

УДК 621.391

МЕТОД НАЛАШТУВАННЯ Й УПРАВЛІННЯ РЕЖИМАМИ РОБОТИ БЕЗПРОВОДОВОЇ СЕНСОРНОЇ МЕРЕЖІ ОПТИКО-ЕЛЕКТРОННИХ СТАНЦІЙ ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЇ LORA



[І.С. Шостко](#), [О.К. Шлома](#), [Д.І. Цибульников](#)
Харківський національний університет радіоелектроніки

Abstract – A method of managing a wireless sensor network of optical-electronic stations (OES) using LoRa technology is developed. Within the framework of the system of optic-electronic stations, two subnets are functioning: the video stream transmission network and the control network, which is the subject of this work. The main requirements for the control network were the following: the ability to ensure performance in radio-electronic suppression conditions, the provision of high-range radio communication, the ability to operate in conditions of dense urban development, the network energy efficiency, and the high stealth level. Technologies available on the market for building sensor networks were analyzed. LoRa technology was chosen for these purposes considering the established requirements for the network. The current network layout was created based on EBYTE E32-433T20DT LoRa radio modules. The basis of the developed method is two classes of commands for configuring the network of optical-electronic stations. The first class of commands is network management commands, operations to add and remove devices from the general routing table. The second class of commands controls communication modules and microcontrollers, changing their configuration parameters. The OES network's control algorithms were developed based on these commands. Created commands and algorithms helped significantly facilitate and speed up the operator's initialization and network configuration process and are already used in the current layout of the optical-electronic station system.

Анотація – Розроблено метод управління безпроводовою сенсорною мережею оптико-електронних станцій (ОЕС) із застосуванням технології LoRa. У межах системи ОЕС функціонує дві підмережі: мережа передачі відеопотоку та мережа управління, якій і присвячена ця робота. Основними вимогами, що висувалися до мережі управління, були наступні: можливість забезпечувати працездатність в умовах радіоелектронного придушення, забезпечення високої дальності радіозв'язку, можливість функціонувати в умовах щільної міської забудови, енергоефективність мережі, високий рівень прихованості. Було проаналізовано наявні на ринку технології для побудови сенсорних мереж. Зважаючи на встановлені вимоги до мережі, було обрано технологію LoRa. Діючий макет мережі було побудовано на базі радіомодулів LoRa EBYTE E32-433T20DT. Основою розробленого методу є два класи команд налаштування мережі оптико-електронних станцій. Перший клас команд – це команди управління мережею, операції для додавання та видалення пристроїв із загальної таблиці пристроїв. Другий клас команд – для управління модулями зв'язку та мікроконтролерами, зміни параметрів їх налаштування. На основі цих команд було розроблено алгоритми управління мережею ОЕС. Розроблені команди та алгоритми допомогли суттєво полегшити та прискорити процес ініціалізації та налаштування мережі для оператора і вже використовуються у діючому макеті системи оптико-електронних станцій.

Вступ

За останні кілька років технології безпроводової передачі даних зазнали стрімкого розвитку. З кожним роком відмовостійкість, захищеність та енергоефективність безпроводових мереж зв'язку зростає. Цей процес постійного розвитку дає можливість покращувати вже існуючі системи та розробки. Однією з таких систем, що потребує використання сучасних технологій безпроводового зв'язку є мережа оптико-електронних станцій (ОЕС). Мережа, що об'єднує велику кількість ОЕС використовується, наприклад, для охорони державних кордонів,

особливо важливих об'єктів, при полігонних випробуваннях літаків, ракет тощо. У такій мережі необхідно керувати великою кількістю просторово розподілених стаціонарних і мобільних ОЕС. Потрібно передавати команди управління опорно-поворотним пристроям ОЕС, команди управління режимами роботи камери, а також команди для налаштування й управління режимами роботи самої мережі.

На сьогодні в умовах зростання кількості військових конфліктів у світі існує потреба в мережах ОЕС своєчасного виявлення, супроводу повітряних цілей і видачі цілевказівок засобам ураження в реальному масштабі часу. Тому розробка таких мереж і методів налаштування й управління режимами роботи самої мережі є актуальною. Особливості такої мережі ОЕС, що призначена для виявлення, супроводу повітряних цілей, технічні характеристики ОЕС і методи їх позиціонування наведено в публікаціях [1, 2]. Робота мережі можлива в умовах радіоелектронного придушення, порушення функціонування чи повного руйнування окремих вузлів. Тому пропонується використовувати технологію безпроводової сенсорної мережі (БСМ) [3-5].

Фактично, у сучасному вигляді, сенсорні мережі з'явилися в 2003 році з появою першої ревізії протоколу IEEE 802.15.4. Основна задача для БСМ – це забезпечити обмін інформацією між великою кількістю пристроїв, задіяних у процесі обробки та збору даних. На рис. 1 зображено структурну схему БСМ.

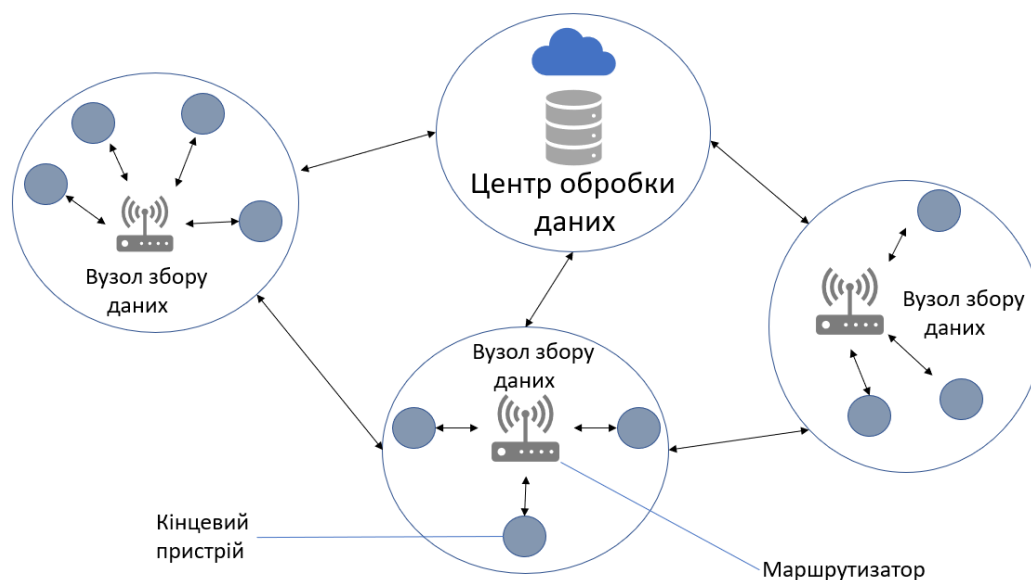


Рис. 1. Структурна схема безпроводової сенсорної мережі

Отримана інформація з датчиків передається від кінцевих пристроїв на єдиний центр обробки даних, який на основі отриманих результатів віддає команди на інші пристрої або просто зберігає їх. Маршрутизатор виконує роль мережного пристрою (шлюзу) та забезпечує комунікацію між вузлами збору та обробки даних. У ролі кінцевих пристроїв виступають ОЕС. Концепція БСМ, порівняно з проводовими мережами, дозволяє легко масштабувати вже існуючі системи, додаючи або

відключаючи деякі вузли мережі. Завдяки великій кількості вузлів, що можуть спілкуватись один з одним, мережі мають високу надійність і відмовостійкість. Якщо один вузол вийде з ладу, інший замінить його і буде виконувати функції маршрутизації.

Ключові елементи БСМ – це кінцеві пристрої, маршрутизатори, вузли обробки даних. Кінцеві пристрої можуть виконувати функції як сенсорів, так і контроль-виконавчих пристроїв. Залежно від необхідних функцій кінцевий пристрій може бути обладнано різними елементами. До загальної конфігурації кінцевого пристрою входить: елемент передачі даних (відеопотік і телеметрія), мікропроцесор для обробки даних, набір сенсорів, батарея, блоки аналого-цифрового перетворення – АЦП та ЦАП. На рис. 2 зображено структурну схему базового кінцевого пристрою.

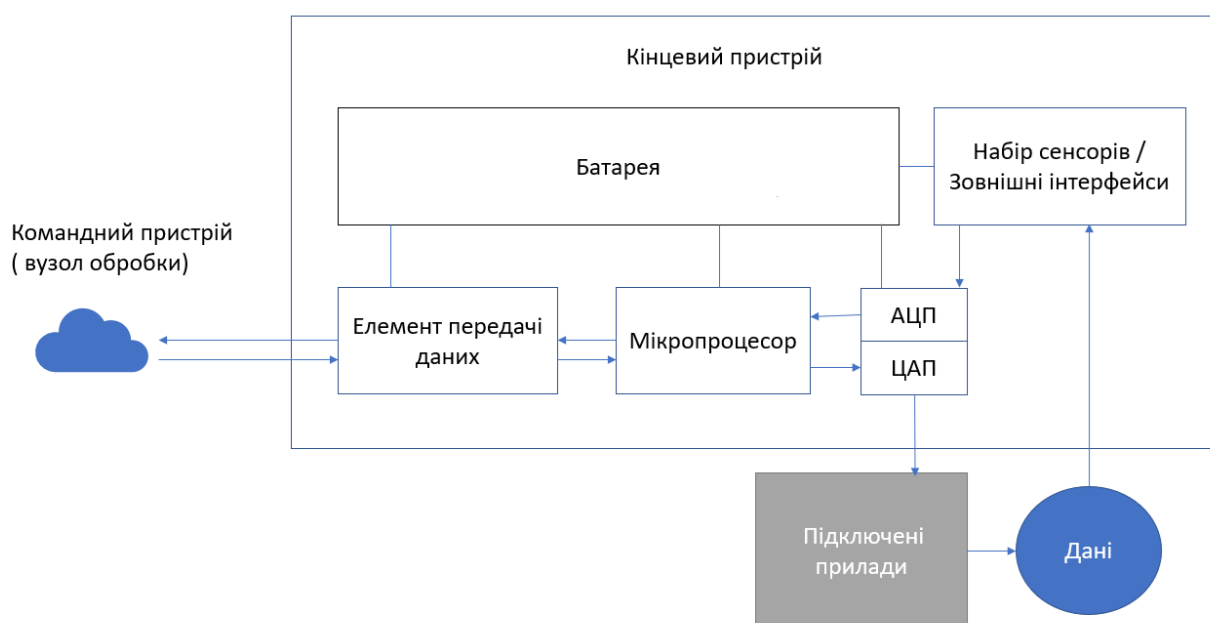


Рис. 2. Структурна схема базового кінцевого пристрою

Як підключені прилади можуть виступати серводвигуни опорно-поворотного пристрою ОЕС або елементи управління камерою. Дані поступають від підключених приладів на вбудовані сенсори, аналого-цифровий перетворювач (АЦП) перетворює сигнал з аналогового на цифровий і передає на мікропроцесор, далі мікропроцесор маніпулює цими даними і через модуль передачі даних виконує комунікацію з командним пристроєм. Відповіддю від вузла обробки даних може бути набір команд для передачі на сервопривод.

Безпроводові сенсорні мережі можуть бути самоорганізовані або керовані. Залежно від типу мережі комунікація між пристроями може здійснюватися за принципами самоорганізації та децентралізації. Таким чином, пристрої будують ad-hoc мережу, яка не має постійної структури. Пристрої починають пересилати пакети, що призначаються іншим мережним пристроям. У разі побудови керованих безпроводових сенсорних мереж комунікація між пристроями може відбуватися

двома методами. Перший метод має назву Single-hop, що означає передачу даних з центра обробки одразу на кінцевий пристрій. Другий метод – це Multi-hop, при якому посередником між центром обробки та кінцевим пристроєм можуть бути декілька сенсорних вузлів. Залежно від кількості ОЕС можливо використання першого або другого методу передачі даних у мережі.

При побудові БСМ треба брати до уваги особливості мережних пристроїв та умови застосування. Мультифункціональність сучасних пристроїв для побудови сенсорних мереж дозволяє побудувати майже усі можливі варіанти топології. Завдяки гнучкості налаштування можуть бути побудовані централізовані, децентралізовані та гібридні мережі. Під час експериментальних досліджень була побудована мережа типу «зірка» з чотирма ОЕС. Основні її переваги полягали у простоті реалізації топології, незалежності кожного вузла від іншого вузла. Існує можливість замінити вузли, що вийшли з ладу, без втручання у роботу мережі, а завдяки розробленим командам та алгоритмам управління цей процес максимально спрощений.

Весь трафік обробляється на координаторі, яким виступає командний вузол. Розроблене програмне забезпечення для вирішення задач маршрутизації створює таблицю пристроїв на командному вузлі. В таблиці знаходиться інформація про порядковий номер кінцевого пристрою, його адреса, частотний канал. При використанні номера ОЕС з таблиці в пакети даних підставляється її адреса та канал, у якому вона працює.

Використання топології типу «зірка» мінімізує завантаження мережі службовим трафіком. Але централізація усього трафіку в одній точці також має свої недоліки. При виході з ладу центрального вузла маршрутизація буде порушена.

Кожна ОЕС в мережі має власний номер. Нумерація починається з командного вузла, він має номер 01. Далі за годинниковою стрілкою іде нумерація ОЕС, наприклад, з 02 до 06. Такий принцип нумерації спрощує у подальшому використанні взаємодії з кожною окремою станцією.

Сенсорні вузли мають дуже високу енергоефективність при виконанні певних вимог [6-8]. Платою за це є мала швидкість передачі даних. Незважаючи на цей недолік та застосовуючи сенсорні мережі можна покрити відстані від декількох метрів до десятків кілометрів. Для зв'язку між ОЕС пропонується використовувати технологію LoRa (Long Range). LoRa – це запатентована технологія малопотужного зв'язку на великі відстані, розроблена компанією Semtech і разом з протоколом LoRaWAN, що розроблений некомерційною організацією LoRa Alliance, формує малопотужні безпроводові мережі LPWAN (Low Power Wide Area Network). LoRaWAN – це протокол рівня управління доступом до середовища. Згідно з офіційною документацією відстань передачі даних варіюється від 1 до 15 кілометрів на відкритому просторі та до 5 км у щільній міській забудові. Комунікація між пристроями виконується у неліцензованому діапазоні частот (433/868/915 МГц), що дозволяє вільно використовувати технологію при побудові мереж інтернету речей (Internet of Things, IoT). Особливість технологій LoRa полягає у використанні методів

модуляції, що базуються на методах розширення спектру SMM і лінійній частотній модуляції CSS. Такий метод модуляції дозволяє приймати сигнали на 20 дБ нижче рівня шумів. Технологія LoRa охоплює фізичний рівень моделі OSI, а LoRaWAN – усі верхні урівні. Швидкість передачі даних варіюється від 0,3 кбіт/с до 50 кбіт/с залежно від коефіцієнту поширення.

Під час експериментальних досліджень БСМ на основі технології LoRa використовувались радіомодулі EBYTE E32 [9, 10].

I. Постановка задачі

Об'єкт дослідження – процес формування команд управління безпроводовою сенсорною мережею оптико-електронних станцій.

Предмет дослідження – метод управління безпроводовою сенсорною мережею оптико-електронних станцій із застосуванням технології LoRa.

Мета дослідження – розробка ефективних алгоритмів і команд для управління безпроводовою сенсорною мережею оптико-електронних станцій із застосуванням технології LoRa.

II. Налаштування безпроводової сенсорної мережі оптико-електронних станцій із застосуванням технології LORA і формування команд управління

Технічні характеристики модулю LoRa EBYTE E32-433T20DT

Для побудови безпроводової мережі з оптико-електронних станцій використовуються модулі LoRa EBYTE E32-433T20DT. Це модулі послідовного порту прийому / передачі (UART). За основу модулю зв'язку використовується РЧ-чип виробництва компанії SEMTECH моделі SX1278. На рис. 3 зображено модуль EBYTE E32-433T20DT [9].

Модуль EBYTE E32 має кілька режимів передачі в діапазоні частот 410 МГц – 441 МГц, за замовчуванням використовується частота 433 МГц. Модуль використовує модуляцію з розширенням спектру, що дозволяє помітно збільшити дальність зв'язку. Модуль E32 має потужність передачі 20 дБм і використовує кварцові генератори промислового класу для забезпечення стабільності. Завдяки вбудованим методам та алгоритмам шифрування і стиснення даних, процес перехоплення та дешифрування даних є неможливим.

Стиснення даних дозволяє скоротити час передачі та зменшити кількість втрачених даних через радіоелектронні завади. Вбудовані алгоритми та методи значно підвищують надійність та ефективність процесів передачі даних між модулями. В табл. 1 наведено основні параметри модулю зв'язку EBYTE E32-433T20DT.

Виходячи з технічних характеристик, можна виділити наступні ключові переваги модулю зв'язку EBYTE E32-433T20DT:

- ✓ гарантована дальність зв'язку: 3 км;
- ✓ використання неліцензійного діапазону частот;
- ✓ підтримка безпроводової передачі даних: до 19,2 кбіт/с;
- ✓ знижене споживання енергії;
- ✓ робочий температурний діапазон: від -45 до 85 °С;
- ✓ конфіденційна передача даних.

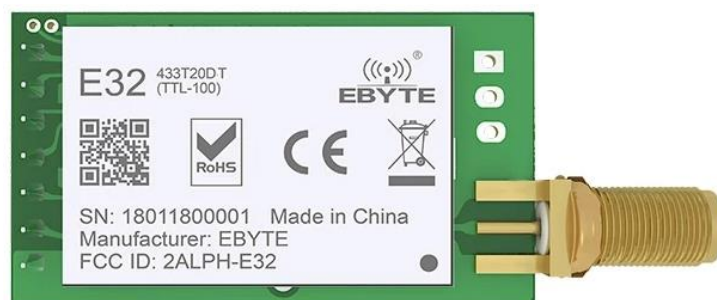


Рис. 3. Модуль EBYTE E32-433T20DT

Таблиця 1 – Основні параметри EBYTE E32-433T20DT

Назва параметру	Значення параметру	Коментар
Напруга живлення, В	2,3 – 5,5	Оптимальна напруга для роботи: 5В
Потужність передачі, дБм	17 – 20	-
Швидкість передачі даних, кбіт/с	0,3 – 19,2	Залежить від налаштувань контролера.
Дальність передачі даних, м	3000	Залежить від середовища, типу і характеристик антени
Довжина пакету, Байт	58	Максимальний розмір одного пакету
Буфер, Байт	512	-
Тип модуляції	LoRa (CSS + SMM)	-

Модуль зв'язку має 10 портів для підключення до мікроконтролера. Для підключення використовується UART (USART) інтерфейс. Як елементи живлення можуть використовуватись акумуляторні батареї, окремий блок живлення або живлення від мікроконтролера за умови, що вистачить потужності для передачі даних. Розміри та порти модуля EBYTE E32-433T20DT зображено на рис. 4 [10], цифрами від 1 до 10 позначено порти та отвори для комунікації з мікроконтролером.

Команди для передачі даних формуються на командному пристрої і далі через мікроконтролер передаються на модуль E32. Детальний опис портів наведено в табл. 2.

Управління модулем EBYTE E32-433T20DT виконується через зовнішній мікроконтролер. Загальну схему підключення модулю зв'язку до мікроконтролера

Arduino UNO через асинхронний/синхронний інтерфейс прийому/передачі (USART) наведено на рис. 5.

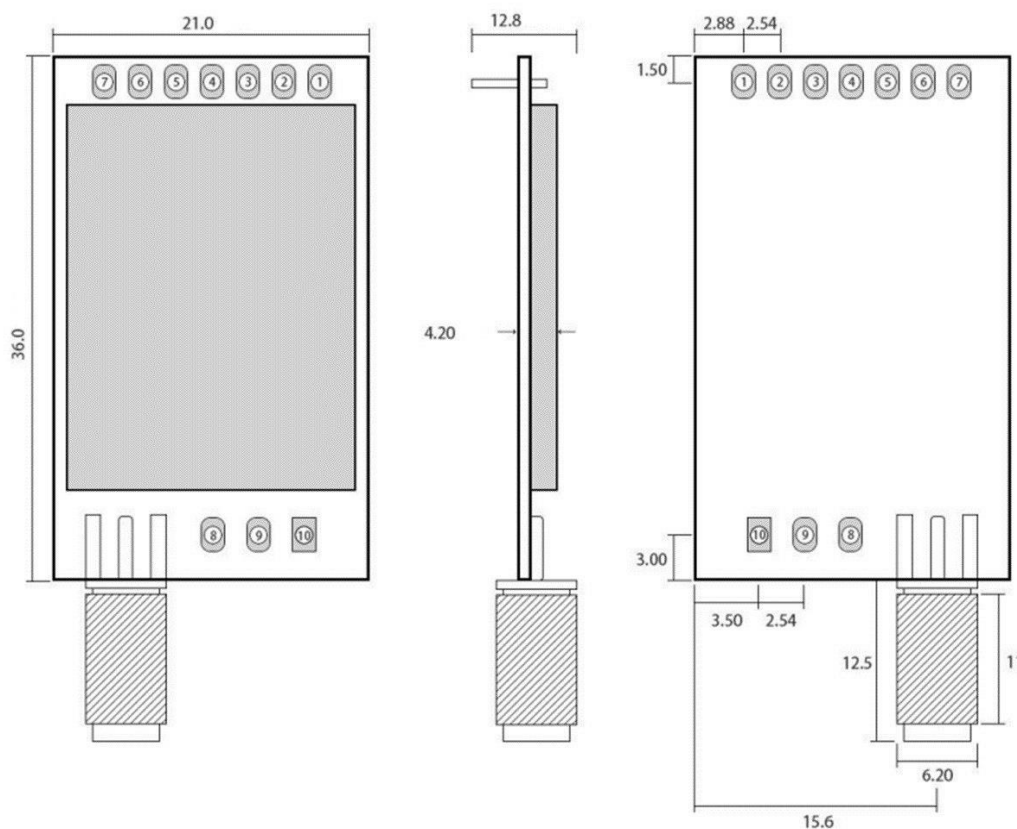


Рис. 4. Розміри та порти модулю EBYTE E32-433T20DT

Таблиця 2. Опис портів модулю EBYTE E32-433T20DT

Номер	Назва	Напрям	Опис
1	M0	Вхід (weak pull-up)	Працює у парі з портом M1. Потрібен для визначення режиму роботи (обирається один з чотирьох), не засинає, якщо не використовується.
2	M1	Вихід (weak pull-up)	Працює у парі з портом M0. Потрібен для визначення режиму роботи (обирається один з чотирьох), не засинає якщо не використовується, може бути заземлений
3	RXD	Вхід	TTL UART порт вводу, що підключається до зовнішніх приладів (ПК або мікроконтролер)
4	TXD	Вихід	TTL UART порт виводу, що підключається до зовнішніх приладів (ПК або мікроконтролер) на порт RXD (вхід).
5	AUX	Вихід	Потрібен для індикацій робочого стану модуля. Може виводити підключені пристрої з режиму сну. Може переходити в стан сну.
6	VCC	Вхід	Живлення (від 2,3 В до 5,5 В)
7	GND	Вхід	Заземлення
8-10	Фіксований отвір		

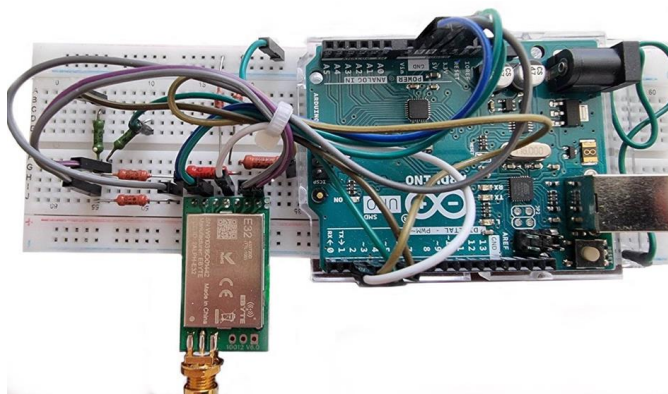


Рис. 5. Схема з'єднання мікроконтролера Arduino UNO та модуля зв'язку EBYTE E32-433T20DT

Режими роботи модулю LoRa EBYTE E32-433T20DT

Модулі EBYTE E32-433T20DT мають два режими передачі даних та чотири режими роботи. Зміна режимів можлива лише тоді, коли закінчиться черга на передачу даних. Режими роботи задаються зміною параметрів M0 та M1 [11].

Перший режим комунікації – це фіксована передача даних. У такому режимі задаються адреса та канал призначення. Значення для управління передаються у шістнадцятковому форматі. Наприклад, якщо у пакеті зазначена адреса 0x0002 та канал 0x04, то пакет приймається тільки модулем з адресом 0002, що «слухає» 4-й канал; інші модулі цей пакет відкинуть. На рис. 6 зображено схему фіксованої передачі даних.

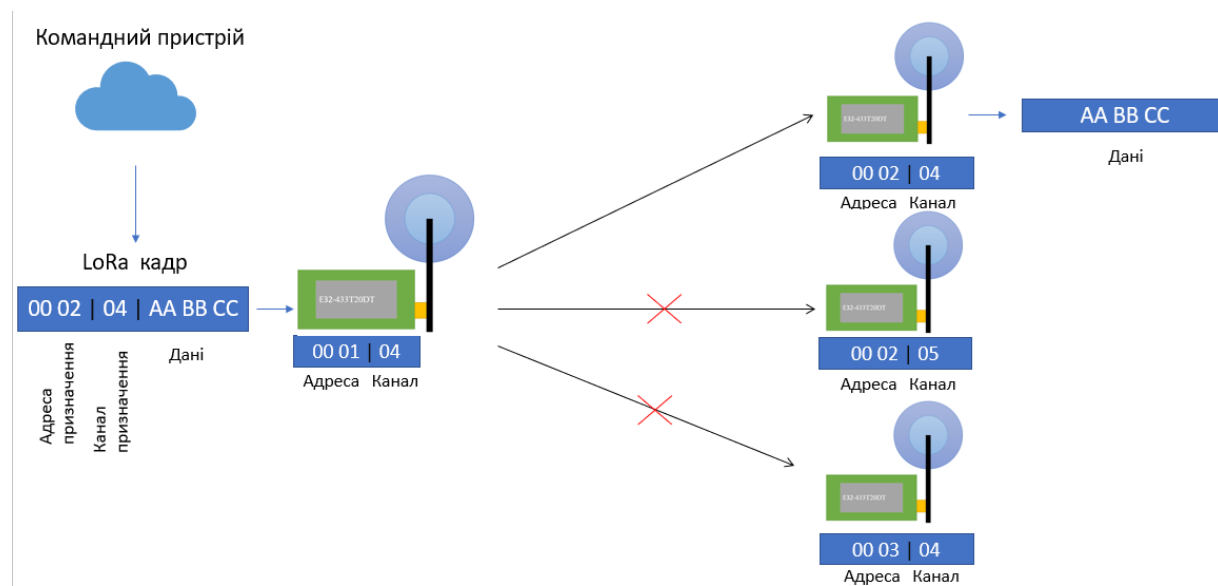


Рис. 6. Схема режиму фіксованої передачі даних

Другий режим комунікації – це ширококомовна розсилка. У пакеті задається адреса у вигляді 0x00 0x00 або ж 0xFF 0xFF і канал, наприклад, 0x04. Усі модулі, що

«слухають» 4 канал, приймуть цей пакет. Слід зауважити, що модуль-відправник не повинен «слухати» саме той канал, на який він розсилає пакети. На рис. 7 зображено схему ширококомовної розсилки даних.

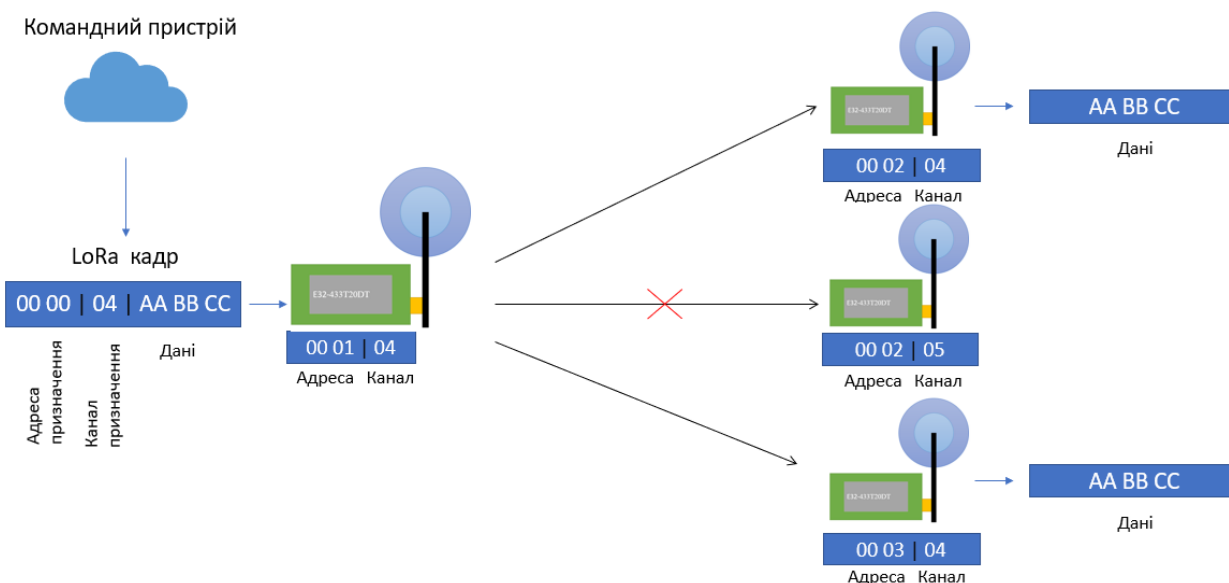


Рис. 7. Схема ширококомовної передачі даних у межах частотного каналу

Модулі зв'язку мають чотири режими роботи. Ці режими задаються параметрами M1 та M0. Режими рахуються з 0 до 3.

Нормальний режим роботи має номер 0, тобто значення M0 та M1 дорівнюють нулю. В цьому режимі модуль приймає сигнали від UART або безпроводового інтерфейсу. Максимальний розмір пакета 58 Байт (табл. 2). Як тільки з контролера надходять дані та пакет набуває свого максимального розміру, одразу починається процес передачі даних. Користувач може продовжувати вводити дані – вони будуть передаватись послідовно.

Режим пробудження має номер 1. У цьому режимі модуль продовжує приймати сигнали від UART або безпроводового інтерфейсу. Параметр M0 приймає значення 1, M1 має значення 0. Відмінність від нормального режиму полягає у тому, що автоматично до кожного пакету додається преамбула, яка дозволяє пробудити модуль-отримувач з режиму збереження енергії (режим 2).

Режим збереження енергії має номер 2. Для цього режиму значення M1 дорівнюється 1, M0 має значення 0. У цьому режимі UART закривається на прийом, а безпроводовий інтерфейс очікує на пакет зі спеціальною преамбулою від іншого модуля, що працює у режимі 1. Як тільки отримується сигнал на пробудження, UART одразу відкривається на прийом або передачу.

Режим сну має номер 3. Для нього є характерним, що значення M1 та M0 дорівнюють 1. Цей режим потрібен для корегування внутрішніх налаштувань

модулів E32. Якщо модулі зв'язку підключені до мікроконтролера або ПК крізь порт serial, тільки тоді є можливість для зміни внутрішніх параметрів.

Структура кадру LoRa

Кадр технології LoRa може бути двох типів – явний та неявний. Явний кадр містить у собі короткий заголовок, інформацію про CR (Coding rate) – код виправлення помилок (корисний потік даних) та CRC (Cyclic redundancy check) – тип циклічної перевірки збитковості у коді. Основні елементи кадру LoRa – це преамбула, опційний заголовок, корисне навантаження. На рис. 8 наведено структуру кадру LoRa.

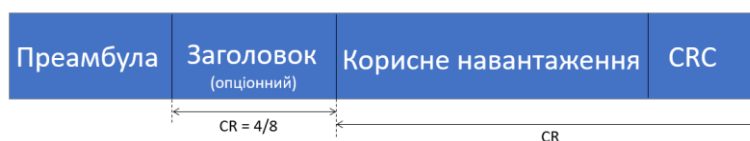


Рис. 8. Структура кадру LoRa

Преамбула потрібна для синхронізації приймача з потоком даних. За замовчуванням приймач налаштовано на послідовність довжиною 12 символів. Це значення програмно змінюється, але змінювати його слід і на стороні передачі, і на стороні прийому.

Явний тип кадру встановлюється за замовчуванням, тоді заголовок містить інформацію про довжину корисного навантаження в байтах, інформацію про CR і наявність додаткового 16-ти бітового CRC для корисного навантаження. Такий заголовок має максимальний розмір корегування помилок 4/8 і власний CRC. Заголовки з невірним CRC відкидаються. Неявний тип кадру використовується у випадках, коли інформація про корисне навантаження, CR та CRC відома заздалегідь. У цьому разі можливе скорочування часу передачі даних із застосуванням неявного заголовка. Для цього потрібно налаштувати у ручному режимі код виправлення помилок та параметри CRC на стороні прийому та передачі.

Корисне навантаження має змінну довжину. Це поле містить дані, які закодовані із використанням певного CR та додаткового CRC. Інформація про тип кодів передається у заголовку або програмується окремо на пристроях.

Додавання або видалення оптико-електронних станцій із мережі

Для вирішення задач дослідження були розроблені алгоритми та програмне забезпечення для додавання ОЕС в одну мережу або їх видалення. Управління мережею виконується із центрального вузла, який має певну програму з розробленим списком команд для налаштувань мережі. Команди відсилаються у вигляді текстових повідомлень з командного вузла на кінцеві пристрої, отримуються модулем EBYTE E32, потім розшифровуються мікроконтролером і виконуються.

Перед початком взаємодії з кожною окремою станцією її треба додати до списку пристроїв. Ініціалізація розпочинається з додавання до списку приладів командного вузла. На рис. 9 зображено схему підключення персонального комп'ютера до мікроконтролера Arduino UNO та модулю зв'язку EBYTE E32 [11].

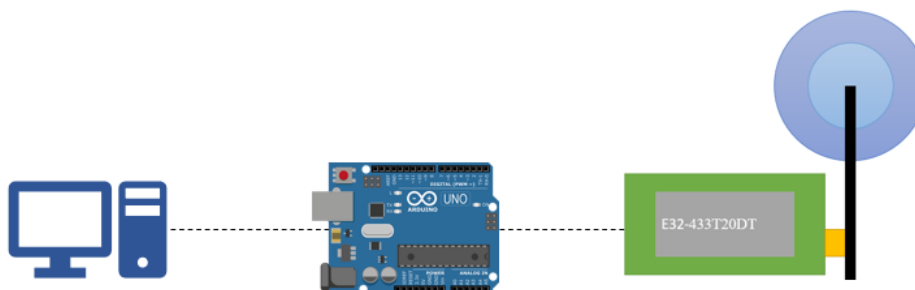


Рис. 9. Схема підключення персонального комп'ютера до мікроконтролера Arduino UNO

Для ініціалізації з персонального комп'ютера, підключеного до мікроконтролера (рис. 9), виконується команда `addStation(АДРЕСА; КАНАЛ; НомерОЕС)`, де АДРЕСА – це адреса станції, КАНАЛ – канал зв'язку, «Номер ОЕС» – власний номер пристрою. Щоб видалити пристрій із таблиці з адресами, необхідно виконати команду `deleteStation(НомерОЕС)`, де «Номер ОЕС» – це номер пристрою в мережі. Також є можливість для перевірки інформації про наявні пристрої в таблиці. Щоб отримати список станцій, потрібно виконати команду `listStation`. Результатом виконання цієї команди є список із адресою, каналом зв'язку та номером станції.

Отримання налаштувань модулів EBYTE E32

Для отримання налаштувань з модулів EBYTE E32 необхідно знати номер станції. Наприклад, до власної таблиці пристроїв буде додана станція з номером 02, яка матиме адресу 00 02 та використовуватиме 17 канал. Для цього необхідно виконати команду `addStation(0002;17;02)`. Якщо станція вже додана до таблиці, то одразу можна виконувати команду `getInfo(НомерОЕС)`, де «Номер ОЕС» – номер станції. Ця команда фактично містить в собі дві підкоманди для модуля зв'язку. Після отримання команди `getInfo` на ОЕС мікроконтролер переводить модуль зв'язку у режим 3 (режим сну), тобто прирівнює значення M0 та M1 до 1. Після цього виконує запит `<0xC3> <0xC3> <0xC3>`, і отриманий результат у виді `<C3> <32> <AA> <BB>` записується до змінної `versionInfo`, де C3 – заголовок відповіді на запит про версію, значення 32 показує, що використовується модуль E32 з частотою 433 МГц, AA буде замінено на версію модуля, а BB – на додаткові параметри модуля.

Далі виконується другий запит для отримання робочої конфігурації. Цей запит має вигляд `<0xC1> <0xC1> <0xC1>`. Результат виконання цього запиту записується до масиву `paramsInfo`. Результат запиту про параметри буде мати вигляд: `<C2> <00> <02> <1A> <17> <44>`, де C2 – заголовок відповіді на запит, який позначає, що

параметри не зберігаються при вимиканні пристрою, 00 02 – адреса модулю, 1A – це параметри комунікації у шістнадцятковому форматі. При перетворенні в двійкову систему матиме вид 00011010, біти нумеруються з 7 до 0. Сьомий та шостий біти відповідають за режим роботи UART; п'ятий, четвертий та третій біти відповідають за швидкість UART, біти з другого по нульовий відповідають за швидкість передачі даних по радіолінії. Цифрою 17 позначено номер каналу зв'язку, а 44 – це додаткові параметри. Далі мікроконтролер змінює параметри M1 та M0 на 0, тобто переводить модуль у нормальний режим роботи і починає передачу даних на пристрій, від якого отримав запит. Відповідь буде мати формат <0x00> <0x01> <0x17> versionInfo paramsInfo. Отримані контрольним вузлом дані декодуються та представляються користувачу у зручному для сприйняття вигляді. Таким чином, виділяються наступні частини пакету: заголовок пакету з версією модуля, номер частотного каналу, версія модуля зв'язку, заголовок пакета з параметрами, адреси (ADDH (High address) та ADDL (Low address)), режим роботи UART, швидкість передачі даних крізь UART, швидкість передачі даних по безпроводовому каналу зв'язку, інформація про канал, режим трансляції, параметри адаптації, затримки для пробудження, підтримка виявлення та виправлення помилок, потужність передачі даних.

Налаштування робочих параметрів модулів EBYTE E32

Для налаштування робочих параметрів модулів EBYTE E32 були розроблені алгоритми та команди, що дають змогу змінити основні параметри модулів зв'язку на кінцевих пристроях. Розроблені команди з префіксом set використовуються для зміни режимів роботи, адресації, швидкостей взаємодії з іншими елементами мережі.

Щоб налаштувати робочі параметри модулів зв'язку, потрібно вказувати в команді setParams ті параметри, які потрібно змінити. Основний синтаксис команди setParams(НомерОЕС;ПАРАМЕТР=ЗНАЧЕННЯ). Командний пакет є 6-ти байтовою послідовністю у шістнадцятковому форматі. Для зміни параметра користувача потрібно ввести назву параметра та значення для нього, після чого формується командний пакет і передається до кінцевого пристрою. Структура командного пакета наведена на рис. 10. Програма сама конвертує усі отримані дані від користувача для зміни параметрів. Такий принцип взаємодії суттєво полегшує користувачу процеси управління станціями, мережею та модулями зв'язку [12].

Командний пакет містить 6 основних байтів. Кожен байт несе корисну інформацію щодо окремого параметра (табл. 3).

Пакет з інформацією про налаштування починається з нульового байту – заголовка. Він містить інформацію про те, що відбудеться зміна параметрів. Значення записуються до змінної SAVE. Далі йдуть два байти, що позначають адресу модуля. Параметри ADDH (англ. High address) – верхня адреса, і ADDL (англ. Low address) – нижня адреса. На двох модулях, що працюють на одному частотному каналі, не можуть співпадати параметри адреси. Значення записуються до відповідних змінних ADDH та ADDL. Слідом іде 3-й байт, що налаштовує швидкість

інтерфейсів зв'язку, проводового та безпроводового. Четвертий байт налаштовує канал роботи модуля, значення записуються до змінної CHANNEL. Наступний 5-й байт містить інформацію про додаткові опції, наприклад, про механізм корегування помилок. У табл. 3 описано основні значення параметрів. Користувач може обирати та вводити значення для параметрів команди зі стовбця «Значення змінних для командного пакету». Розроблене програмне забезпечення конвертує усі введені значення на мову, зрозумілу мікроконтролеру.

Таблиця 3. Значення байтів командного пакету

Номер байту	Параметр	Значення у 16-тиричний системі	Значення у двійковій системі	Значення змінних для командного пакету	Коментар
0	Заголовок	0xC0	11000000	true	Збереження налаштувань після вимкнення
		0xC2	11000010	false	Вимкнення без збереження налаштувань
1	ADDH	00 – FF	00000000 – 11111111	0-255	Верхня адреса
2	ADDL	00 – FF	00000000 – 11111111	0-255	Нижня адреса
3	Швидкість	Значення кожного біта конфігурується окремо, докладніше в табл. 4 – 6			
4	Канал	00 – FF	00000000 – 11111111	0-255	Номер каналу
5	Опції	Значення кожного біта конфігурується окремо, докладніше в табл. 7 – 11			

Конфігурація 3-го байту командного пакету, а саме швидкостей передачі даних, виконується шляхом вибору окремих швидкісних параметрів з тих, що підтримуються РЧ-чіпом SX1278 модулів зв'язку. Кожен біт з цього байту відповідає за певну швидкісну характеристику пристрою зв'язку. Бітова структура байту, що відповідає за швидкості передачі даних, наведено на рис. 10.

Побітове налаштування режиму роботи порту UART наведено у табл. 4. Користувач для зміни параметрів має записати значення до змінної UARTmode під час виконання команди setParams. Тобто оператор мережі для вибору режиму роботи UART з перевіркою на парність буде вказувати параметр у такому виді: UARTmode=8E1.

Варіанти побітового налаштування швидкості порту UART наведено в табл. 5. Швидкість порту не впливає на швидкість передачі даних безпроводовим каналом.

За замочуванням швидкість передачі порту UART дорівнює 9600 біт/с, це значення оператор мережі має записувати до змінної UARTrate.

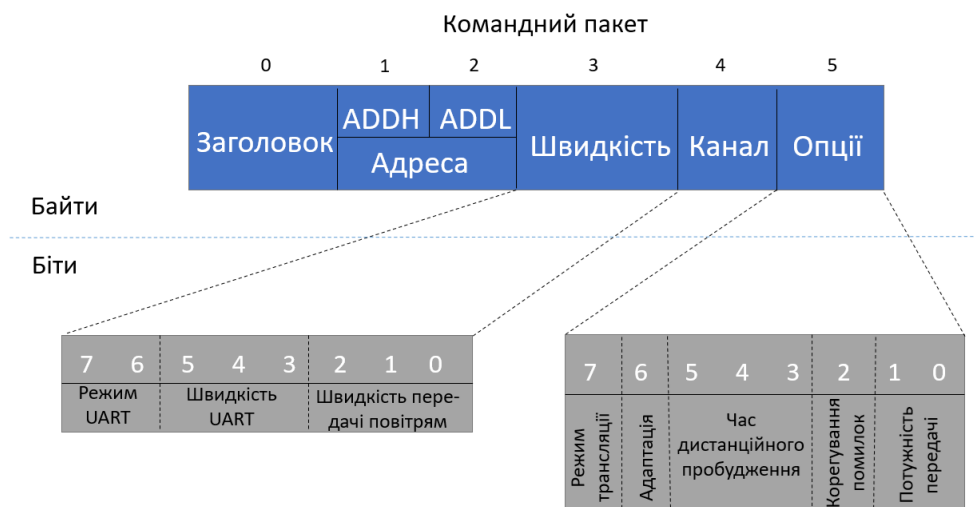


Рис. 10. Структура командного пакету у бітах та байтах

Таблиця 4. Конфігурація роботи порту UART

Біти		Режими роботи
7-й біт	6-й біт	
0	0	Порт UART та радіоканал активні, передача даних можлива
1	0	Порт UART та радіоканал активні, перед передачею даних необхідно передати команду для пробудження модуля-приймача
0	1	Порт UART закритий, передача даних не можлива
1	1	Режим очікування з можливістю приймати команди

Таблиця 5. Конфігурація швидкості порту UART

Біти			Швидкість порту UART
5-й біт	4-й біт	3-й біт	
0	0	0	1200 біт/с
0	0	1	2400 біт/с
0	1	0	4800 біт/с
0	1	1	9600 біт/с
1	0	0	19200 біт/с
1	0	1	38400 біт/с
1	1	0	57600 біт/с
1	1	1	115200 біт/с

Наступні три біти відповідають за швидкість передачі даних радіоканалом. За замовчуванням ця швидкість дорівнює 2,4 кбіт/с. Цей параметр, як і канал, повинен бути встановлений однаковим на усіх пристроях, що беруть участь у обміні даними. Побітові варіанти конфігурації швидкості передачі даних безпроводовою мережею

наведено в табл. 6. Значення режиму передачі даних у БСМ оператор має записати до змінної AIRrate.

У 5-му байті командного пакету зберігається інформація про додаткові параметри, а саме про режим передачі даних, режим роботи pull-up резистора, час пробудження, режим корегування помилок, потужність передачі даних. Побітова структура 5-го байту, що відповідає за додаткові опції передачі даних, наведено на рис. 10.

У табл. 7 наведено параметри для конфігурування 7-го біту, тобто режиму передачі даних. У фіксованому режимі передачі даних, перші три байти пакета даних будуть використовуватись як елементи адресації. Перший байт – верхня адреса, другий байт – нижня адреса, третій байт – канал передачі.

Таблиця 6. Конфігурація швидкості передачі даних радіоканалом

Біти			Швидкість передачі даних радіоканалом
2-й біт	1-й біт	0-й біт	
0	0	0	0,3 кбіт/с
0	0	1	1,2 кбіт/с
0	1	0	2,4 кбіт/с
0	1	1	4,8 кбіт/с
1	0	0	9,6 кбіт/с
1	0	1	19,2 кбіт/с
1	1	0	
1	1	1	

Таблиця 7. Побітове налаштування режиму передачі даних

7-й біт	Режим роботи
0	Широкомовний режим
1	Фіксований режим

У ширококомовному режимі передачі перші два байти будуть мати вид 0x00 або 0xFF, ключовим елементом адресації буде виступати значення третього байту, тобто каналу передачі. Значення оператор мережі має записувати до змінної AIRmode. У топології мережі ОЕС за замовчуванням використовується фіксований режим. Але для швидкої зміни швидкості передачі даних, наприклад, можна застосувати ширококомовний режим.

Наступний 6-й біт відповідає за режим роботи pull-up резистора. Він може знаходитись у вимкненому або увімкненому стані. В табл. 8 наведено відповідні параметри для конфігурування.

Таблиця 8. Побітове налаштування роботи pull-up резистора

6-й біт	Режим роботи pull-up резистора
0	Використовується pull-up резистор
1	Не використовується pull-up резистор

Далі три наступних біта налаштовують час пробудження модуля з 2-го режиму роботи. За замочуванням час пробудження – 250 мс. Щоб пробудити модуль з 2-го режиму, передача даних має здійснюватися лише у 1-му режимі. Час пробудження записуються до змінної WUP.

Другий біт відповідає за підтримку FEC (Forward error correction) – прямої корекції помилок. За замовчуванням FEC увімкнено. При відключенні FEC зростає швидкість передачі даних, але зменшується відстань передачі. У модулів, які беруть участь у передачі даних, налаштування прямої корекції помилок повинні бути однакові.

Таблиця 9. Побітове налаштування часу пробудження

Біти			Час пробудження
5-й біт	4-й біт	3-й біт	
0	0	0	250 мс
0	0	1	500 мс
0	1	0	750 мс
0	1	1	1000 мс
1	0	0	1250 мс
1	0	1	1500 мс
1	1	0	1750 мс
1	1	1	2000 мс

Таблиця 10. Побітове налаштування попереджувальної корекції помилок

2-й біт	Режим роботи
0	Пряма корекція помилок вимкнена
1	Пряма корекція помилок ввімкнена

Останні 2 біти відповідають за налаштування потужності передачі. За замовчуванням потужність становить 20 дБм. Значення вихідної потужності передавача записуються до змінної AIRpower.

Таблиця 11. Конфігурація потужності передачі даних

Біти		Рівень потужності
1-й біт	0-й біт	
0	0	20 дБм
1	0	17 дБм
0	1	14 дБм
1	1	10 дБм

Таким чином, команда для програмування другої станції ОЕС зі збереженням параметрів, зміною адреса на 00 04, каналу на 14, режимом роботи UART 8E1, швидкістю UART 9600 біт/с, швидкістю передачі даних безпроводовим каналом 4,8 кбіт/с, з роботою у фіксованому режимі, з вимкненим FEC, потужністю передачі

17 дБм буде мати вид: setParams(02; SAVE=true; ADDH=00; ADDL=04; CHANNEL=14; UARTmode=8E1; UARTrate=9600; AIRrate=4,8; AIRmode=fixed; ADAPTATION=true; WUP=250; FEC=false; AIRpower=17). Вже закодований командний пакет, але без системних даних, буде мати вигляд: <0xC0> <0x00> <0x04> <0x9B> <0x14> <0xC1>. На виході з командного пристрою перед командним пакетом додається заголовок, байт , який вказує мікроконтролеру, що треба перейти в режим 3 і змінити отримані параметри. До кінця командного пакету додається байт <C8>, який дає команду перейти до нормального режиму і повернути значення ОК після налаштувань. Додавання системних байтів є невід'ємною частиною розробленого програмного забезпечення. Під час передачі пакет буде мати вигляд: <0x00> <0x02> <0x17> <0xB> <0xC0> <0x00> <0x04> <0x9B> <0x14> <0xC1> <0xC8>.

Щоб налаштувати усі значення за замовчуванням, можна викликати setParams(02;DEFAULT), де 02 – номер станції, DEFAULT – команда для налаштувань усіх параметрів за замовчуванням.

Для додавання даних про командний вузол на кінцевих пристроях використовується команда setComand(НомерОЕС). Ця команда збирає інформацію про поточні налаштування командного пристрою та відсилає їх на ОЕС, яка зазначена в параметрі команди. Без цього налаштування кінцеві пристрої не зможуть комунікувати з командним вузлом. Змінити налаштування та перевірити їх на командному пристрої можна за допомогою команд setParams або getInfo, але номер станції обов'язково треба вказувати 01. Тому командний пристрій першим додається до таблиці пристроїв. Перевірка з'єднання виконується шляхом виконання команди ring(НомерОЕС). З командного пристрою відсилається байт <0xC8> і очікується відповідь. Якщо відповідь отримано, то зв'язок встановлено.

Розроблені команди суттєво полегшують взаємодію користувача з мережею станції ОЕС. Використання команд управління позбавляє оператора мережі від постійного запам'ятовування значень кожного біта з байтів командного пакету. Використовуючи дану розробку суттєво спрощуються процеси масштабування мережі, бо є можливість без затримок змінити ключові параметри елементів зв'язку, наприклад швидко змінити канал зв'язку чи швидкість передачі даних.

III. Ініціалізація мережі ОЕС

Основні процеси, що пояснюють роботу мережі, зручно показати за допомогою алгоритмів. Алгоритм ініціалізації мережі наведено на рис. 11. Цей алгоритм описує основні процеси ініціалізації мережі. Для управління мережею ОЕС використовується розроблене спеціальне програмне забезпечення. Набору закладених команд вистачає, щоб покрити основні вимоги для ініціалізації мережі. Як можна побачити з рис. 11, ініціалізація розпочинається з ввімкнення живлення усієї мережі. Після цього відбувається ініціалізація пристроїв, по завершенню якої пристрої готові виконувати базові команди для налаштування. Оскільки командний вузол є контролером усієї мережі, першим налаштовується саме він. Відбувається

перевірка робочих параметрів шляхом виконання команди `getInfo(01)`. Якщо перевірку пройдено, то можна приступити до перевірки таблиці доданих пристроїв.



Рис. 11. Алгоритм ініціалізації мережі

У випадку, якщо конфігурація командного вузла некоректна, її треба змінити шляхом виконання команди `setParams(01;ПАРАМЕТРИ)`. Перевірка пристроїв проводиться шляхом виконання команди `listStation`. Якщо пристрої відсутні, їх потрібно додати.

Пристрій додається командою `addStation(ПАРАМЕТРИ)`, далі на пристрій надсилається інформація про адресу та канал командного вузла `setComand(НомерОЕС)`, далі здійснюється перевірка зв'язку. У разі успішного виконання перевірки мережа та станції ОЕС готові до роботи. Якщо зв'язок не встановлено, потрібно видалити пристрій з таблиці та заново його додати шляхом відтворення попередніх кроків.

IV. Алгоритми додавання та видалення пристроїв

Додавання та видалення пристроїв з відповідної таблиці є однією з важливих операцій у межах взаємодії з модулями зв'язку. На рис. 12 зображено алгоритм додавання пристроїв до таблиці пристроїв, а на рис. 13 – алгоритм видалення пристроїв з цієї таблиці.

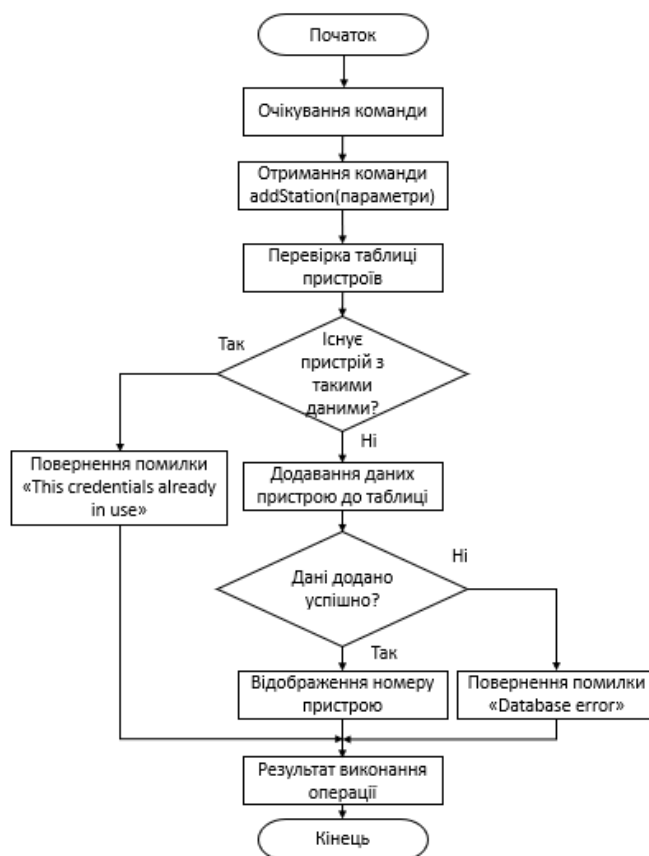


Рис. 12. Алгоритми додавання нового пристрою до таблиці пристроїв

Розроблені алгоритми мають схожу структуру. Суттєва відмінність полягає в операціях виконання команд і повідомленнях про помилки. Процес взаємодії розпочинається з очікування команди на мікроконтролері, далі надаються команди

на додавання або видалення пристрою з таблиці пристроїв. Після отримання команди відбувається перевірка на наявність пристроїв з таким самим номером та адресою у межах частотного каналу. У випадку додавання розглядається наявність схожих за параметрами пристроїв для збереження унікальності кожного модуля зв'язку. В процесі видалення виконується перевірка на наявність пристрою з потрібним номером. Далі, якщо перевірку пройдено, відбувається виконання самої операції. Після видалення пристрою з таблиці відбувається перевірка успішності виконання операції. Якщо усе виконано успішно, то користувачу в термінал повертається номер пристрою, над яким виконувалась операція.

У разі невиконання перевірок, розроблений алгоритм поверне помилки. У випадку з додаванням пристрою, як зображено на рис. 12, може повернутись помилка «This credentials already in use», яка вказує, що пристрій з такими даними вже існує в таблиці пристроїв. Повідомлення «Database error» вказує про помилку в операції з базою даних. У випадку з операціями видалення (рис. 13) можуть виникнути дві помилки. Перша – про відсутність пристрою у таблиці «No device», друга – про помилку в таблиці «Database error».

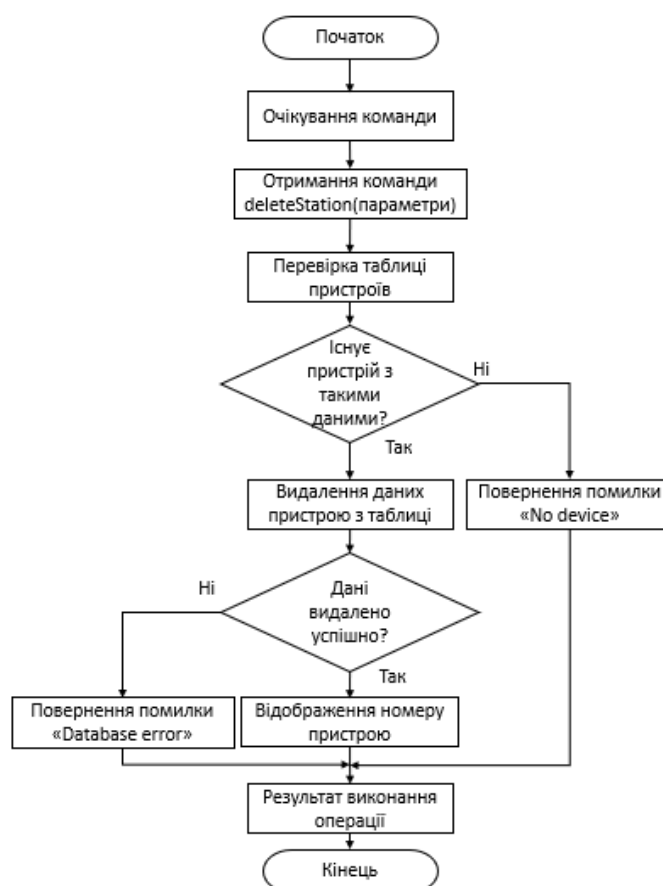


Рис. 13. Алгоритми видалення пристрою з таблиці пристроїв

V. Алгоритми виконання команд командного пристрою на кінцевих пристроях

Під час розробки усіх функціональних команд з префіксом set, а саме команд setComand, setParams, та команд getInfo, ping було поставлено завдання розробити систему надійною та доступною для розуміння іншими розробниками. Тому під час проектування було закладено загальний алгоритм виконання команд. Цей алгоритм складається з двох частин: частини командного пристрою та частини кінцевого пристрою. Алгоритм, наведений на рис. 14, описує основні процеси, ініціатором яких є командний пристрій.

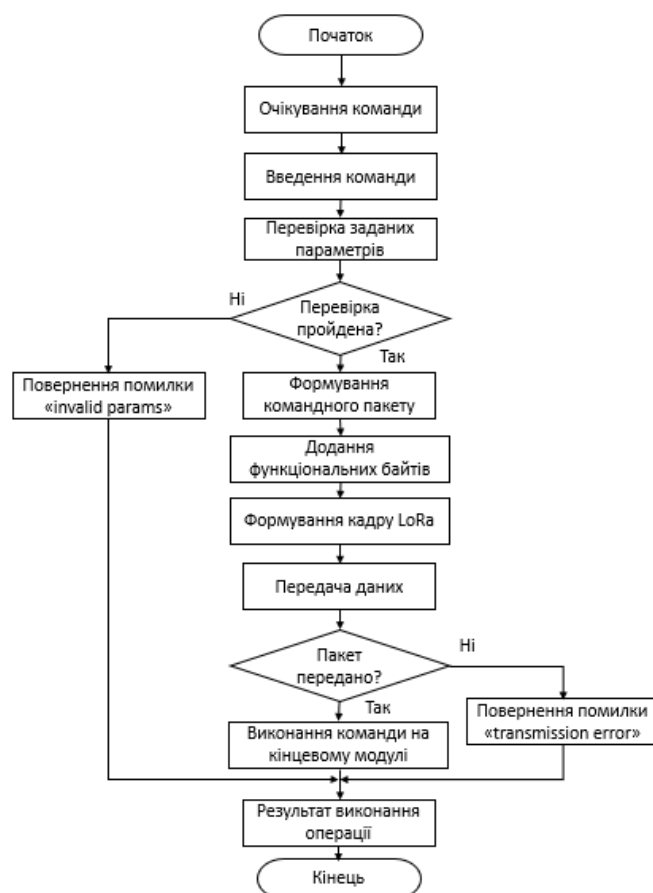


Рис. 14. Алгоритм формування та передачі команди з командного пристрою

Процес формування команд розпочинається з очікування початку введення команди користувачем. Як тільки користувач виконає команду, наприклад, setParams(02; SAVE=true; ADDH=00; ADDL=04; CHANNEL=14; UARTmode=8E1; UARTrate=9600; AIRrate=4,8; AIRmode=fixed; ADAPTATION=true; WUP=250; FEC=false; AIRpower=17), буде виконана перевірка кожного параметру. За результатом перевірки буде встановлено існування таких параметрів і наявність їх значень серед можливих варіантів. У разі успішного виконання перевірок розпочинається формування командного пакета, тобто конвертація параметрів у зрозумілий для

модулів EBYTE E32 вигляд. Наступним кроком додаються функціональні байти, як наприклад байт перед командним пакетом, що дає інформацію про зміну режиму роботи модулю.

Після додавання додаткової інформації починається формування кадру LoRa, до командного пакету додається інформація про адресу отримувача пакета та номер частотного каналу. Після завершення додавання інформації розпочинається процес передачі даних. Далі відбувається виконання алгоритму отримання та виконання команди кінцевим пристроєм, схему цього процесу наведено на рис. 15. При невиконанні команди або вичерпанні часу очікування до користувача повертається повідомлення про помилку.

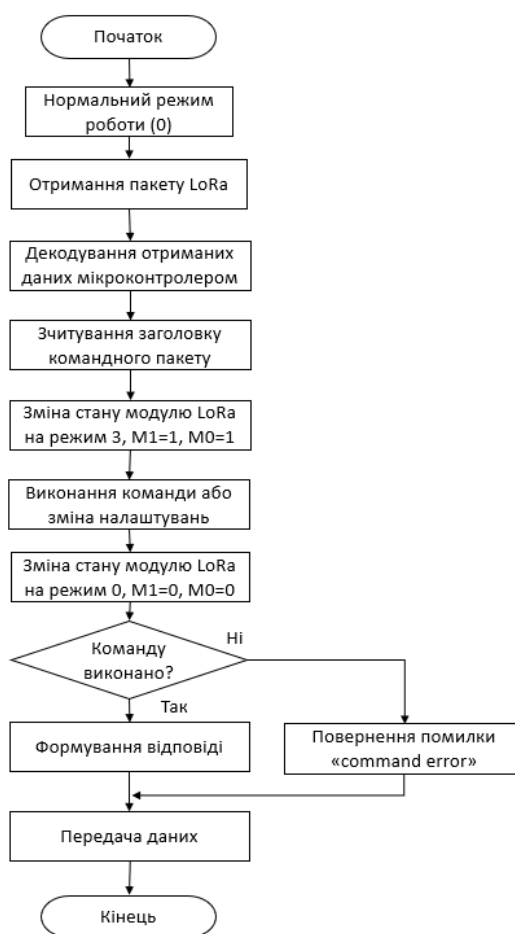


Рис. 15. Алгоритм виконання команди кінцевим пристроєм

Як можна побачити з рис. 15, модуль зв'язку працює у нормальному режимі та очікує дані від командного пристрою. Як тільки отримується будь-який пакет, він передається з модулю EBYTE E32 на мікроконтролер Arduino UNO та декодується, зчитується спочатку системна інформація, тобто додатково доданий байт , після декодування якого мікроконтролер змінює логічні параметри модулю зв'язку: M1=1 та M0=1, чим переводить модуль у режим 3. У цьому режимі модуль готовий виконувати команди.

На наступному кроці передається команда з мікроконтролера на модуль зв'язку і виконуються операції зі зміни налаштувань модулю EBYTE E32. При успішному виконанні формується відповідь ОК. Далі мікроконтролер змінює режим модуля зв'язку на нульовий, тобто нормальний режим прийому/передачі. Далі виконується передача результату виконання команди на командний пристрій.

VI. Приклад практичного застосування

Щоб налаштувати розроблену мережу для ефективної роботи у щільній міській забудові треба змінити параметри передачі даних. Мережа будується, наприклад, з чотирьох вузлів, поєднаних з ОЕС, три з яких розташовані по кутах трикутника, виділеного на рис. 16 жовтим кольором. Командний вузол, поєднаний з першою ОЕС, знаходиться рівно у центрі кола, яким рухається літак.

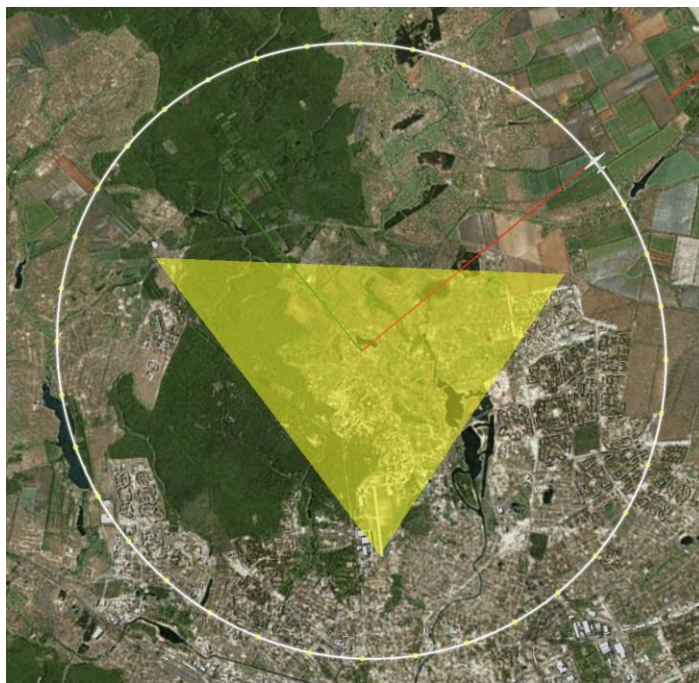


Рис. 16. Розташування вузлів мережі

На першому кроці згідно з алгоритмом (рис. 12) треба додати три ОЕС до мережі. Для цього потрібно виконати команди `addStation(02; 0002;17)`, `addStation(03; 0003;17)` та `addStation(04; 0004;17)`.

На другому кроці після додавання пристроїв до таблиці пристроїв треба поширити серед них інформацію про командний вузол, щоб вони могли встановити з ним зв'язок. Для цього виконуються команди `setComand(02)`, `setComand(03)` та `setComand(04)`.

В межах третього кроку для використання усіх переваг модулів LoRA необхідно внести зміни до налаштувань пристроїв. Під час функціонування пристроїв зв'язку у щільній міській забудові не стільки грає роль швидкість передачі даних, як завадозахищеність. Тому потрібно знизити швидкість передачі, ввімкнути FEC,

виставити максимальну потужність роботи. Спочатку це треба зробити для командного вузлу, а потім для кінцевих пристроїв. Завдяки розробленим командам, налаштування виконуються з використанням лише однієї команди, змінюючи лише номер станції: `setParams(01;SAVE=true; CHANNEL=11; AIRrate=1,2; FEC=true; AIRpower=20)`.

На четвертому кроці після зміни каналу роботи необхідно оновити таблицю з адресами станції. Для цього треба прибрати станції з таблиці пристроїв шляхом виконання команд `deleteStation(02)`, `deleteStation(03)` та `deleteStation(04)`, а потім додати знову, але з іншим частотним каналом, використовуючи команду `addStation`.

На п'ятому кроці, щоб переконатись у працездатності мережі, потрібно виконати команду `ping` з метою перевірки з'єднання зі станціями: з другою станцією `ping(02)`, з третьою `ping(03)` та четвертою `ping(04)`.

Висновки

Таким чином, вирішено задачу щодо розробки методу управління безпроводовою сенсорною мережею оптико-електронних станції із застосуванням технології LoRa. Особливістю запропонованого методу стало використання двох класів команд налаштування мережі. Перший клас команд відповідає за управління мережею, тобто за операції додавання (`addStation`) та видалення (`deleteStation`) пристроїв з власної таблиці пристроїв. Другий клас команд використовується для управління модулями зв'язку та мікроконтролерами. До цього класу відносяться команди `setParams`, `setComand`, `getInfo`, `ping`. Ці команди служать для прямої взаємодії з модулями зв'язку через мікроконтролер. Кожна команда має свою окрему функцію:

- `setParams` потрібна для зміни налаштувань модулів зв'язку,
- `setComand` використовується для поширення до кінцевих пристроїв інформації про контрольний вузол,
- `getInfo` збирає дані про поточну конфігурацію вузлів,
- `ping` дозволяє перевірити наявність з'єднання.

Обидва класи команд реалізують алгоритми управління мережею, які покроково описують процеси від ввімкнення живлення до переведення мережі у повністю працездатний стан. Перший клас команд реалізує алгоритм для додавання та видалення ОЕС з таблиці пристроїв, другий клас команд – алгоритм виконання та формування команд на командному вузлі та алгоритм отримання, декодування команди на кінцевому пристрої.

Список літератури

1. *Tevyashv, A., Shostko, I., Zemlyany, O.* (2021), "Network optoelectronic airspace monitoring system. International Scientific Journal", Transfer of Innovative Technologies, No. 4(1), P. 106–107. DOI: <https://doi.org/10.32347/tit2141.0308>

2. Шостко, И.С., Тевяшев, А.Д., Неофитный, М.В., Куля, Ю.Э., Колядин, А.В. (2019), "Методы позиционирования узлов беспроводной сенсорной сети", Проблемы телекомунікацій, No. 1(24), С. 68-89. DOI: <https://doi.org/10.30837/pt.2019.1.05>
3. Buratti, C., Conti, A., Dardari, D., Verdone, R. (2009), "An Overview on Wireless Sensor Networks Technology and Evolution", Sensors, No. 9, P. 6869-6896. DOI: <https://doi.org/10.3390/s90906869>
4. Шостко, И.С., Куля, Ю.Э. (2015), "Разработка модели маршрутизации беспроводной сенсорной сети с учётом устранения дисбаланса энергопотребления", Технологический аудит и резервы производства, No. 3/2(23), С. 94-98. DOI: <https://doi.org/10.15587/2312-8372.2015.44817>.
5. Олифер, В. Г., Олифер, Н. А. (2010), Компьютерные сети. Принципы, технологии, протоколы, 4-е изд., Санкт-Петербург : Питер, 943 с.
6. Shostko, I., Kulia, Y. (2015), "Applications of Multipath Routing for Energy Balancing in Sensor Networks Scientific and Technical Journal", RADIOELECTRONICS & INFORMATICS, No. 1, P. 12-16.
7. Шостко И.С., Куля Ю.Э. (2015), "Метод увеличения времени жизни автономной беспроводной сенсорной сети, за счет применения энергоэффективных режимов работы" Проблемы телекомунікацій, No. 1(16), С. 3–15 URL: https://pt.nure.ua/wp-content/uploads/2020/01/151_shostko_ttl.pdf
8. Chen, Y., Nasser, N. (2006), "Energy-balancing multipath routing protocol for wireless sensor networks", QShine '06 : proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks, New York, P. 1-4. DOI: <https://doi.org/10.1145/1185373.1185401>
9. SEMTECH Semtech Corporation, (2012), "SX1278 Wireless Module E32", available at: <https://pdf1.alldatasheet.com/datasheet-pdf/view/501039/SEMTECH/SX1278.html>. (last accessed 11.08.2022)
10. SEMTECH Semtech Corporation, (2012), "X1276/77/78 - 137-1050 MHz Ultra Low Power Long Range Transceiver", available at: <https://pdf1.alldatasheet.com/datasheet-pdf/view/501037/SEMTECH/SX1276.html>. (last accessed 11.08.2022)
11. Arduino.cc, (2022), "Arduino UNO documentation" available at: <https://docs.arduino.cc/static/d9af5fb18d52a1ea2cb848e8d0a325db/A000066-datasheet.pdf> (last accessed 10.08.2022)
12. SEMTECH Semtech Corporation, (2015), "LoRa Modulation Basics AN1200.22", available at: <https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf> (last accessed 11.08.2022)