RS-485, зокрема більш глибоко проаналізовано протокол Modbus.

З використанням інформації, викладеної в літературному огляді, було розроблено алгоритм роботи, структурну, функціональну та принципову електричну схеми системи передачі даних, реалізованої на основі мережі RS-485. Було розроблено протокол передачі даних для такої мережі та створене програмне забезпечення для роботи модулів системи передачі даних в складі мережі.

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		52

СПИСОК ЛІТЕРАТУРИ

1. Zhang P. Networked Microgrids . — Cambridge, 2021. — 234p.
2. Zurawski Richard. The Industrial Communication Technology Handbook Second Edition. — CRC Press, 2014. — 1756 p.
3. Reynders D., Mackay S., Wright E. Practical Industrial Data Communications: Best Practice Techniques. — Burlington, MA : Newnes, Elsevier, 2005. — 414 p.
4. Sen S.K. Fieldbus and Networking in Process Automation . — CRC Press, Taylor & Taylor & Francis Group, 2014. 461 p. 294
5. Wilamowski B.M., Irwin J.D. Industrial Communication Systems . — CRC Press, 2011. — 908 p.
6. Борисов А.М. Основы построения промышленных сетей автоматизации . — Челябинск: Издательский центр ЮУрГУ, 2012. – 108 с.
7. Елизаров И.А. и др. Промышленные вычислительные сети Учебное пособие. И.А. Елизаров, П.М. Оневский, В.А. Погонин, А.А. Третьяков. — Тамбов : ТГТУ, 2012. — 152 с.
8. Пупена, О. М. Промислові мережі та інтеграційні технології в автоматизованих системах : навчальний посібник / О. М. Пупена, І. В. Ельперін, Н. М. Луцька, А. П. Ладанюк. – К. : Ліра, 2011. – 552 с.
9. Промышленные сети: учебное пособие — Екатеринбург: Уральский федеральный университет, 2020. — 124 с.
10. Фетисов В.С. Протокол информационного обмена HART в измерительных и управляющих системах : Учебное пособие. — Уфа: ФОТОН, 2011. — 88 с.
11. Quick reference for RS485, RS422, RS232 AND RS423 [Электронный ресурс] - Режим доступа: <http://www.rs485.com/rs485spec.html>
12. STM32F103C8T6 datasheet. Medium-density performance line ARM®-based 32-bit MCU with 64 or 128 KB Flash, USB, CAN, 7 timers, 2 ADCs, 9 com. interfaces [Электронный ресурс] - Режим доступа: <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>
13. Правильная разводка сетей RS-485 [Электронный ресурс] - Режим доступа: <http://www.gaw.ru/html.cgi/txt/interface/rs485/app.htm>
14. RS485 — стандарт промышленных сетей. [Электронный ресурс] - Режим доступа: <https://habr.com/ru/company/milandr/blog/540084/>

					ЕЛІТ 6.172. 352 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		53