КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ

Кафедра радіотехніки та радіоелектронних систем

До захисту допущено: «На правах рукопису»
Завідувач кафедри _____ Ігор АНІСІМОВ
« __ » червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

«РОЗРОБКА ТА АНАЛІЗ КАНАЛУ ЗВ'ЯЗКУ НА БАЗІ LoRa SX 1278»

Виконав:
студент 4-го курсу
денної форми навчання
спеціальності 172 - Телекомунікації та радіотехніка
ОП «Інформаційна безпека телекомунікаційних систем і мереж»
Малєєв Сергій Олександрович _____

Науковий керівник:
к.т.н., ас. Аль Шурайфі МУШТАК _____

Рецензент:
д.т.н., проф. Хлапонін Юрій Іванович _____

Засвідчую, що у цій бакалаврській роботі
немає запозичень з праць інших авторів без
відповідних посилань
Студент _____

Робота допущена до захисту в ЕК рішенням кафедри радіотехніки та радіоелектронних систем від «22» червня 2023 р., протокол № 21.

Завідувач кафедри радіотехніки та радіоелектронних систем,
доктор фіз.-мат. наук, професор
Анісімов Ігор Олексійович _____

Київ – 2023

ЗМІСТ

Анотація…….4

Вступ.....5

РОЗДІЛ 1. ТЕОРІЯ.....6

1.1 Технологія Low-Power Wide Area Network……..6

1.1.1 Мінімальна структура LPWAN повідомлення……...8

1.2 Протокол Long Range Wide Area Network……...9

1.2.1 Архітектура мережі LoRaWAN……..….9

1.2.2 Технічні характеристики LoRaWAN…….…11

1.2.3 Метод модуляції LoRaWAN……...12

1.2.4 Безпека мережі LoRaWAN…….13

1.2.5 Алгоритм контролю доступу LoRaWAN……..14

1.3 Протокол NB-IoT……..17

1.3.1 Архітектура мережі NB-IoT…….…..17

1.3.2 Технічні характеристики NB-IoT…….…19

1.3.3 Безпека мережі NB-IoT……...21

1.3.4 Алгоритм контролю доступу NB-IoT……....22

1.4 Протокол Sigfox……....23

1.4.1 Архітектура мережі Sigfox…….....24

1.4.2 Технічні характеристики Sigfox……....25

1.4.3 Безпека мережі Sigfox…….26

1.4.4 Алгоритм контролю доступу Sigfox…….27

1.5 Протокол LTE-M……..…29

1.5.1 Архітектура мережі LTE-M……...……30

1.5.2 Технічні характеристики LTE-M……
32
1.5.3 Безпека мережі LTE-
M……....33
1.5.4 Алгоритм контролю доступу LTE-M……...
…..34
РОЗДІЛ 2. КОМПОНЕНТНА БАЗА…….36
2.1 LoRa модуль
sx1278……..36
2.1.1 Характеристики SX1278 (Ra-02)……
36
2.1.2 Види модуляції SX1278 (Ra-02)……37
2.2 Мікроконтролер ESP 32……...……....37
2.2.1 Ключові характеристики ESP32……...….38
2.3 Датчик чадного газу MQ-7 модуль…….…40
2.3.1 Характеристики MQ-7……...…….41
2.4 Датчик температури та вологості DHT11…….….42
2.4.1 Характеристики……..….42

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СХЕМИ……..…..44
3.1 Блок схема……...……..44
3.2 Кодова реалізація……..…
46
3.2.1 Приймач……..…..…46
3.2.2 Код приймача……..…48
3.2.3 Передавач……56
3.2.4 Код передавача……56

3.3 Результат роботи каналу зв’язку ……..…
61
3.4 Інтерфейс в телеграм боті……62

ВИСНОВКИ ……..……..63

СПИСОК ВИКОРИСТАНИХ
ДЖЕРЕЛ.....65

Анотація

Ця робота включає в себе порівняльний аналіз різних протоколів LPWAN (Low-Power Wide Area Network) з метою їх використання в Інтернеті речей (IoT). Зокрема, в роботі порівнюються протоколи LoRaWAN, NB-IoT, Sigfox та LTE-M. Для забезпечення зв'язку на основі протоколу LoRaWAN розроблено канал зв'язку на базі радіочастотного модуля SX1278.

В роботі описані технічні характеристики кожного протоколу, їх переваги та недоліки, що дозволяє зробити висновок, який протокол краще підходить для конкретних застосувань. Крім того, описано процес розробки каналу зв'язку на базі LoRa SX1278, включаючи вибір компонентів, схемотехнічне проектування та розробку програмного забезпечення.

Основною метою роботи є розробка та дослідження каналу зв'язку на основі LoRa SX1278 з використанням оптимальних параметрів передачі даних. Результатом роботи буде розробка каналу зв'язку, який зможе передавати дані на великі відстані з використанням технології LoRa з високою ефективністю та мінімальними втратами даних, використовуючи візуалізацію в телеграмі. Для досягнення поставленої мети необхідно вирішити наступні завдання:

- Описати технологію LoRa та її основні характеристики;
- Проаналізувати параметри передачі даних LoRa SX1278 та обрати оптимальні налаштування;
- Реалізація канал зв'язку на основі LoRa SX1278 з використанням оптимальних параметрів передачі даних;
- Провести експериментальні дослідження розробленого каналу зв'язку з метою оцінки ефективності передачі даних та дальності зв'язку.

Ключові слова: LPWAN, IoT, LoRaWAN, NB-IoT, Sigfox, LTE-M, радіочастотний модуль SX1278, технічні характеристики, переваги, недоліки, розробка, зв'язок.

Вступ

Актуальність цієї теми зумовлена зростанням інтернету речей (IoT), що вимагає надійних і енергоефективних способів передачі даних на великі відстані. Технології LPWAN (Low Power Wide Area Network) відкривають широкі можливості для застосування IoT у найрізноманітніших галузях - від розумних будинків до сільського господарства та промислового виробництва.

Протоколи LPWAN, такі як LoRaWAN, NB-IoT, Sigfox і LTE-M, мають унікальні характеристики, які роблять їх придатними для різних застосувань IoT. Однак, щоб ефективно використовувати їх у конкретному контексті, важливо розуміти їхні відмінності та обмеження. Саме тому порівняльний аналіз цих протоколів має велику значимість.

Крім того, розробка каналу зв'язку на основі LoRa SX 1278 представляє інтерес у рамках дипломної роботи. Технологія LoRa дає змогу створювати енергоефективні та довговічні системи передавання даних, що робить її актуальною для досліджень і розробок у сфері IoT. У цьому контексті, створення робочого прототипу каналу зв'язку та оцінка його продуктивності може дати цінні відомості про можливості та обмеження цієї технології.

Таким чином, актуальність цієї теми зумовлена важливістю дослідження та розуміння різних технологій LPWAN для розвитку IoT, а також практичним значенням розробки та оцінки каналу зв'язку на базі LoRa.

Метою роботи є проведення порівняльного аналізу різних протоколів LPWAN (LoRaWAN, NB-IoT, Sigfox і LTE-M) для їхнього застосування в галузі IoT, а також проєктування та реалізація каналу зв'язку на основі LoRa SX 1278 з телеграм інтерфейсом. Це включатиме в себе дослідження характеристик,

переваг і недоліків кожного з цих протоколів, досліджування прототипу каналу зв'язку на основі LoRa SX 1278 і оцінку його продуктивності в реальних умовах.

Предметом роботи є методи і технології, пов'язані з реалізацією системи зв'язку на основі LoRa з використанням мікроконтролера ESP32 і подальшою візуалізацією прийнятих даних через телеграм-бот.

Об’єктом роботи є канал зв’язку на основі мікроконтролера ESP32, модуля LoRa SX 1278, формувачі інформації для передачі по каналу: датчики MQ-7 та DHT11, візуалізація інформації через телеграм бот.

РОЗДІЛ 1. ТЕОРІЯ

1.1 Технологія Low-Power Wide Area Network

Технологія LPWAN (Low-Power Wide Area Network) є групою бездротових мереж, розроблених для забезпечення низького енергоспоживання, дальньої дальності передавання даних і низької вартості під'єднання пристроїв в Інтернет речей (IoT).

LPWAN було створено для забезпечення зв'язності пристроїв із низьким енергоспоживанням і обмеженими ресурсами, як-от датчики, монітори, лічильники та інші пристрої, які використовують для збирання даних. Технологія забезпечує широкий радіус покриття, що дає змогу передавати дані на значні відстані.

Однією з найбільш відомих технологій LPWAN є LoRaWAN (Long Range Wide Area Network). Вона працює в ліцензованому або неліцензованому діапазоні частот і забезпечує дальність передавання даних до кількох кілометрів у міських умовах. LoRaWAN використовує технологію широкого спектра для передачі даних з низьким рівнем енергоспоживання.

Ще одна популярна технологія LPWAN - NB-IoT (Narrowband IoT). Вона заснована на стандартах мобільного зв'язку і надає низькошвидкісний канал передачі даних. NB-IoT забезпечує хорошу проникаючу здатність всередині будівель і має дальність до декількох кілометрів. Вона може працювати в наявних мережах стільникового зв'язку і потребує менших змін в інфраструктурі.

Sigfox - це глобальна мережа LPWAN, яка використовує спеціальний протокол зв'язку для передачі невеликих обсягів даних на великі відстані. Мережа Sigfox працює в ліцензованому діапазоні частот і забезпечує глобальне покриття в

більш ніж 70 країнах. Вона надає низьку швидкість передачі даних, але має низьке енергоспоживання, що робить її ідеальною для простих IoT-пристроїв, які вимагають тривалого терміну служби батареї.

LTE-M (Long Term Evolution for Machines) - це стандарт мобільного зв'язку, розроблений спеціально для підключення пристроїв IoT через мережі LTE. Він надає вищу швидкість передавання даних порівняно з іншими технологіями LPWAN, а також підтримує функції, необхідні для IoT, як-от низьке енергоспоживання і глибока проникаюча здатність усередині будівель. LTE-M працює в наявній інфраструктурі стільникових мереж, що робить його привабливим для масштабування наявних мереж і використання в різних застосуваннях IoT.

1.1.1 Мінімальна структура LPWAN повідомлення:

Преамбула (4 байти): Преамбула зазвичай використовується для вказівки початку повідомлення і допомагає приймачу синхронізуватися з сигналом передачі.

Адреса пристрою в мережі (4 байти): Це унікальний ідентифікатор пристрою в мережі, який допомагає визначити, кому призначено повідомлення.

Інформація користувача (0-12 байт): Це власне дані, які користувач хоче передати. Величина цих даних може варіюватися.

CRC (2 байти): CRC (Cyclic Redundancy Check) використовується для перевірки наявності помилок у повідомленні.

Іммітовставка (2 байти): Це елемент криптографічного захисту даних, призначений для забезпечення цілісності даних.

Динамічна вставка / лічильник (2 байти): Цей елемент може бути використаний для підрахунку кількості переданих повідомлень або для внесення додаткової динамічної інформації.

Службова інформація (0-4 байти): Це додаткова інформація, яка може містити елементи керування повідомленням або додаткові метадані.

Перешкодозахищене кодування x2: Це метод кодування, який подвоює розмір переданих даних, але забезпечує підвищений перешкодозахист. Це особливо корисно в умовах шуму або перешкод для забезпечення доставки даних [1].

1.2 Протокол Long Range Wide Area Network

LoRaWAN (Long Range Wide Area Network) - це протокол і технологія бездротового зв'язку, розроблені для зв'язку між пристроями IoT (Internet of Things) на великих відстанях і за низького енергоспоживання. LoRaWAN заснований на модуляції LoRa (Long Range), яку розробила компанія Semtech, яка забезпечує дальність передачі даних на відстанях до декількох кілометрів і більше у відкритих умовах [2].

1.2.1 Архітектура мережі LoRaWAN

LoRaWAN використовує унікальну архітектуру, відому як "зірка із зірок" або "зіркоподібна". У цій архітектурі вузли (end-devices) передають дані на шлюзи (gateways), які потім передають ці дані мережевому серверу (network server) мережевий сервер потім передає дані на додаток.

Давайте детальніше розглянемо кожен із цих компонентів:

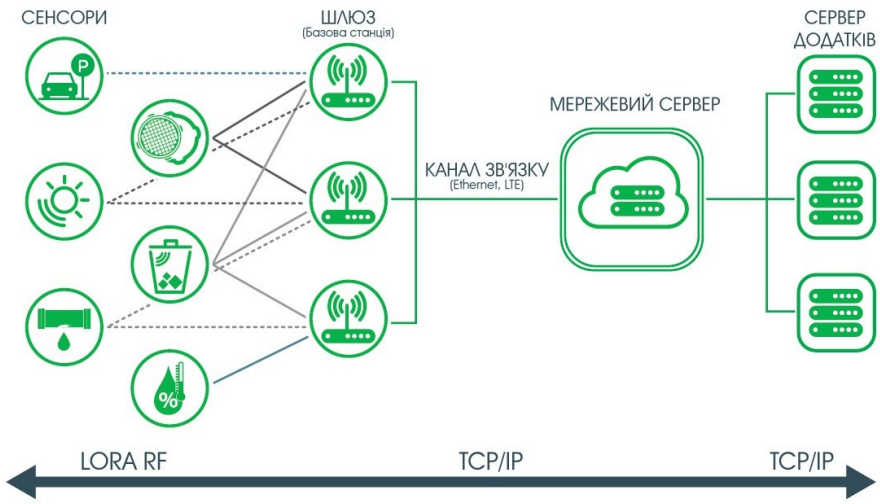

Рис. 1.1 Архітектура мережі LoRaWAN [3]

Вузли (End-Devices): Це пристрої, які збирають і передають дані. Вузли зазвичай мають низьке енергоспоживання і можуть передавати дані на великі відстані.

Шлюзи (Gateways): Шлюзи слухають дані, що передаються вузлами, і пересилають їх на мережевий сервер. Шлюзи можуть обслуговувати безліч вузлів одночасно, і вони зазвичай під'єднані до мережі з великою пропускною здатністю, такої як Ethernet або 4G/5G.

Мережевий сервер (Network Server): Мережевий сервер обробляє дані, отримані від шлюзів, і керує мережею. Він відповідає за маршрутизацію даних до додатків, керування частотою передавання та потужністю, а також забезпечення безпеки мережі.

Додаток (Application): Додатки отримують і обробляють дані від мережевого сервера. Вони можуть представляти дані в зручному для користувача форматі та виконувати різні функції, залежно від конкретної сфери застосування.

Важливо зазначити, що LoRaWAN також включає два рівні шифрування для забезпечення безпеки даних: один на рівні мережі та один на рівні застосунку. Це забезпечує, що тільки авторизовані пристрої можуть приєднуватися до мережі, а дані, що передаються через мережу, захищені від несанкціонованого доступу [4].

1.2.2 Технічні характеристики LoRaWAN

LoRaWAN (Long Range Wide Area Network) пропонує низку технічних характеристик, що роблять його ідеальним для застосування в інтернеті речей (IoT). Ось деякі з цих характеристик:

Дальність: LoRaWAN забезпечує довгу дальність зв'язку, яка може сягати до 15 км у сільській місцевості та до 5 км у міських умовах .

Енергоефективність: Пристрої LoRaWAN зазвичай мають низьке енергоспоживання, що дає змогу їм працювати на батарейках упродовж кількох років. Це досягається завдяки використанню модуляції LoRa, яка ефективно використовує енергію, і ефективного управління живленням.

Швидкість передачі даних: LoRaWAN забезпечує низьку швидкість передачі даних, яка може варіюватися від 0,3 кбіт/с до 50 кбіт/с залежно від налаштувань і умов зв'язку. Хоча це може бути обмеженням для деяких застосунків, це цілком достатньо для більшості застосунків IoT, які потребують передавання тільки невеликих обсягів даних.

Масштабованість: LoRaWAN підтримує велику кількість пристроїв, що робить його ідеальним для великих застосувань IoT. Один шлюз LoRaWAN може обслуговувати тисячі пристроїв.

Безпека: LoRaWAN містить вбудовані функції безпеки, як-от двостороннє шифрування на рівні мережі та застосунків, забезпечуючи безпеку даних на всіх етапах передавання.

Адаптивна швидкість передавання даних: LoRaWAN використовує технологію адаптивної швидкості передавання даних (Adaptive Data Rate, ADR), яка автоматично оптимізує швидкість передавання даних і потужність передавання для кожного пристрою на основі його середовища та переданих даних.

Геолокація (продовження): LoRaWAN підтримує геолокацію без використання GPS, що може бути корисно для відстеження місця розташування пристроїв у додатках, де GPS може бути недоступний або занадто енергоємним.

Трикласова модель: LoRaWAN підтримує три класи пристроїв (A, B і C) для різних типів застосунків. Клас A оптимізовано для пристроїв із низьким енергоспоживанням, клас B підтримує заплановані приймальні слоти для більш передбачуваної поведінки, а клас C надає майже постійну слухаючу здатність завдяки збільшеному енергоспоживанню.

Багаторазова активація: LoRaWAN підтримує два типи процедур активації пристрою: активацію за персональним ключем активації (ABP) і активацію за запитом (OTAA). OTAA кращий, оскільки він пропонує більшу безпеку за рахунок динамічного призначення ключів сеансу.

Затримка: Час передачі в LoRaWAN може бути відносно високим, близько секунд, що робить його менш підходящим для додатків, що вимагають низької затримки. Однак для більшості додатків IoT, висока затримка не є критичною [5].

1.2.3 Метод модуляції LoRaWAN

LoRaWAN використовує метод модуляції даних, який називається LoRa, розроблений компанією Semtech. LoRa є скороченням від "Long Range", що має на увазі довгу дистанцію передачі даних.

LoRa використовує метод модуляції під назвою частотно-маніпульоване чирп-спектральне розгортання (Chirp Spread Spectrum, CSS), який представляє собою форму розгортання спектра. Це дає змогу передачі даних бути дуже стійкою до перешкод і дає змогу сигналу проходити на великі відстані, водночас споживаючи відносно мало енергії [1].

У CSS, сигнал "чірп" є сигналом, який збільшує (або зменшує) свою частоту протягом певного часу. Це створює дуже унікальний патерн, який дає змогу приймачу легко розпізнати сигнал, навіть за умов високого рівня шуму або перешкод.

LoRa також використовує адаптивні швидкості передачі даних для управління швидкістю передачі залежно від умов каналу. Це означає, що коли канал зв'язку хороший, LoRa може використовувати вищу швидкість передачі даних для збільшення пропускної здатності. Коли канал стає гіршим (наприклад, через збільшення відстані або рівня перешкод), LoRa може зменшити швидкість передавання даних, щоб поліпшити стійкість передавання [4].

Такі можливості роблять LoRa ідеальним для багатьох додатків IoT, які вимагають довгого радіусу дії та низького енергоспоживання, але не потребують високої швидкості передачі даних.

1.2.4 Безпека мережі LoRaWAN

LoRaWAN використовує кілька механізмів для забезпечення безпеки мережі та даних, які передаються через мережу.

Шифрування даних: LoRaWAN використовує два рівні шифрування даних. На рівні мережі використовується AES-128 (Advanced Encryption Standard) для забезпечення безпеки та цілісності даних, що передаються між вузлами та шлюзами. На рівні застосунку також використовується AES-128 для шифрування самих даних, щоб забезпечити їхню конфіденційність. Це означає, що навіть якщо хтось перехопить дані, він не зможе їх прочитати без відповідного ключа.

Аутентифікація пристроїв: LoRaWAN використовує унікальні ідентифікатори пристроїв і мережеві ключі сесії для аутентифікації пристроїв, перш ніж вони можуть приєднатися до мережі. Це запобігає несанкціонованому доступу до мережі.

Цілісність повідомлення: LoRaWAN використовує механізм під назвою MIC (Message Integrity Code), щоб гарантувати, що повідомлення не були змінені в процесі передачі. Якщо повідомлення було змінено, воно буде відкинуто.

Лічильники послідовності повідомлень: LoRaWAN використовує лічильники послідовності повідомлень, щоб запобігти повторному передаванню повідомлень (так звані replay attacks), коли зловмисник намагається повторно надіслати зашифроване повідомлення [4].

У LoRaWAN 1.1 було внесено додаткові поліпшення в галузі безпеки, включно з окремими ключами мережі та застосунків для кожної сесії, поліпшеними процедурами приєднання та додатковими механізмами безпеки для мультисегментних мереж.

1.2.5 Алгоритм контролю доступу LoRaWAN

У LoRaWAN, механізм контролю доступу до середовища (Medium Access Control, MAC) заснований на ALOHA з деякими поліпшеннями. Це означає, що вузли (end-devices) в LoRaWAN можуть передавати дані в будь-який час, коли вони мають дані для передачі (це називається "Pure ALOHA").

Однак є кілька ключових механізмів, які допомагають керувати передачею даних і поліпшити ефективність використання мережі[4]:

Вікна прийому (Receive Windows): Щоразу, коли пристрій LoRaWAN (end-device) передає пакет даних, він відкриває два послідовних вікна для приймання даних. Перше вікно відкривається приблизно через одну секунду після закінчення передачі, а друге вікно - через дві секунди після закінчення передачі. Якщо мережа (через шлюз) має будь-які дані для передачі пристрою, вона може зробити це в ці вікна. Це дає змогу мережі керувати двостороннім зв'язком і управляти активністю пристроїв.

Швидкість передачі даних (Data Rate): LoRaWAN підтримує широкий діапазон швидкостей передачі даних, від дуже низьких до відносно високих. Низька швидкість передавання даних дає змогу передавати дані на великі відстані та через перешкоди, але займає більше часу та енергії. Висока швидкість передачі даних дає змогу швидше передавати дані, але на коротші відстані. Мережа може керувати швидкістю передачі даних пристроїв через механізм під назвою ADR (Adaptive Data Rate), що дає змогу оптимізувати енергоспоживання та використання радіочастотного спектра.

Дьюті цикл (Duty Cycle) і регулювання передачі (Transmission Regulation): У деяких регіонах, наприклад у Європі, радіочастотне законодавство обмежує кількість часу, протягом якого пристрій може передавати дані. Це зазвичай виражається у вигляді дьюті-циклу, наприклад, 1% дьюті-цикл означає, що пристрій може передавати дані не більше 1% часу. Це запобігає

перевантаженню радіочастотного спектра і гарантує, що всі пристрої мають рівний доступ до середовища.

Планування передачі (Transmission Scheduling): У класах B і C LoRaWAN мережа може планувати передачу даних до пристроїв. У класі B мережа і пристрій узгоджують "пінг-слоти" - зумовлені тимчасові інтервали, коли пристрій прослуховує можливі передачі даних від мережі. Це дає змогу мережі планувати передачу даних і керувати активністю пристроїв. У класі C, пристрій постійно прослуховує можливі передачі даних, коли він не передає дані сам, що дає змогу мережі в будь-який час надсилати дані пристрою, але споживає більше енергії [1].

Ці механізми контролю доступу до середовища дають змогу LoRaWAN підтримувати велику кількість пристроїв у мережі, забезпечуючи ефективне використання радіочастотного спектра та мінімізуючи енергоспоживання пристроїв.

LoRaWAN визначає три класи пристроїв (Class A, B і C), які забезпечують різні рівні функціональності та енергоефективності:

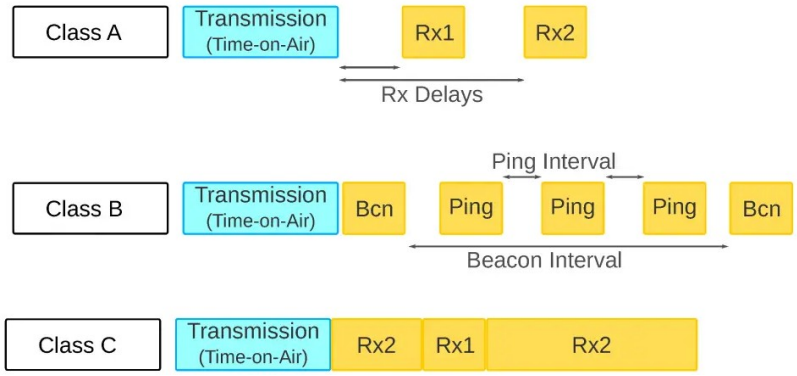

Рис. 1.2 Класи пристроїв [3]

Class A - Пристрої з найнижчим енергоспоживанням: Це найбільш загальний клас пристроїв у LoRaWAN. Пристрої Class A ініціюють зв'язок із

середовищем, і тільки після передавання вони відкривають два коротких вікна для приймання. Це дає змогу пристроям працювати на батарейках упродовж кількох років, оскільки вони зазвичай перебувають у режимі сну і споживають мінімальну кількість енергії.

Class B - Пристрої з додатковими планованими вікнами приймання: Пристрої Class B мають усі функції Class A, але з додатковими вікнами приймання, що відкриваються в плановані часи. Це дає змогу мережі ініціювати передавання в ці "пінг-слоти", що робить їх більш придатними для застосунків, які потребують частішого оновлення даних.

Class C - Пристрої з максимальною доступністю: Пристрої Class C майже постійно слухають (крім тих моментів, коли вони самі передають дані). Це забезпечує мінімальну затримку під час приймання, але значно збільшує енергоспоживання порівняно з Class A і B. Цей клас підходить для додатків, що вимагають високої доступності, і які не обмежені в енергоспоживанні [4].

Вибір класу для пристрою залежить від конкретних вимог застосунку, як-от необхідна енергоефективність, частота оновлення даних і необхідна затримка.

1.3 Протокол NB-IoT

NB-IoT, або Narrowband Internet of Things, - це технологія бездротового зв'язку, розроблена спеціально для інтернету речей (IoT). Вона була стандартизована 3GPP (3rd Generation Partnership Project) у релізі 13, який було опубліковано у 2016 році.

NB-IoT призначена для підключення великої кількості простих пристроїв у широкому географічному районі, які передають невеликі обсяги даних на низьких швидкостях.

NB-IoT - це технологія ліцензованого спектра, що означає, що оператори мобільного зв'язку повинні придбати ліцензії на використання радіочастот від уряду, щоб розгорнути мережі NB-IoT. Це відрізняє NB-IoT від технологій безліцензійного спектра, таких як LoRaWAN і Sigfox [6].

1.3.1 Архітектура мережі NB-IoT

Архітектура мережі NB-IoT ґрунтується на вже існуючій інфраструктурі стільникового зв'язку, що дає їй змогу легко інтегруватися з мережами 4G і 5G. Ось основні елементи архітектури NB-IoT:

NB-IoT Пристрої: Це "кінцеві точки" в мережі NB-IoT, які збирають і передають дані. Ці пристрої можуть бути дуже різноманітними залежно від конкретного застосування, включно з датчиками температури, вологості, руху, лічильниками води або електроенергії та багатьма іншими. Вони зазвичай мають низьке енергоспоживання і можуть працювати на батарейках протягом декількох років.

Базові станції (eNodeB): Базові станції слухають і приймають сигнали від пристроїв, а потім передають ці дані в мережу. Вони також відповідають за передачу сигналів назад до пристроїв. Базові станції можуть обслуговувати велику кількість пристроїв у своїй зоні покриття.

Сервери керування мережею: Ці сервери керують базовими станціями та мережевою інфраструктурою. Вони відповідають за маршрутизацію повідомлень між пристроями та додатками, підтримують безпеку мережі та можуть обробляти дані для подання в зручній формі.

Ядро мережі (Core Network): Ядро мережі - це центральна частина мережі, яка забезпечує зв'язок між різними базовими станціями, керує підключенням пристроїв і забезпечує безпеку. Воно також надає підключення до зовнішніх

мереж, таких як інтернет, що дає змогу передавати дані від пристроїв до кінцевих застосунків і назад.

Застосунки та послуги: Щойно дані зібрано і передано через мережу, їх зазвичай використовують застосунки для аналізу або виконання будь-яких дій. Це може включати в себе системи моніторингу, автоматизації, аналітики та багато інших.

NB-IoT використовує технології стільникового зв'язку для забезпечення зв'язку між пристроями і мережею, що дає змогу їй пропонувати широке покриття і високу проникаючу здатність сигналу, а також підтримку великої кількості пристроїв.

Способи розгортання: NB-IoT може бути розгорнута трьома різними способами [6]:

- У смузі LTE: Тут NB-IoT використовує ресурси в існуючій смузі LTE. Це дає змогу операторам легко додати підтримку NB-IoT в їхні наявні мережі.
- Поза смугою LTE: У цьому випадку NB-IoT використовує спектр поза існуючою смугою LTE. Це може бути корисно, якщо в оператора є вільний спектр, який він хоче використовувати для NB-IoT.
- В окремому спектрі: Тут NB-IoT використовує абсолютно окремий спектр. Це може бути корисно для операторів, які хочуть створити мережу тільки для NB-IoT.

Ця гнучкість у розгортанні робить NB-IoT привабливою для багатьох операторів мобільного зв'язку.Всі ці деталі засновані на стандартах 3GPP, офіційної організації, яка стандартизувала NB-IoT.

1.3.2 Технічні характеристики NB-IoT

Ширина смуги: NB-IoT використовує вузьку смугу в 200 кГц для зв'язку в обох напрямках. Це значно менше, ніж ширина смуги, яка використовується в більшості мобільних мереж. Вузька смуга допомагає знизити енергоспоживання і збільшити дальність зв'язку.

Дальність: NB-IoT забезпечує дальність зв'язку, що досягає декількох кілометрів у міських умовах і до десятків кілометрів у сільській місцевості. Це досягається завдяки використанню вузькосмугового зв'язку та оптимізації технології для довгого радіусу.

Кількість підключень: NB-IoT здатний обслуговувати до 50 000 пристроїв на кожну комірку мережі. Це забезпечує підтримку великої кількості пристроїв IoT, що важливо, враховуючи масштаби потенційних застосувань IoT.

Енергоспоживання: Пристрої NB-IoT зазвичай мають дуже низьке енергоспоживання і можуть працювати на батарейках протягом декількох років. Це досягається завдяки використанню схеми керування живленням, відомої як PSM (Power Saving Mode), і схеми eDRX (extended Discontinuous Reception), які дають змогу пристрою переходити в сплячий режим між передаваннями даних.

Швидкість передачі даних: NB-IoT забезпечує низьку швидкість передачі даних, зазвичай близько 200 кбіт/с. Цього достатньо для більшості застосунків IoT, які вимагають передавання тільки невеликих обсягів даних, наприклад, датчики температури або датчики вологості.

Повторення передачі: NB-IoT використовує повторення передачі для збільшення надійності зв'язку. Це означає, що дані можуть бути передані кілька разів для забезпечення їхньої доставки, особливо в умовах складного радіосередовища.

Затримка: Час передачі в NB-IoT може бути відносно високим, близько секунд, що робить його менш придатним для застосунків, які потребують низької затримки. Однак для більшості застосунків IoT, як-от зчитування

показань із датчиків або відстеження місця розташування, висока затримка не є критичною, оскільки дані не потребують реального часу для передавання [7].

Безпека: NB-IoT включає вбудовані функції безпеки для захисту даних і пристроїв. Аутентифікація пристроїв забезпечує, щоб тільки авторизовані пристрої могли підключатися до мережі. Шифрування даних захищає інформацію, що передається між пристроєм і мережею, запобігаючи її перехопленню та несанкціонованому доступу. Ці функції безпеки особливо важливі в контексті IoT, де пристрої можуть бути розкидані по всьому світу і схильні до різних загроз.

1.3.3 Безпека мережі NB-IoT

Технологія NB-IoT була розроблена з урахуванням вимог безпеки для пристроїв інтернету речей. Вона включає в себе кілька ключових функцій безпеки [8]:

Аутентифікація пристроїв: Це гарантує, що кожен пристрій, підключений до мережі NB-IoT, є довіреним і має право підключатися до мережі. Це здійснюється за допомогою процедури обміну ключами, під час якої пристрій і мережа обмінюються унікальними ключами для підтвердження ідентичності один одного.

Шифрування даних: Усі дані, що передаються між пристроєм і мережею, шифруються, щоб запобігти їхньому перехопленню і незаконному використанню. Це здійснюється за допомогою алгоритму шифрування, який використовує унікальні ключі, обміняні під час процедури аутентифікації.

Цілісність даних: Технологія NB-IoT також забезпечує захист цілісності даних, щоб гарантувати, що дані, які передаються між пристроєм і мережею, не були змінені в процесі передачі. Це досягається за допомогою алгоритму

контролю цілісності, який дає змогу пристрою та мережі перевірити, чи не були дані змінені після їхнього відправлення.

Захист від повторних атак: NB-IoT включає механізми для захисту від так званих "атак повторного відтворення", коли зловмисник намагається перехопити і повторно відправити пакети даних. Це досягається шляхом включення лічильника в кожен пакет даних, який збільшується з кожною новою передачею. Якщо мережа отримує пакет із застарілим значенням лічильника, вона відхиляє пакет.

1.3.4 Алгоритм контролю доступу NB-IoT

Технологія NB-IoT (Narrowband Internet of Things) використовує керовану мережею модель доступу, яка регулюється базовою станцією. Вона забезпечує координацію між пристроями для ефективного використання радіочастотного спектра та запобігання зіткненням.

Ось деякі ключові аспекти контролю доступу в мережах NB-IoT [6]:

Черговість передачі: У мережах NB-IoT використовується черговість передачі, яка визначається базовою станцією. Коли пристрій готовий передати дані, він повинен запросити дозвіл у базової станції. Базова станція потім визначає, коли пристрій може передати дані, щоб мінімізувати зіткнення і забезпечити ефективне використання радіочастотного спектра.

Поділ часу і частоти: NB-IoT використовує схему поділу часу і частоти (TDMA/FDMA) для контролю доступу до каналу. Це означає, що радіочастотний спектр розділений на тимчасові слоти і частотні канали, і кожному пристрою присвоюється певний часовий слот і частотний канал для передачі даних. Це дає змогу пристроям працювати паралельно, мінімізуючи зіткнення та забезпечуючи високу пропускну здатність.

Енергоефективність: Технологія NB-IoT спроектована для забезпечення енергоефективності, що особливо важливо для пристроїв IoT, які часто працюють від батареї. Алгоритм контролю доступу оптимізовано для мінімізації енергоспоживання, даючи змогу пристроям залишатися в режимі очікування, коли вони не передають дані, і прокидатися тільки для передавання даних і приймання підтверджень.

Підтримка різних класів пристроїв: NB-IoT здатна підтримувати різні класи пристроїв з різними вимогами до передачі даних. Наприклад, деякі пристрої можуть вимагати періодичного передавання даних (наприклад, лічильники обліку енергії), тоді як інші можуть вимагати випадкового або рідкісного передавання даних (наприклад, датчики диму). Алгоритм контролю доступу NB-IoT здатний керувати цими різними вимогами, забезпечуючи ефективне використання радіочастотного спектра та енергії.

Випадковий доступ: У разі, коли пристрій має передати дані поза своїм звичайним часовим слотом (наприклад, у разі термінової тривоги), NB-IoT підтримує механізм випадкового доступу. Пристрій може надіслати запит на передачу даних, і якщо канал вільний, базова станція може дозволити передачу.

Насамкінець, алгоритм контролю доступу NB-IoT призначений для забезпечення ефективного використання радіоресурсів і енергії, а також підтримки різних вимог до передачі даних пристроїв IoT. Він використовує комбінацію черговості передачі, поділу часу і частоти, енергоефективності та підтримки різних класів пристроїв для досягнення цих цілей.

1.4 Протокол Sigfox

Sigfox - це глобальна телекомунікаційна технологія LPWAN (Low-Power Wide Area Network), розроблена для зв'язку пристроїв Інтернету речей (IoT) з

низьким енергоспоживанням і вимогами до дальності передачі даних. Вона була створена з метою надання простого, надійного та енергоефективного рішення для зв'язку між пристроями IoT і хмарною інфраструктурою [9].

1.4.1 Архітектура мережі Sigfox

З точки зору архітектури мережі, Sigfox складається з трьох основних компонентів: пристроїв, базових станцій і хмарної платформи. Ось докладний опис архітектури мережі Sigfox з наукової точки зору [10]:

Рис.1.3 Архітектура мережі Sigfox [10]

Пристрої (End Devices): Sigfox пристрої являють собою кінцеві точки в мережі, які збирають і передають дані. Це можуть бути різні пристрої, включно з сенсорами, контролерами та іншими IoT пристроями. Пристрої використовують ультра-вузькосмугову (UNB) радіомодуляцію, що дає їм змогу ефективно використовувати радіоспектр і мінімізувати енергоспоживання. Це означає, що пристрої можуть працювати на батарейках протягом декількох років без заміни.

Базові станції (Base Stations): Базові станції - це проміжні точки, які приймають дані від пристроїв і пересилають їх у хмарну платформу Sigfox. Вони розташовані на значній відстані одна від одної, що забезпечує широку зону покриття. Наприклад, у сільській місцевості одна базова станція може обслуговувати радіус до 50 км, а в міському середовищі - до 10 км.

Хмарна платформа (Sigfox Cloud): Після того, як дані були передані на базову станцію, вони відправляються в хмарну платформу Sigfox. Ця платформа обробляє і зберігає дані, прийняті від базових станцій. Вона також надає API, які дають змогу розробникам легко інтегрувати дані Sigfox з іншими додатками та сервісами.

Безпека і шифрування: Sigfox забезпечує безпеку своєї мережі за допомогою механізмів шифрування та аутентифікації. Дані, що передаються через мережу, захищені за допомогою криптографічного шифрування. Крім того, кожен пристрій має унікальний ідентифікатор, який використовується для аутентифікації пристрою перед передачею даних.

Підтримка глобальних частот: Sigfox підтримує різні радіочастоти в різних регіонах світу. У Європі, наприклад, Sigfox працює на частоті 868 МГц, тоді як у Північній Америці використовується частота 902 МГц. Це забезпечує глобальну сумісність пристроїв і дає змогу розгорнути Sigfox у будь-якій частині світу, з урахуванням місцевих регуляторних вимог до радіочастот [12].

Також варто зазначити, що Sigfox використовує "зіркоподібну" топологію мережі, що означає, що всі пристрої безпосередньо підключаються до базових станцій, а не один до одного. Це спрощує мережеву архітектуру та зменшує енергоспоживання пристроїв.

На закінчення, головна перевага Sigfox полягає в тому, що ця мережа призначена для передавання малих обсягів даних на великі відстані з мінімальним енергоспоживанням.

1.4.2 Технічні характеристики Sigfox

Ультра-вузькосмугова модуляція (Ultra-Narrowband, UNB): Sigfox використовує технологію UNB, яка передає дуже малі обсяги інформації через дуже вузьку смугу частот (близько 100 Гц). Це дає змогу мережі Sigfox бути стійкою до шумів і перешкод, а також збільшує дальність передавання даних. UNB також забезпечує високу енергоефективність, що дає змогу пристроям працювати на батареях протягом тривалого часу.

Покриття та дальність: На відміну від традиційних мереж стільникового зв'язку, які вимагають щільної мережі базових станцій, Sigfox потребує значно менше станцій для забезпечення покриття. Одна базова станція може обслуговувати радіус до 50 км у сільській місцевості та до 10 км у міських умовах. Це значно знижує вартість інфраструктури та забезпечує широку географічну доступність.

Енергоефективність: Sigfox-пристрої ефективно використовують енергію завдяки низькій пропускній здатності та UNB. Сигнали можуть бути передані з дуже низьким рівнем потужності (близько 25 мВт), що дає змогу пристроям працювати на батареях протягом декількох років без заміни [11].

Пропускна здатність і обсяг даних: Sigfox обмежує розмір повідомлень до 12 байт і кількість повідомлень до 140 на день для кожного пристрою. Це забезпечує низьку пропускну здатність, що зменшує енергоспоживання і збільшує дальність передачі.

1.4.3 Безпека мережі Sigfox

Безпека є критичним аспектом будь-якої мережі Інтернету речей, і Sigfox забезпечує кілька рівнів захисту для забезпечення безпеки даних і пристроїв у мережі.

Аутентифікація пристроїв: Кожен пристрій Sigfox має унікальний ідентифікатор (ID), який використовується для аутентифікації пристрою під час під'єднання до мережі. Це гарантує, що тільки зареєстровані та довірені пристрої можуть підключатися до мережі та передавати дані.

Шифрування: Усі дані, що передаються через мережу Sigfox, шифруються. Це запобігає перехопленню даних зловмисниками і гарантує конфіденційність даних користувача.

Анти-ретрансляційний захист (Replay Attack Protection): Sigfox використовує методи захисту від ретрансляційних атак. Це запобігає атакам, за яких зловмисник перехоплює і повторно передає валідні дані з метою ввести систему в оману.

Секретні ключі: Sigfox використовує секретні ключі для шифрування даних і аутентифікації пристроїв. Ці ключі зберігаються в захищеному вигляді та недоступні для сторонніх осіб.

Захист від атак типу "людина посередині" (Man-in-the-Middle Attacks): Sigfox використовує методи захисту від атак "людина посередині", під час яких зловмисник намагається перехопити і змінити дані, що передаються між двома сторонами.

Сигнальний захист: Sigfox використовує технологію ультра-вузькосмугового зв'язку, яка забезпечує стійкість до перешкод і ускладнює перехоплення та аналіз сигналів [9].

Однак варто зазначити, що, як і будь-яка технологія, Sigfox не може гарантувати 100% безпеки. Тому важливо, щоб розробники і користувачі пристроїв також дотримувалися найкращих практик безпеки, таких як

забезпечення безпеки своїх пристроїв і використання безпечних методів передачі даних.

1.4.4 Алгоритм контролю доступу Sigfox

Технологія Sigfox заснована на принципі "заводського підключення" (star-on-star topology), де кожен пристрій безпосередньо зв'язується з базовою станцією. У цьому контексті, алгоритми контролю доступу насамперед відносяться до методів, які пристрої використовують для забезпечення ефективного використання радіочастотного спектра і мінімізації зіткнень сигналів.

Random Time-Frequency Access: Цей метод контролю доступу використовується в мережах Sigfox для оптимізації використання радіочастотного спектра. Суть його в такому: коли пристрій готовий передати дані, він обирає випадковий час і випадкову частоту в межах заданого діапазону для передавання свого сигналу. Сигнали передаються на різних частотах, що дає змогу уникнути перекриття і зіткнень сигналів, що могло б призвести до втрати даних. Це важливо, оскільки в радіочастотному спектрі є обмежена кількість "простору" для передавання даних, і ефективне його використання критичне для забезпечення надійності та ефективності мережі.

Аутентифікація пристроїв: Кожен пристрій Sigfox має унікальний ідентифікатор (ID), який використовується для аутентифікації пристрою під час під'єднання до мережі. Цей ідентифікатор надається виробником пристрою і зареєстрований у мережі Sigfox. Коли пристрій намагається підключитися до мережі, він надає свій ID, і мережа перевіряє, чи є цей ID дійсним. Це гарантує, що тільки зареєстровані пристрої можуть підключатися і передавати дані через мережу, що підвищує безпеку мережі.

Обмеження на передачу даних: Sigfox встановлює обмеження на кількість даних, яку може передавати кожен пристрій протягом дня. Це робиться для запобігання перевантаженню мережі та забезпечення рівномірного розподілу пропускної здатності мережі між усіма підключеними пристроями. Обмеження становить 140 повідомлень на день на пристрій, що, як правило, достатньо для більшості додатків Інтернету речей.

Стандарти радіочастот: Sigfox, як і будь-який інший оператор бездротового зв'язку, повинна відповідати стандартам і регуляторним вимогам для радіочастотних служб. Це включає в себе обмеження на використання певних частот, потужність передачі та інші параметри, які можуть впливати на інші служби радіозв'язку. Дотримуючись цих стандартів, Sigfox забезпечує сумісність зі своїми мережами та іншими радіочастотними службами, мінімізуючи можливі перешкоди та конфлікти [12].

У багатьох країнах існують специфічні регуляторні вимоги, які визначають, які частоти можна використовувати для яких видів служб, які обмеження щодо потужності передавача тощо. Sigfox повинна відповідати цим вимогам, щоб забезпечити свою діяльність у різних юрисдикціях. Це включає в себе отримання відповідних ліцензій і дозволів від регуляторних органів, а також постійне дотримання вимог і регулятивних стандартів.

Крім того, Sigfox активно використовує методи управління радіочастотним спектром, щоб забезпечити ефективне використання доступних частот і мінімізувати ймовірність перешкод або зіткнень сигналів. Це включає в себе використання технік, таких як адаптивна модуляція і кодування, які дозволяють збільшувати пропускну здатність і надійність мережі .

1.5 Протокол LTE-M

LTE-M (LTE for Machines), також відома як LTE Cat M1, це технологія стільникового зв'язку, розроблена для інтернету речей (IoT) і машинних застосунків (M2M). Вона є частиною стандарту 3GPP і призначена для підтримки малоінтенсивних IoT-пристроїв з низьким енергоспоживанням і тривалим терміном служби батареї [13].

1.5.1 Архітектура мережі LTE-M

LTE-M, як і стандартний LTE, базується на архітектурі, яка заснована на моделі все в IP (All-IP). Це означає, що всі дані, включно з голосовими даними, передаються як IP-пакети .

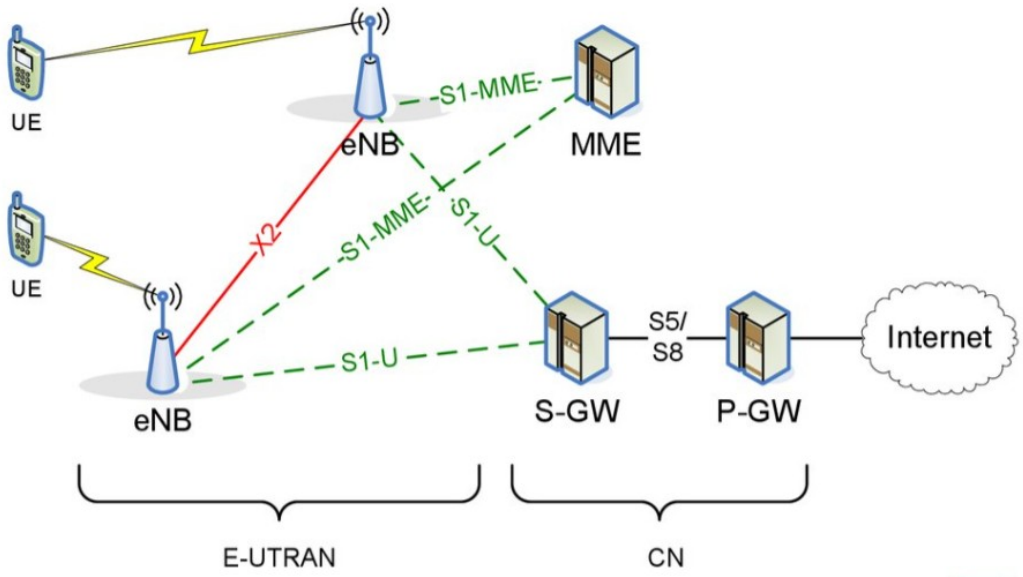

Рис. 1.4 Архітектура мережі LTE-M [14]

eNodeB (Базова станція): eNodeB, у контексті LTE-M, містить у собі функції радіоуправління, як-от управління радіоресурсами, управління інтерференцією та управління мобільністю на рівні комірки. Вони також виконують завдання з передачі даних у прямому і зворотному каналах. Важливо відзначити, що базові станції можуть бути оновлені програмним забезпеченням

для підтримки стандарту LTE-M, що означає, що оператори можуть впровадити LTE-M, не вносячи значних змін в існуючу інфраструктуру.

Мережа Evolved Packet Core (EPC): EPC - це основний компонент мережі LTE, який відповідає за управління сеансами, маршрутизацію і передачу пакетів даних. EPC складається з таких функціональних елементів:

MME (Mobility Management Entity): MME є ключовим вузлом управління в мережі LTE. Вона відповідає за управління ідентифікацією пристроїв, відстеженням їхнього місця розташування, автентифікацією, авторизацією, встановленням маршрутів для призначеного для користувача трафіку і управлінням станами підключення пристроїв.

SGW (Serving Gateway): SGW діє як маршрутизатор і точка передачі даних між базовими станціями (eNodeB) і PGW. Вона також відіграє важливу роль в управлінні мобільністю, особливо під час переходу між різними eNodeB.

PGW (Packet Data Network Gateway): PGW забезпечує підключення до зовнішніх IP-мереж, таких як інтернет. Вона виконує функції IP-адресації, розподілу трафіку і управління зарядом. Крім цього, PGW також може забезпечувати підтримку якості обслуговування (QoS) для різних типів трафіку.
HSS (Home Subscriber Server): HSS - це основна база даних для зберігання та управління інформацією про передплатників. HSS також відповідає за підтримку процедур переміщення і роумінгу. У контексті LTE-M, HSS відіграватиме важливу роль у забезпеченні безпеки та аутентифікації пристроїв. Вона забезпечує збереження інформації про передплатників і їхні служби, що дає змогу керувати великою кількістю підключень, характерних для IoT-мереж.

Важливо зазначити, що багато з цих функцій і компонентів архітектури мережі LTE-M можуть бути оптимізовані або налаштовані спеціально для IoT-додатків. Наприклад, функції управління енергією, як-от PSM (Power Saving

Mode) та eDRX (Extended Discontinuous Reception), можуть бути впроваджені та налаштовані для максимізації тривалості життя батареї IoT-пристроїв [13].

Також варто зазначити, що LTE-M розроблено для забезпечення сумісності зі стандартними LTE-мережами. Це означає, що оператори мобільного зв'язку можуть запропонувати IoT-послуги у своїх наявних LTE-мережах з мінімальними змінами інфраструктури. Це робить LTE-M дуже привабливим варіантом для операторів, які хочуть впровадити IoT-послуги на своїх мережах.

1.5.2 Технічні характеристики LTE-M

Пропускна здатність: Пропускна здатність LTE-M може досягати 1 Мбіт/с у прямому і зворотному каналах. Цього достатньо для більшості IoT-додатків, які не вимагають високої пропускної спроможності, але потребують стабільного та надійного підключення. Наприклад, сенсори, пристрої моніторингу та системи відстеження.

Затримка: LTE-M має низьку затримку в районі 10-15 мс, що робить його придатним для застосунків, які потребують швидкого часу відгуку, як-от аварійні сповіщення, системи керування та моніторингу в реальному часі.

Споживання енергії: Режими енергозбереження PSM і eDRX дають змогу пристроям тривалий час перебувати в стані очікування, істотно знижуючи споживання енергії. Це критично важливо для пристроїв, що працюють від батареї, які повинні функціонувати протягом тривалого часу без заміни або зарядки.

Щільність підключень: LTE-M здатний підтримувати до 1000 пристроїв на комірку. Це означає, що він може обслуговувати велику кількість пристроїв у

невеликій ділянці, що робить його ідеальним для міських середовищ і промислових підприємств із великою кількістю IoT-пристроїв.

Мобільність: LTE-M підтримує мобільність пристроїв і роумінг між різними мережами та країнами. Це означає, що пристрої можуть пересуватися і залишатися під'єднаними, що ідеально підходить для транспортних і логістичних додатків [15].

Зворотна сумісність: LTE-M може бути впроваджений в існуючі LTE-мережі без необхідності заміни обладнання. Це означає, що оператори можуть впроваджувати IoT-послуги швидко і з мінімальними витратами.

1.5.3 Безпека мережі LTE-M

LTE-M використовує багаторівневу систему безпеки, яка забезпечує захист на рівні мережі, передавання даних і пристрою. Ось більш детальна інформація про кожен із цих рівнів:

Безпека на рівні мережі: У мережі LTE-M використовується автентифікація мережі та пристрою на основі симетричного ключа, який зберігається і на SIM-карті пристрою, і в мережі оператора. Цей процес автентифікації гарантує, що пристрій і мережа можуть довіряти один одному і що пристрій має право отримати доступ до мережі.

Безпека передачі даних: Дані, що передаються між пристроєм і мережею, шифруються для забезпечення їхньої конфіденційності та цілісності. Алгоритми шифрування, такі як AES (Advanced Encryption Standard), зазвичай використовуються для цієї мети. Шифрування даних допомагає захистити інформацію від перехоплення і модифікації.

Безпека пристрою: Пристрої в мережі LTE-M можуть бути захищені різними механізмами, такими як захист від несанкціонованого доступу та атак

на апаратне забезпечення. Також можуть бути використані технології безпечного зберігання ключів і даних.

Безпека на рівні додатків: Додатки, що працюють на пристроях, також можуть бути захищені. Однак це залежить від конкретного застосунку та заходів безпеки, що застосовуються розробниками застосунків [15].

Важливо зазначити, що хоча LTE-M надає багаторівневу систему безпеки, повна безпека системи залежить від правильної конфігурації та управління мережею, а також від безпеки використовуваних пристроїв і додатків.

1.5.4 Алгоритм контролю доступу LTE-M

Алгоритми контролю доступу в LTE-M належать до механізмів, які визначають, як пристрої отримують доступ до мережі та використовують радіочастотні ресурси для передавання даних. У LTE-M використовуються такі основні алгоритми контролю доступу [13]:

Аутентифікація: Цей процес гарантує, що мережа і пристрої можуть взаємодіяти в безпечному середовищі. Процедура автентифікації в LTE-M базується на використанні SIM-карток (або eSIM), на яких зберігаються ключі шифрування, які також присутні в мережі оператора. Під час аутентифікації ці ключі використовуються для підтвердження ідентифікації пристрою та його права на доступ до мережі. Це важлива частина захисту від несанкціонованого доступу та атак.

Випадковий доступ: Процес випадкового доступу в LTE-M використовується пристроями для початку зв'язку з базовою станцією. Пристрій відправляє спеціальний сигнал, відомий як "запит на випадковий доступ", який потім обробляється базовою станцією. Цей сигнал містить інформацію,

необхідну для базової станції, щоб правильно розподілити радіочастотні ресурси для пристрою.

Динамічний розподіл ресурсів: У мережі LTE-M базові станції активно керують розподілом радіочастотних ресурсів серед пристроїв. Вони можуть адаптуватися до мінливих умов мережі та потреб пристроїв, динамічно змінюючи розподіл ресурсів для оптимізації продуктивності мережі. Базові станції можуть керувати часом передавання, частотами та рівнем потужності для кожного пристрою, щоб забезпечити ефективне використання ресурсів.

Адаптивна зміна модуляції та кодування (AMC): Це технологія, що дає змогу мережі LTE-M адаптуватися до мінливих умов радіоканалу. Залежно від якості каналу, базова станція може вирішити використовувати вищий або нижчий рівень модуляції, щоб збільшити пропускну здатність або зменшити помилки передавання. Це дає змогу мережі LTE-M забезпечувати надійну та ефективну передачу даних у різних умовах.

РОЗДІЛ 2. КОМПОНЕНТНА БАЗА

2.1 LoRa модуль sx1278.

Модулі SX1278 оснащені модемом дальнього радіусу дії LoRa, який забезпечує наддалекий зв'язок з розширеним спектром і високу завадостійкість при мінімальному споживанні струму.

Рис.2.1 Модуль sx1278 (Ra-02) [17]

2.1.1 Характеристики SX1278 (Ra-02) [5]:

- Вихідна потужність: до +18 dBm
- Чутливість: до - 127 dBm
- Програмована швидкість передавання даних може досягати 300 кбіт/с
- Модуляція: FSK, GFSK, MSK, GMSK, LoRa і OOK
- Частотний діапазон: 410 - 525 МГц
- Споживана потужність: 10.8 mA (прийом) і 120mA (передача)
- Пам'ять: 256 байт
- Робоча температура: -40°C до +85°C

Блок схема чіпа sx1278:

2.1.2 Види модуляції [5]:

FSK (Frequency Shift Keying): У FSK, інформація кодується в різні частоти. Це найбільш простий тип частотної модуляції. Коли біт даних змінюється з 0 на 1 або навпаки, частота несучої хвилі змінюється.

GFSK (Gaussian Frequency Shift Keying): GFSK - це модифікація FSK, де використовується Гауссовський фільтр для згладжування переходів між бітами. Це покращує спектральну ефективність за рахунок зменшення ширини смуги, необхідної для передачі сигналу.

MSK (Minimum Shift Keying): MSK - це ще одна форма FSK, де різниця в частоті між "0" і "1" становить половину швидкості символів. Це зменшує інтерференцію і робить MSK більш ефективною, ніж звичайний FSK.

GMSK (Gaussian Minimum Shift Keying): GMSK - це різновид MSK, що використовує Гауссівський фільтр для згладжування переходів між бітами, покращуючи таким чином спектральну ефективність.

LoRa (Long Range): LoRa - це технологія модуляції, розроблена спеціально для IoT і LPWAN. Вона використовує підхід, званий чирп-спред-спектрум, який забезпечує високу стійкість до перешкод і дає змогу передавати дані на великі відстані з низьким енергоспоживанням.

OOK (On-Off Keying): OOK - це найпростіший формат модуляції, де присутність сигналу може позначати один стан (наприклад, "1"), а його відсутність - інший (наприклад, "0").

2.2 Мікроконтролер ESP 32

ESP32 (Espressif Systems Product) - це потужний мікроконтролер від Espressif, який володіє вбудованими можливостями Wi-Fi і Bluetooth. Його

характеристики роблять його ідеальним вибором для різних додатків у сфері інтернету речей (IoT) [18].

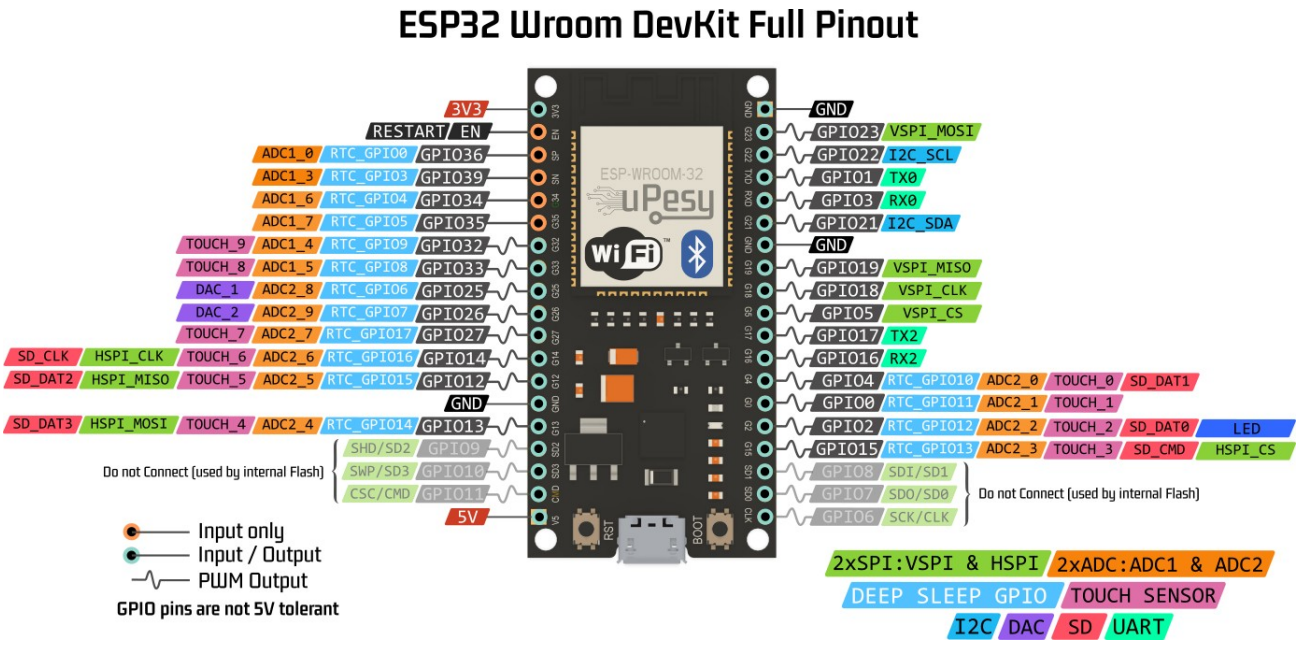

Рис.2.2 Порти ESP 32 [19]

2.2.1 Ключові характеристики ESP32 [18]:

- Процесор:

  Двоядерний Tensilica Xtensa LX6

  Тактова частота до 240 МГц
- Пам'ять:

  520 КБ внутрішнього SRAM

  4 МБ вбудованої Flash-пам'яті

  Підтримка зовнішньої Flash-пам'яті та SRAM
- Бездротовий зв'язок:

  Wi-Fi: 802.11b/g/n (2.4 ГГц), підтримка WPA/WPA2/WPA2-Enterprise/WPS

  Bluetooth: v4.2 BR/EDR і BLE (Bluetooth Low Energy)

- GPIO: До 36 GPIO-пінів (включно з 18 аналоговими входами)
- Підтримка різних функцій, включно з SPI, I2C, UART, CAN, IR, PWM, DAC, ADC :

SPI (Serial Peripheral Interface): ESP32 підтримує до 4 SPI-інтерфейсів, які можна використовувати для зв'язку з SPI-пристроями, як-от дисплеї та SD-карти. SPI надає швидкий напівдуплексний зв'язок між мікроконтролером і периферійними пристроями.

I2C (Inter-Integrated Circuit): I2C - це двопровідний серійний інтерфейс, який використовується для зв'язку між мікроконтролерами та іншими пристроями, такими як датчики, EEPROM та інші I/O інтерфейси. ESP32 підтримує множинне підключення пристроїв на одній шині I2C.

UART (Universal Asynchronous Receiver/Transmitter): ESP32 має 3 UART-інтерфейси. Одним з основних застосувань UART є обмін даними між мікроконтролером і комп'ютером або іншими пристроями, що підтримують серійний зв'язок.

CAN (Controller Area Network): ESP32 підтримує CAN 2.0 і забезпечує протокол зв'язку, який дає змогу різним пристроям в автомобілі або інших застосуваннях обмінюватися даними. Однак для використання CAN з ESP32 може знадобитися додатковий апаратний інтерфейс.

IR (Infrared): ESP32 може бути програмно налаштований для надсилання та приймання IR-сигналів, що дає йому змогу взаємодіяти з різними пристроями, як-от телевізори та кондиціонери.

PWM (Pulse Width Modulation): ESP32 підтримує до 16 каналів PWM, які можна використовувати для керування сервоприводами, світлодіодами та іншими пристроями, що вимагають модуляції ширини імпульсу.

ADC (Analog to Digital Converter): ESP32 інтегрує два 12-бітові SAR (Successive Approximation Register) АЦП, що підтримують загалом 18 каналів вимірювань (аналогові увімкнені контакти).

DAC (Digital to Analog Converter): це пристрій, який перетворює цифрові значення (зазвичай у двійковому форматі) на аналоговий сигнал (напруга або струм).

Захист і безпека:

- Підтримка захисту від запису і читання для Flash-пам'яті
  Апаратне прискорення для AES, HASH (SHA-2), RSA, ECC.
- Енергоспоживання:
  Різні режими сну для енергоефективності
  Напруга живлення: 2.2V-3.6V

2.3 Датчик чадного газу MQ-7 модуль.

MQ-7 - це модуль датчика моноксиду вуглецю (CO) на основі напівпровідникового матеріалу SnO2 (оксиду олова) (чутлива частина яка змінює опір при взаємодії з газом). Він часто використовується в різних системах для виявлення наявності моноксиду вуглецю в атмосфері [20].

Рис.2.3 MQ-7 модуль [21].

2.3.1 Характеристики MQ-7 [20]:

- Робоча напруга: Він працює на DC 5V, хоча може функціонувати в діапазоні від 5 до 9 вольт.
- Споживання струму: MQ-7 споживає близько 150 мА.
- Виходи: MQ-7 має два основні виходи - AO (аналоговий вихід) і DO (цифровий вихід).

  AO (аналоговий вихід): Цей вихід являє собою аналоговий сигнал, який змінюється залежно від концентрації моноксиду вуглецю. Цей сигнал можна під'єднати до аналогового входу мікроконтролера для читання й аналізу.

  DO (цифровий вихід): Цей вихід подає цифровий сигнал (HIGH або LOW), коли рівень моноксиду вуглецю досягає певного порога. Поріг можна налаштувати за допомогою вбудованого потенціометра на модулі.
- Робочий цикл: Для отримання точних даних з датчика MQ-7 потрібна періодичність у перемиканні режиму нагрівання. Протягом 60 секунд датчик має бути підігрітий до 1.4V, потім протягом 90 секунд до 5V. Це забезпечує правильність і точність даних.
- Температурний діапазон: MQ-7 зазвичай може працювати в діапазоні температур від -10 до +50 градусів Цельсія.
- Вологість: Відносна вологість має бути в діапазоні 95% RH і нижче.
- Робочий принцип:

  При збільшенні концентрації CO в повітрі, опір датчика зменшується. Резистор опору, під'єднаний до виводу AO, забезпечує зміну напруги на цьому виводі. Ця напруга може бути прочитана мікроконтролером або мікропроцесором для подальшого аналізу. Вивід DO видаватиме високий рівень сигналу (5V), якщо концентрація CO перевищить певний поріг, який можна налаштувати за допомогою потенціометра на платі модуля.

2.4 Датчик температури та вологості DHT11

DHT11 - це цифровий датчик температури та вологості [22].

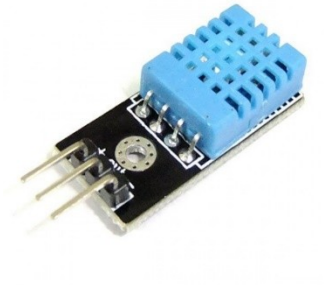

Рис. 2.4 DHT11 [23]

2.4.1 Характеристики [22]:

- Температурний діапазон: DHT11 може вимірювати температуру в діапазоні від 0 до 50 градусів Цельсія з точністю ±2 градуси.
- Діапазон вологості: DHT11 може вимірювати відносну вологість у діапазоні від 20% до 80% з точністю ±5%.
- Напруга живлення: DHT11 зазвичай працює від 3 до 5 вольт постійного струму.
- Споживання струму: У режимі очікування DHT11 споживає близько 0.5-2.5 мА, а під час активного вимірювання - до 2.5 мА.
- Інтерфейс: DHT11 використовує однопровідний інтерфейс для зв'язку з мікроконтролером. Це означає, що всі дані передаються по одному дроту.
- Частота оновлення даних: DHT11 оновлює вимірювання температури та вологості приблизно щосекунди. Однак між запитами має бути пауза щонайменше 2 секунди.
- Розміри: Розміри модуля DHT11 становлять 23mm x 12mm x 5mm, включно з чотирма виводами для підключення до мікроконтролера.

Важливо зазначити, що під час використання DHT11 важливо врахувати, що він має певні обмеження, включно з відносно низькою точністю та обмеженим діапазоном вимірювань, як порівняти з деякими іншими датчиками температури та вологості. Також DHT11 не підходить для застосунків, яким потрібне швидке оновлення даних.

Всі елементи було обрано із розрахунку ціна – якість та наявність на українському ринку.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СХЕМИ

3.1 Блок схема

Зображена нижче блок-схема являє собою архітектуру нашого каналу зв'язку, побудованого на основі модуля LoRa SX1278. У цій системі, ми маємо такі основні компоненти:

Передавач (Transmitter): У цьому блоці відбувається збір даних із двох різних датчиків - DHT-11 (відносна вологість і температура) і MQ-7 (детектор монооксиду вуглецю). Дані з датчиків надходять на обробку в мікроконтролер ESP32.

Мікроконтролер ESP32: Тут дані, отримані від датчиків, перетворяться у відповідний формат для передачі через LoRa SX1278.

Модуль LoRa SX1278: Цей модуль приймає оброблені дані від мікроконтролера і передає їх через радіоканал.

Приймач (Receiver): Цей блок приймає передані дані через свій власний модуль LoRa SX1278. Отримані дані потім передаються на обробку в другий мікроконтролер ESP32.

Другий мікроконтролер ESP32: Тут прийняті дані декодуються і обробляються для подальшого використання. Завдяки вбудованому Wi-Fi модулю, цей мікроконтролер зв'язується з нашим Telegram-ботом через відкритий API.

Telegram-бот: Після обробки дані відправляються в наш Telegram-бот, забезпечуючи нам миттєвий доступ до зібраної інформації.

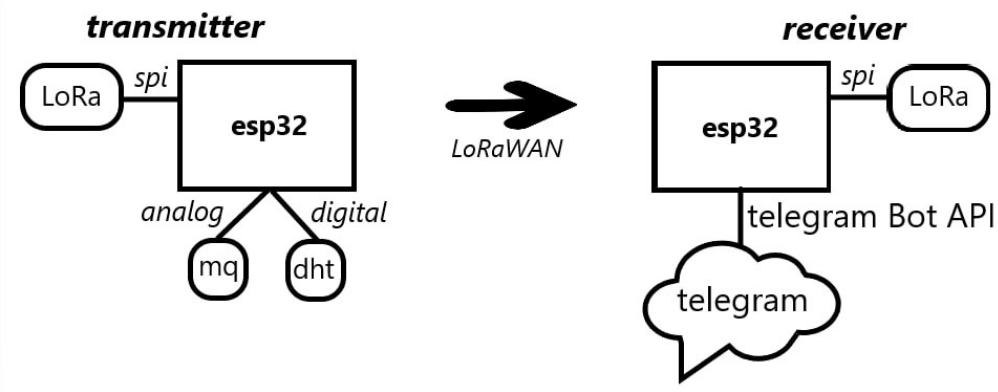

Рис.3.1Блок схема проекта

Порти для з’єднання LoRa SX1278 з ESP32:

LoRa SX1278	ESP32
3.3V	3.3V
GND	GND
NSS	D5
DIO0	D2
SCK	D18
MISO	D19
MOSI	D23
RST	D14

Рис.3.2 З'єднання портів

На представленій далі схемі (рис.3.3)ми спостерігаємо таку структуру. З лівого боку зображено пристрій-приймач, функціональне завдання якого полягає в прийомі інформації з використанням модуля зв'язку LoRa SX1278 від передавача та подальшої трансляції цієї інформації в телеграм-бота через API Telegram.

З правого боку представлено передавач, який здійснює процес зчитування даних із датчиків DHT11 і MQ7. Отримана інформація пересилається з використанням модуля зв'язку LoRa SX1278, забезпечуючи таким чином з'єднання з приймальним пристроєм.

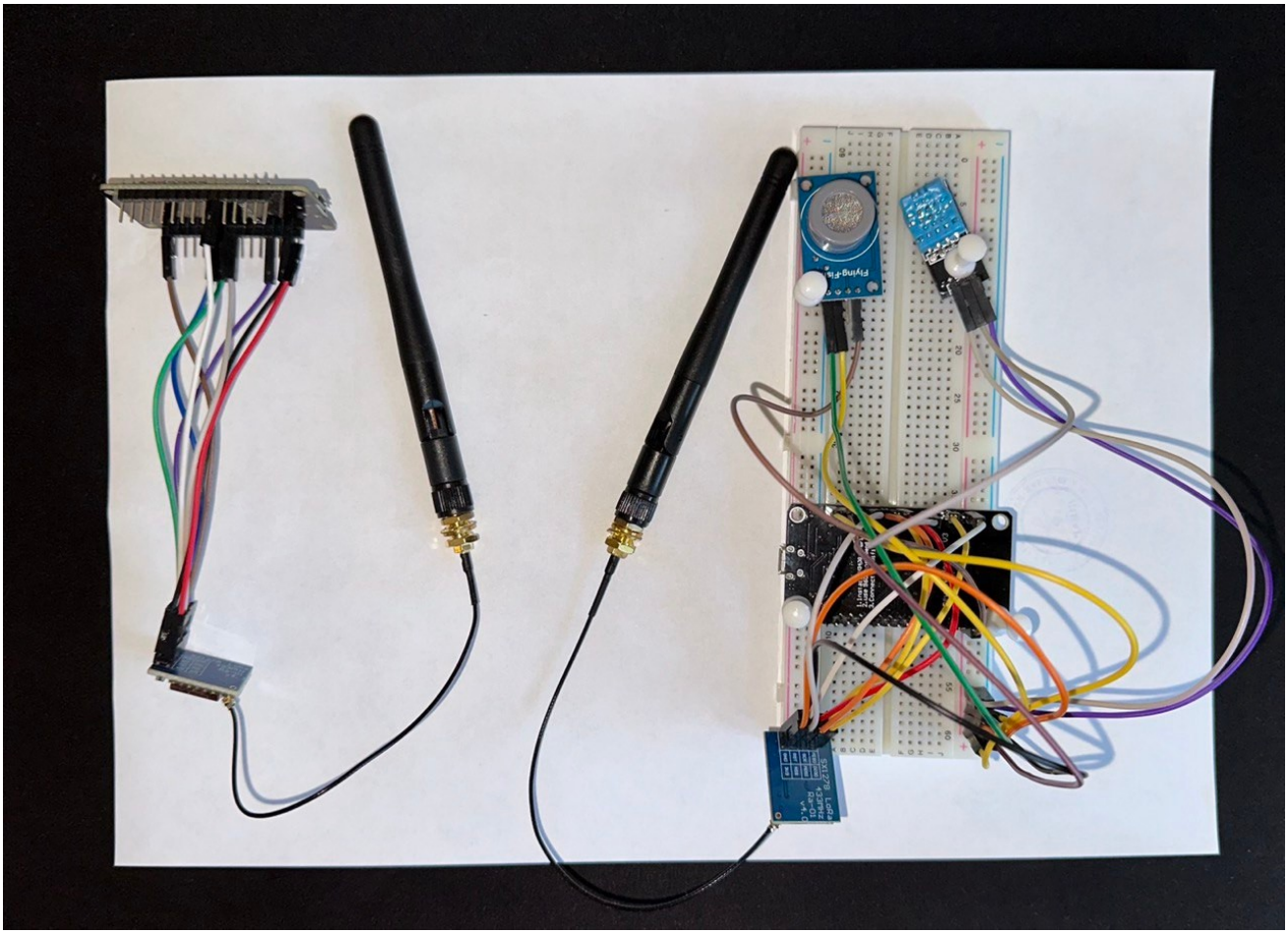

Рис.3.3 Зібрана схема

3.2 Кодова реалізація

3.2.1 Приймач:

Підключення бібліотек: На початку файлу підключаються всі необхідні бібліотеки: SPI (для обміну даними між LoRa та мікроконтролером), LoRa (для зв'язку між передавачем та приймачем), WiFi і WiFiClientSecure (для підключення до мережі та забезпечення безпечного з'єднання), UniversalTelegramBot (для роботи з Telegram API), і ArduinoJson (для роботи з JSON даними).

Налаштування параметрів та ініціалізація об'єктів: Потім вказуються необхідні параметри, такі як порти для LoRa, облікові дані для підключення до WiFi, токен бота Telegram тощо. Об'єкти для роботи з WiFi і Telegram також ініціалізуються в цій частині коду.

Обробка нових повідомлень: Функція handleNewMessages(int numNewMessages) використовується для обробки вхідних повідомлень від Telegram. Якщо вхідне повідомлення має команду "/start", вона надсилає вітальне повідомлення. Якщо команда "/GET_INFO", надсилається інформація, що зберігається у змінній Incoming.

Отримання та обробка LoRa-пакетів: Функція onReceive(int packetSize) зчитує та обробляє вхідні LoRa-пакети.

Налаштування перед початком роботи: У функції setup() відбувається ініціалізація різних компонентів: відкривається послідовний порт, відбувається підключення до WiFi, ініціалізується LoRa тощо.

Основний цикл: У функції loop() на кожній ітерації відбувається читання і обробка LoRa-пакетів, а також перевірка на наявність нових повідомлень у Telegram. Якщо такі є, вони обробляються функцією handleNewMessages(). Цей цикл повторюється протягом усього часу роботи програми.

Загалом, цей код слугує для зв'язку між пристроями через LoRa та спілкування з користувачем через Telegram.

3.2.2 Код приймача

Підключаємо необхідні бібліотеки:

```
#include <SPI.h>
#include <LoRa.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
```

Задаємо порти для LoRa:

```
#define ss 5
#define rst 14
#define dio0 2
```

Вказуємо дані для підключення до WiFi:

```
const char* ssid = "Iphone ";
const char* password = "123456789";
```

Токен бота в Telegram:

```
#define BOTtoken "6238120988:AAER26SSJRT_bvE8nXv9A814E3DkhQiqrsc"
```

ID чату в Telegram:

```
#define CHAT_ID "308100665"
```

Змінні для обробки вхідних повідомлень:

```
String Incoming = "";
String Message = "";
```

Створюємо клієнта для підключення до серверів Telegram через SSL:

```
WiFiClientSecure client;
```

Ініціалізація бота з токеном і клієнтом:

```
UniversalTelegramBot bot(BOTtoken, client);
```

Вказуємо затримку між запитами до Telegram-сервера:

```
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;
```

Обробка нових повідомлень від Telegram:

```
void handleNewMessages(int numNewMessages) {
  Serial.println("handleNewMessages");
  Serial.println(String(numNewMessages));

  for (int i=0; i<numNewMessages; i++) {

    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID){
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }
    String text = bot.messages[i].text;
    Serial.println(text);

    if (text == "/start") {

      String welcome = "";
      welcome += "Привіт Сергій Малєєв.\n Цей бот використовує ESP32 для зв'язку
      з телеграмом та LoraWAN для передачі інформацію на делку відстань\n";
      welcome += "Для того щоб отримати інформацію введи команду /GET_INFO \n";
      bot.sendMessage(chat_id, welcome, "");
    }
    if (text == "/GET_INFO") {
      bot.sendMessage(chat_id, Incoming, "");
    }
  }
}
```

Функція приймає цілочисельний аргумент numNewMessages, який вказує наявність нових повідомлень, отриманих від Telegram-бота.

На початку функції handleNewMessages дві команди Serial.println використовуються для виведення повідомлень "handleNewMessages" і numNewMessages (являє собою довжину повідомлення).

Далі цикл for використовується для обробки кожного нового повідомлення.

У кожній ітерації циклу з повідомлення витягується chat_id (унікальний ідентифікатор чату, в якому було отримано повідомлення) і перевіряється на

збіг із глобально заданим CHAT_ID. Якщо вони не збігаються, бот надсилає повідомлення "Unauthorized user" на chat_id і пропускає решту обробки цього повідомлення (за допомогою команди continue).

Якщо chat_id збігається з CHAT_ID, функція витягує текст повідомлення та ім'я відправника з повідомлення.

Далі відбувається перевірка вмісту тексту повідомлення:

Якщо текст повідомлення дорівнює "/start", бот формує привітальне повідомлення і відправляє його назад у чат.

Якщо текст повідомлення дорівнює "/GET_INFO", бот відправляє вміст змінної Incoming назад у чат. Змінна Incoming містить інформацію, отриману від LoRa.

Ця функція дає змогу боту обробляти вхідні повідомлення і реагувати на команди, надіслані користувачем.

Адреси пристроїв LoRa (LocalAddress – приймач, Destination_Master – передавач):

```
byte LocalAddress = 0x02;
byte Destination_Master = 0x01;
```

Функція для обробки отриманих LoRa пакетів:

```
void onReceive(int packetSize) {

  if (packetSize == 0) return;
  byte sender = LoRa.read();
  byte incomingLength = LoRa.read();
  Incoming = "";
  while (LoRa.available()) {
    Incoming += (char)LoRa.read();
  }
  if (incomingLength != Incoming.length()) {
    Serial.println("error: message length does not match length");
    return;
  }
  Serial.println();
  Serial.println("Received from: 0x" + String(sender, HEX));
  Serial.println("Message length: " + String(incomingLength));
  Serial.println("Message: " + Incoming);
}
```

Функція onReceive() викликається при отриманні LoRa пакета. Як аргумент вона приймає packetSize, що представляє розмір отриманого пакета.

- if (packetSize == 0) return; - якщо розмір пакета дорівнює 0 (тобто пакета немає), функція припиняє виконання і виходить з функції.
- byte sender = LoRa.read(); - читання байта пакета, який представляє адресу відправника.
- byte incomingLength = LoRa.read(); - читання байта пакета, який представляє довжину повідомлення, що приходить.
- Incoming = ""; - скидання вмісту рядка Incoming, готуючи його до прийому нового повідомлення.
- while (LoRa.available()) { Incoming += (char)LoRa.read(); } - у циклі зчитуються байти пакета, що залишилися (тобто саме повідомлення), і додаються в рядок Incoming.
- if (incomingLength != Incoming.length()) { ... } - перевірка на відповідність зазначеної в пакеті довжини повідомлення фактичній довжині рядка Incoming.

Якщо довжини не збігаються, виводиться повідомлення про помилку, і функція припиняє виконання.

Наприкінці функції виводяться відомості про отриманий пакет: адреса відправника, довжина повідомлення і саме повідомлення.

Функція onReceive() обробляє вхідні LoRa пакети, перевіряючи їхню коректність і виводячи інформацію про них.

Налаштування перед початком роботи:

```
void setup() {

  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

    client.setCACert(TELEGRAM_CERTIFICATE_ROOT);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }

  Serial.println(WiFi.localIP());
  LoRa.setPins(ss, rst, dio0);
  Serial.println();
  Serial.println("Start LoRa init...");
  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa init failed. Check your connections.");
    while (true);
    }
  Serial.println("LoRa init succeeded.");
}
```

Ця функція викликається тільки один раз під час старту програми.

- Serial.begin(115200); - ініціалізує послідовне з'єднання на швидкості 115200 біт/с.
- WiFi.mode(WIFI_STA); - встановлює режим роботи модуля WiFi в режимі станції.
- WiFi.begin(ssid, password); - підключається до WiFi-мережі із зазначеними ім'ям (ssid) і паролем (password).

- client.setCACert(TELEGRAM_CERTIFICATE_ROOT); Це необхідно для встановлення безпечного з'єднання з Telegram.

У циклі while (WiFi.status() != WL_CONNECTED) { ... } перевіряється статус підключення WiFi. Якщо підключення не встановлено, очікування триває, поки з'єднання не буде встановлено. Кожну секунду в консоль виводиться повідомлення "Connecting to WiFi...".

- Serial.println(WiFi.localIP()); - друкує локальну IP-адресу пристрою в WiFi-мережі.
- LoRa.setPins(ss, rst, dio0); - задає контакти, до яких підключений модуль LoRa.
- Serial.println("Start LoRa init..."); - виводить у консоль повідомлення про початок ініціалізації LoRa.
- if (!LoRa.begin(433E6)) { ... } - ініціалізує модуль LoRa з частотою 433 МГц. Якщо ініціалізація не вдалася, виводиться повідомлення про помилку, і програма зупиняється.
- Serial.println("LoRa init succeeded."); - якщо ініціалізація LoRa пройшла успішно, виводиться відповідне повідомлення.

Отже, функція setup() виконує початкове налаштування пристрою, включно з підключенням до WiFi, налаштуванням безпечного зв’язку для роботи з Telegram та ініціалізацією модуля LoRa.

Основний цикл:

```
void loop() {
  onReceive(LoRa.parsePacket());

  if (millis() > lastTimeBotRan + botRequestDelay)  {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

    while(numNewMessages) {
      Serial.println("got response");
      onReceive(LoRa.parsePacket());
      handleNewMessages(numNewMessages);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }

    lastTimeBotRan = millis();
}
}
```

Функція loop() - основний цикл програми, який виконується нескінченно після виклику функції setup().

onReceive(LoRa.parsePacket()); - викликається функція onReceive, якій передається результат функції LoRa.parsePacket(). LoRa.parsePacket() - це функція з бібліотеки LoRa, яка перевіряє, чи є доступні для читання пакети даних. Якщо пакет доступний, функція повертає його розмір. Вхідний пакет потім обробляється функцією onReceive().

if (millis() > lastTimeBotRan + botRequestDelay) {…}- тут ми дивимося, чи минуло достатньо часу з моменту останнього виконання перевірки повідомлень бота (botRequestDelay). millis() - функція, що повертає кількість мілісекунд, які минули з моменту запуску програми.

Усередині умовного оператора if виконуються такі дії:

int numNewMessages = bot.getUpdates(bot.last_message_received + 1); - ми намагаємося отримати оновлення від бота. Функція bot.getUpdates() надсилає запит серверу Telegram на отримання нових повідомлень. Як аргумент функції передається ідентифікатор останнього обробленого повідомлення, збільшений на 1, щоб отримати всі нові повідомлення.

Потім, якщо є нові повідомлення (while(numNewMessages)), у циклі виконуються такі дії:

- Serial.println("got response"); - виводиться повідомлення "got response", що свідчить про те, що було отримано відповідь.
- onReceive(LoRa.parsePacket()); - знову викликається функція onReceive().
- handleNewMessages(numNewMessages); - викликається функція handleNewMessages, якій передається кількість нових повідомлень. Ця функція обробляє вхідні повідомлення.
- numNewMessages = bot.getUpdates(bot.last_message_received + 1); - знову виконується запит на отримання нових повідомлень.
- lastTimeBotRan = millis(); - після того, як усі нові повідомлення були оброблені, змінна lastTimeBotRan оновлюється поточним часом для подальшої перевірки в умові if.

Загалом, функція loop() відповідає за приймання даних від LoRa і за обробку повідомлень від бота Telegram

3.2.3 Передавач

Підключення бібліотек та ініціалізація параметрів: На початку коду підключаються всі необхідні бібліотеки: MQ135 (для роботи з датчиком якості повітря), DHT (для роботи з датчиком вологості та температури), SPI і LoRa (для роботи з LoRa модулем). Визначаються порти, до яких під'єднані модулі та датчики, а також адреси пристроїв у LoRa мережі.

Функція sendMessage: Ця функція приймає рядок і адресу призначення, після чого вона формує і надсилає LoRa пакет на вказану адресу. Пакет містить адресу одержувача, адресу відправника, довжину повідомлення і саме повідомлення.

Налаштування у функції setup: У цій функції відбувається ініціалізація послідовного порту і датчика DHT, а також налаштування і запуск модуля LoRa.

Цикл loop: У цьому циклі, з інтервалом у 3 секунди, проводяться вимірювання температури за допомогою датчика DHT і якості повітря за допомогою датчика MQ7. Потім ці дані форматуються в рядок і надсилаються через LoRa на приймач.

Цей код може бути використаний для моніторингу параметрів довкілля за допомогою датчиків DHT і MQ7 та передавання цих даних на віддалені пристрої через LoRa.

3.2.4 Код передавача

Підключаємо необхідні бібліотеки для роботи з датчиками та модулем LoRa:

```
#include "MQ135.h"
#include "DHT.h"
#include <SPI.h>
#include <LoRa.h>
```

Визначаємо макропідстановку для порту, до якого підключений датчик DHT:

```
#define DHTPIN 2
```

Визначаємо макропідстановки для портів LoRa:

```
#define ss 15
#define rst 16
#define dio0 4
```

Визначаємо макропідстановку для типу датчика DHT, а саме DHT11:

```
#define DHTTYPE DHT11
```

Ініціалізуємо об'єкт для роботи з датчиком DHT:

```
DHT dht(DHTPIN, DHTTYPE);
```

Рядок для повідомлення:

```
String Message = "";
```

Змінні для зберігання локальної адреси передавача та адреси приймача:

```
byte LocalAddress = 0x01;
byte Destination_ESP32_Slave_1 = 0x02;
```

Змінні для зберігання контролю часу надсилання повідомлень:

```
unsigned long previousMillis_SendMSG = 0;
const long interval_SendMSG = 3000;
```

Функція для надсилання повідомлення через LoRa:

```
void sendMessage(String Outgoing, byte Destination) {
  LoRa.beginPacket();
  LoRa.write(Destination);
  LoRa.write(LocalAddress);
  LoRa.write(Outgoing.length());
  LoRa.print(Outgoing);
  LoRa.endPacket();
}
```

Це функція для надсилання повідомлень через модуль LoRa. У неї передаються два параметри: рядок Outgoing, який потрібно надіслати, і адреса Destination, на яку потрібно надіслати.

- LoRa.beginPacket(); - Ця функція з бібліотеки LoRa створює пакет для передачі даних.
- LoRa.write(Destination); - Записує байт Destination у пакет як адресу одержувача.

- LoRa.write(LocalAddress); - Записує байт LocalAddress у пакет як адресу відправника.
- LoRa.write(Outgoing.length()); - Записує довжину вихідного повідомлення в пакет.
- LoRa.print(Outgoing); - Записує саме вихідне повідомлення в пакет.
- LoRa.endPacket(); - Ця функція завершує формування пакету і відправляє його. Після цього починається передача пакету через модуль LoRa.

Функція sendMessage() бере інформацію і адресу, упаковує їх у пакет даних LoRa і потім відправляє цей пакет.

Налаштування перед початком роботи:

```
void setup() {
  Serial.begin(115200);
  dht.begin();
  LoRa.setPins(ss, rst, dio0);
  Serial.println("Start LoRa init...");
  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa init failed. Check your connections.");
    while (true);
  }
  Serial.println("LoRa init succeeded.");
}
```

- Serial.begin(115200); - Цей код запускає серійний порт зі швидкістю передавання даних 115200 біт/с.
- dht.begin(); - Цей код ініціалізує сенсор температури та вологості DHT (DHT11), який під'єднаний до вашого ESP32.
- LoRa.setPins(ss, rst, dio0); - Цей код визначає контактні порти для LoRa модуля. Тут ss – інформаційний порт інтерфейсу SPI, rst - це номер порта для скидання і dio0 - це номер порта для переривання.
- Serial.println("Start LoRa init..."); - Цей код друкує рядок "Start LoRa init..." на серійний порт.

- if (!LoRa.begin(433E6)) { ... } - Цей код ініціалізує LoRa модуль із частотою 433 МГц. Якщо ініціалізація не вдалася, буде надруковано повідомлення про помилку, і ваша програма зациклиться в нескінченному циклі.
- Serial.println("LoRa init succeeded."); - Якщо ініціалізація LoRa модуля пройшла успішно, цей код надрукує "LoRa init succeeded." на серійний порт.

Загалом, функція setup() тут використовується для ініціалізації серійного порту, сенсора DHT і модуля LoRa перед початком основного циклу програми.

Основний цикл роботи програми:

```
void loop() {

unsigned long currentMillis_SendMSG = millis();

  if (currentMillis_SendMSG - previousMillis_SendMSG >= interval_SendMSG) {
  previousMillis_SendMSG = currentMillis_SendMSG;
  float t = dht.readTemperature();

  MQ135 gasSensor = MQ135(A0);
  float air_quality = gasSensor.getPPM();

  Message = "Temp : " + String(t) + "C" + " PPM : " +  String(air_quality);
  Serial.println("Lora sx1278 Transmit : ");
  Serial.print("Send message to ESP32 Receiver ");
  Serial.println();
  Serial.println("Message Data : " + Message);
  sendMessage(Message, Destination_ESP32_Slave_1);
 }
}
```

- unsigned long currentMillis_SendMSG = millis(); - Тут зберігається поточне значення часу в мілісекундах з моменту запуску програми.
- if (currentMillis_SendMSG - previousMillis_SendMSG >= interval_SendMSG) { ... } - Цей умовний вираз перевіряє, чи минуло достатньо часу з моменту останнього читання датчика. Якщо минуло, то виконується блок коду всередині фігурних дужок.

- float t = dht.readTemperature(); - Тут зчитується температура з датчика DHT.
- MQ135 gasSensor = MQ135(A0); - Тут створюється об'єкт gasSensor для датчика якості повітря MQ135, підключеного до аналогового піна A0.
- float air_quality = gasSensor.getPPM(); - Тут зчитується значення якості повітря з датчика MQ135 в одиницях PPM (часток на мільйон).
- Serial.print("Air Quality: "); ... - Тут поточне значення якості повітря друкується на серійний порт.
- Message = "Temp : " + String(t) + "C" + "PPM : " + String(air_quality); - Тут формується повідомлення з поточними значеннями температури та якості повітря.
- Serial.println(…); - Тут друкується інформація про повідомлення на серійний порт.
- sendMessage(Message, Destination_ESP32_Slave_1); - Відправляємо повідомлення на приймач.

Таким чином, у циклі loop() регулярно відбувається читання значень із датчиків, формування повідомлень та їхнє надсилання.

3.3 Результат роботи каналу зв’язку

На цьому (рис.3.4) зображені представлено виведення результатів виконання коду. Цей вивід складається з декількох ключових елементів, що містять адресу відправника, розмір повідомлення і текст повідомлення.

```
Received from: 0x1
Message length: 26
Message: Temp : 21.00C PPM : 327.74

Received from: 0x1
Message length: 26
Message: Temp : 21.00C PPM : 321.17

Received from: 0x1
Message length: 26
Message: Temp : 21.00C PPM : 314.70

Received from: 0x1
Message length: 26
Message: Temp : 21.00C PPM : 308.32

Received from: 0x1
Message length: 26
Message: Temp : 21.00C PPM : 308.32
```

Рис.3.4 Інформація з передавача

На наступному (рис.3.5) зображенні ми спостерігаємо саме повідомлення на приймаче.

```
Lora sx1278 Transmit :
Send message to ESP32 Receiver
Message Data : Temp : 21.00C PPM : 327.74

Lora sx1278 Transmit :
Send message to ESP32 Receiver
Message Data : Temp : 21.00C PPM : 321.17

Lora sx1278 Transmit :
Send message to ESP32 Receiver
Message Data : Temp : 21.00C PPM : 321.17

Lora sx1278 Transmit :
Send message to ESP32 Receiver
Message Data : Temp : 21.00C PPM : 314.70

Lora sx1278 Transmit :
Send message to ESP32 Receiver
Message Data : Temp : 21.00C PPM : 308.32
```

Рис.3.5 Інформація з приймача

3.4 Інтерфейс в telegram bot

На наступному (рис.3.6) зображенні ми бачимо результат роботи каналу зв’язку з візуалізацією в телеграм боті.

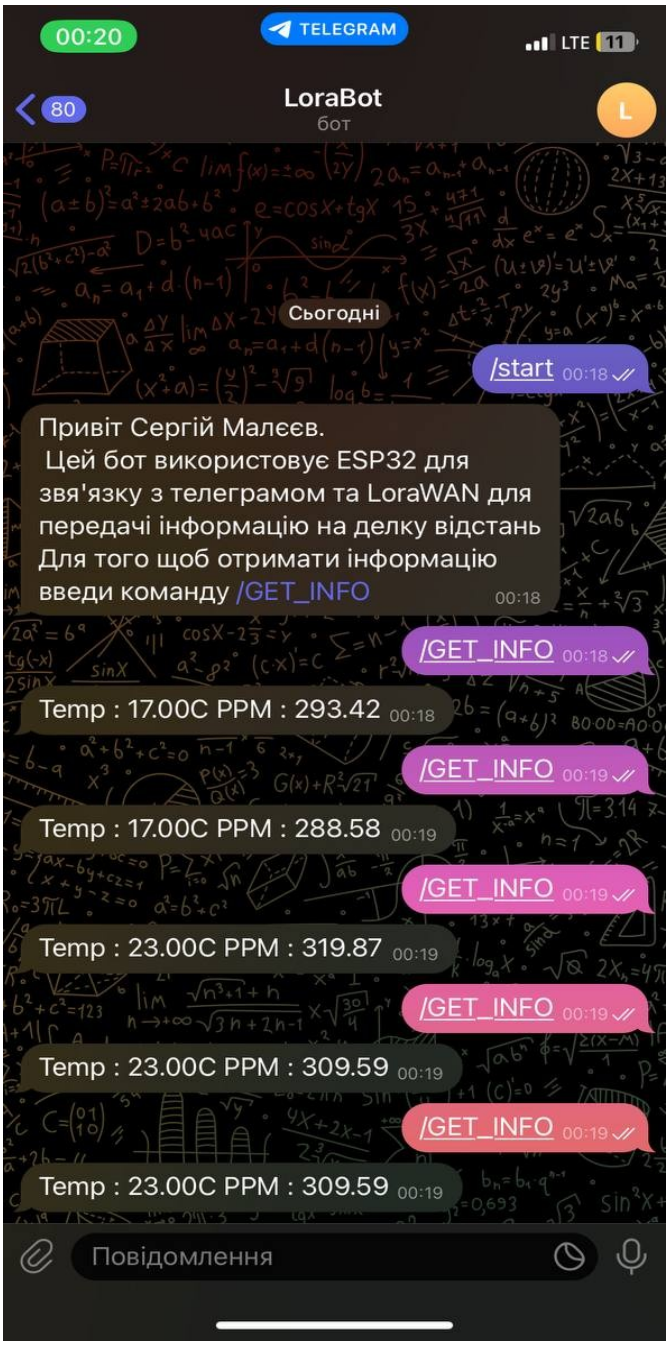

Рис. 3.6 Telegram Bot

ВИСНОВКИ

У результаті роботи було зроблено наступні висновки:

Модуль LoRa SX1278 краще використовувати з ESP 32 ніж Arduino , тому що у ESP 32 більше оперативної пам'яті і логіка початково в ESP 32 як і в LoRa SX1278 3.3 В. Також це зручніше з точки зору візуалізації інформації, тому що ми не обмежуємося LED дисплеєм, а маємо вихід в інтернет за допомогою вбудованого WiFi модуля в ESP 32.

Корисна інформація в ефірі значно подовжується. Для простого датчика, наприклад температури, корисна інформація становить 6 байт, але в ефірі передається щонайменше 24 байти. Це відбувається через те, що до корисної інформації додаються службові дані, як-от преамбула, CRC, інформація про адресу тощо.

Усі ці службові дані необхідні для забезпечення надійної та правильної передачі інформації. Вони допомагають забезпечити синхронізацію між передавачем і приймачем, підтверджують, що дані були передані без помилок, і вказують, куди саме повинні були бути відправлені дані.

Збільшення швидкості у 2 рази - у 2 рази зменшує площу покриття шлюзу.

Гранична площа покриття радіосистеми прямо пропорційна енергії біта інформації і обернено пропорційна загасанню сигналу і шумам у каналі.

Дальність роботи LPWAN систем критично залежить від висоти підйому антени шлюзу, і це не тому, що земля закруглюється. У місті найчастіше сигнал приходить на приймач відбившись від стін найближчих будинків. Піднімаючи приймач, Ви збільшуєте ефективну площу відбитого сигналу, яку вона бачить. Збільшення висоти підйому антени у 2 рази - у 2 рази збільшує площу покриття шлюзу.

Переваги коротких повідомлень:

Можливе збільшення дальності за рахунок зменшення швидкості передачі: Якщо передане повідомлення коротке, його можна передати на більшу відстань, зменшивши швидкість передачі. Це допомагає збільшити радіус дії мережі.

Більший термін служби батарейки: Короткі повідомлення вимагають меншої кількості енергії для передавання, що збільшує термін служби батарейок у пристроях LPWAN.

Більша пропускна здатність системи: Під час передавання коротких повідомлень можливо обробити більше повідомлень за той самий проміжок часу, збільшуючи тим самим пропускну спроможність системи.

Більша завадозахищеність повідомлення: Короткі повідомлення можуть бути більш стійкими до перешкод, оскільки ймовірність помилок під час передачі даних зменшується. Плюс, завадозахищене кодування (як згадувалося в попередньому вашому запитанні) додатково збільшує стійкість до шумів і перешкод.

СПИСОК ДЖЕРЕЛ

[1] LoRaWAN for advanced users. UNIVERSITÉ SAVOIE MONT BLANC. Sylvain MONTAGNY Посилання: https://www.univ-smb.fr/lorawan/wp-content/uploads/2022/04/Book-LoRaWAN-Advanced.pdf

[2] LoRa® Products. Long Range, Low Power ConsumptionSecure Device to Cloud Solutions Посилання: https://www.semtech.com/lora/what-is-lora

[3] Особливості технології LoRaWAN 27 Квітня 2020. вул. Маричанська, 18, м. Київ, 03040, Україна Посилання: https://iotji.io/osoblyvosti-lorawan/

[4] Semtech ProductsShort Form Catalog. 200 Flynn Road, Camarillo, California 9301.Посилання: https://lora-alliance.org/about-lorawan/

[5] DATA SHEET SX1276/77/78/79. WIRELESS, SENSING & TIMING. 2016 Semtech Corporation. Rev. 5 - August 2016. Посилання: https://1wire.com.ua/download/sx1278.pdf

[6] NB-IoT Deployment Guide to Basic Feature set Requirements. June 2019 Посилання: 201906-GSMA-NB-IoT-Deployment-Guide-v3.pdf

[7] NB-IoT Application Development Guide. Technology architecture and AT command examplesApplication Note.UBX-16017368-R05 Посилання: NB-IoT Application Development Guide (u-blox.com)

[8] NB-IoT explained: a complete guide to Narrowband-IoT. Посилання: https://www.i-scoop.eu/internet-of-things-iot/lpwan/nb-iot-narrowband-iot/

[9] Sigfox support. Sigfox documentation. 2023 © Sigfox. Edit multiple devices simultaneously. Посилання: https://support.sigfox.com/docs/edit-multiple-devices-simultaneously

[10] Copyright ©2017 mesh-net.co.uk. All Rights Reserved. Посилання: https://www.mesh-net.co.uk/project/sigfox/

[11] Sigfox support. Sigfox documentation. 2023 © Sigfox. Backend user account creation. Посилання: https://support.sigfox.com/docs/backend-user-account-creation
[12] Sigfox support. Sigfox documentation. 2023 © Sigfox. UnaConnect. Посилання: https://support.sigfox.com/docs/unaconnect
[13] Digi XBee® 3 Cellular LTE-M/NB-IoT Global Smart Modem User Guide. Посилання: https://www.digi.com/resources/documentation/digidocs/PDFs/90002258.pdf
[14] Архітектура мережі LTE-M. Посилання: https://ppt-online.org/196207
[15] NB-IoT vs LTE-M: A comparison of the two IoT technologies. Michael Bosson Content Specialist at Onomondo 10.05.2023. Посилання: https://onomondo.com/blog/nb-iot-vs-lte-m-a-comparison-of-the-two-iot-technology-standards/
[16] Розуміння технології NB-IoT проти LTE-M. Copyright © 2023 DusunIoT. Посилання: https://www.dusuniot.com/uk/blog/nb-iot-vs-lte-m-technology/
[17] Mini-Tech. LoRa SX 1278 Посилання: https://www.mini-tech.com.ua/image/cache/catalog/communication/Lora_Ra-02_1-550x550.jpg
[18] ESP32 Series Datasheet. Посилання: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
[19] Модуль WiFi ESP-32s плата ESP-WROOM-32. 2004—2022 (c) ЧП «ВОРОН». Посилання: https://voron.ua/files/pic/Modules/WIFi/039384_4-b.jpg
[20] TECHNICAL DATA MQ-7 GAS SENSOR. TEL:86-371-67169070 67169080 FAX:86-371-67169. HANWEI ELECTRONICS CO Посилання: https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf
[21] Mini-Tech. MQ-7 Посилання: https://www.mini-tech.com.ua/image/cache/catalog/sensors/mq7_module-550x550.jpg

[22] DHT11–Temperature and Humidity Sensor. Посилання: DHT11 Sensor Pinout, Features, Equivalents & Datasheet (components101.com)

[23] DHT11-UNIVERSAL-SOLDER Electronics Ltd. Copyright © 2023 honestwin, All Rights Reserved Посилання: https://www.honestwin.net/search/?keyword=dht11&gclid=CjwKCAjwpayjBhAnEiwA-7ena5ksXO2GOcpZl9sjIxArRLzFzafKhhgznrapLOTsPk_Pxvu3fK-xqxoCWx8QAvD_BwE