

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення автоматизованих систем

Пояснювальна записка

до магістерської роботи
на ступінь вищої освіти магістр

на тему: **«РОЗРОБКА МОДЕЛІ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТРЕКІНГУ
ОБ'ЄКТІВ З ВИКОРИСТАННЯМ ІОТ-ТЕХНОЛОГІЙ»**

Виконав: студент 6 курсу, групи ІСДМ-61
спеціальності 126 Інформаційні системи та технології
освітня програма «Інформаційні системи та технології»
(шифр і назва спеціальності)

_____ **Колодяженський Б.М.**

(прізвище та ініціали)

Керівник _____ **Тушич А.М.**

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____ **Чорна В.М**

(прізвище та ініціали)

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти - «Магістр»

Спеціальність підготовки 126 Інформаційні системи та технології

Освітня програма «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

К.П.Сторчак

“ _____ ” _____ 2021 року

З А В Д А Н Н Я НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Колодяженський Богдан Михайлович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка моделі інформаційної системи трекінгу об'єктів з використанням IoT-технологій»

Керівник роботи: Тушич Аліна Миколаївна, доктор філософії

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від _____ року № _____

2. Строк подання студентом роботи _____

3. Вхідні дані до роботи :

Науково-технічна література щодо питань топологій мереж та технологій бездротової передачі даних

Технічна література щодо роботи Arduino Uno та модулів-датчиків, що сумісні з мікроконтролером

Науково-технічна література щодо архітектури інформаційних систем

Науково-технічна література щодо технологій визначення геолокації

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1 Цілі та перспективи використання систем для трекінгу об'єктів.

4.2 Архітектури інформаційних систем.

4.3 Технології для бездротової передачі даних між компонентами системи.

4.4 Дослідження технологій для відстеження геолокації об'єктів.

4.5 Дослідження технологій для серверної частини моделі.

4.6 Компоненти необхідні для роботи системи трекінгу об'єктів.

4.5 Розробка моделі інформаційної системи трекінгу об'єктів.

5. Перелік графічного матеріалу

1. Аналіз архітектур та топологій побудови IoT

2. Аналіз технологій для бездротової передачі даних

-
3. Аналіз технологій для визначення геолокції об'єкта

 4. Розробка моделі інформаційної систем трекінгу об'єктів

 5. Вибір апаратних та програмних засобів для створення системи

 6. Створення робочого макету GPS

 7. Створення серверу для моделі трекінгу об'єктів

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури		
2	Вивчення матеріалів для подальшої взаємодії з ними		
3	Огляд протоколів передачі даних		
4	Визначення технічного завдання		
5	Розробка сценарію взаємодії		
6	Розробка моделі		
7	Вступ, висновки, реферат		
8	Розробка демонстраційних матеріалів		
9	Попередній захист роботи		

Студент Колодяженський Б.М.
(підпис) (прізвище та ініціали)

Керівник роботи Тушич А.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи 77 с., 37 рис., 21 джерело.

ІОТ, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, ІНФОРМАЦІЙНА СИСТЕМА, ТРЕКІНГ ОБ'ЄКТІВ, МОДЕЛЬ

Об'єкт дослідження: процес створення моделі інформаційної системи трекінгу об'єктів з використанням ІоТ-технологій.

Предмет дослідження: складові та компоненти моделі інформаційної системи трекінгу об'єктів.

Мета роботи: аналіз моделі інформаційної системи трекінгу об'єктів, її компонентів, а також проектування та розробка такої системи на основі проаналізованої інформації.

Методи дослідження: аналітичні методи, методи бездротової передачі даних, методи отримання геолокації об'єктів.

У роботі було визначено основні параметри моделі інформаційної системи трекінгу об'єктів. Проведено аналіз можливих архітектур таких систем, визначено їх переваги та недоліки. Було досліджено методи та технології для відстеження та отримання геолокації об'єктів. Проаналізовано технології, які дозволяють створити серверну частину моделі, а саме мови програмування та бази даних. Визначено необхідні компоненти для побудови моделі інформаційної системи трекінгу об'єктів та надана їх коротка характеристика.

На основі результатів виконаних досліджень було розроблено модель інформаційної системи трекінгу об'єктів, створено трекер з можливістю отримання його геопозиції та серверна частина моделі, яка дозволяє зберігати та віддавати по запиту координати трекеру.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	прикладний програмний інтерфейс
IoT	Internet of Things
RESTful	передача репрезентативного стану
REST	Representational State Transfer
HTTP	HyperText Transfer Protocol
GPS	Global Positioning System
LBS	Location-based service

АНОТАЦІЯ

«Інтернет речей» відноситься до глобальної розподіленої мережі фізичних об'єктів, які здатні, взаємодіяти зі своїм середовищем, один з одним, іншими машинами чи комп'ютерами. Такі «розумні» об'єкти мають широкий діапазон розмірів і можливостей – від побутової техніки, промислових роботів, автомобілів, поїздів до предметів, які присутні в повсякденному житті будь-якої людини.

Інтернет речей (ІоТ) в логістиці набув широкого поширення в останнє десятиліття, в основному у вигляді пристроїв відстеження. Технологія – це інструмент, який забезпечує плавний і безперебійний процес доставки. Дозволяє відстежувати ті параметри в роботі обладнання, які недоступні для людини або вимагають багато уваги і часу.

Одним з основних засобів логістики є моніторинг автопарку та вантажів у всіх дрібницях, що перевозяться. Щоб задовольнити це, на підприємстві необхідно впровадити систему відстеження об'єктів.

Впровадження технологій відстеження об'єктів дозволяє:

- керувати інтелектуальною транспортною інфраструктурою, отримувати доступ до даних про стан доріг та уникати перевантажень за допомогою аналітики трафіку (як у реальному часі, так і збережених даних).
- спростувати логістику, використовуючи дані та сповіщення в режимі реального часу, щоб оптимізувати маршрути доставки, відстежувати продуктивність і реагувати на затримки або проблеми в міру їх виникнення.
- забезпечувати безпеку, ефективність та надійність вашого ланцюга поставок.
- відстежувати розумні транспортні засоби, вантажі (наприклад, піддони, посилки та візки) та відправляти під час їх руху.

Така система дозволяє не тільки покращити транспортну інфраструктуру, а й створити всі необхідні передумови для її зростання.

На основі даних, зібраних такою системою, компанії можуть аналізувати загальну продуктивність і вживати маневрів для підвищення безпеки та продуктивності.

Отримання даних про місцезнаходження здійснюється різними способами, але найпопулярнішим є GPS. Дані про місцезнаходження спрощують процес відстеження маршруту доставки вантажу. Використання таких даних покращить логістичні операції. Дані допоможуть визначити найефективніших кур'єрів, водіїв вантажівок, визначити найефективніші маршрути доставки тощо.

ЗМІСТ

1 АНАЛІЗ СФЕР ВИКОРИСТАННЯ ТА АРХІТЕКТУР ІНФОРМАЦІЙНОЇ СИСТЕМ ТРЕКІНГУ ОБ’ЄКТІВ	15
1.1 Використання інформаційної систем трекінгу об’єктів в сфері логістики	15
1.2 Аналіз архітектур побудови інформаційної систем трекінгу об’єктів.....	19
1.2.1 Централізована архітектура системи	21
1.2.2 Децентралізовані архітектура системи.....	24
1.2.3 Способи взаємодії між IoT пристроями	27
2 АНАЛІЗ ТЕХНОЛОГІЇ ДЛЯ ПЕРЕДАЧІ ДАНИХ МІЖ ПРИСТРОЯМИ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ТРЕКІНГУ ОБ’ЄКТІВ.....	28
2.1 Використання технології Bluetooth для передачі даних.....	28
2.2 Використання технології Zigbee для передачі даних	31
2.3 Використання технології LPWAN для передачі дани	34
2.4 Використання технології Z-Wave для передачі даних.....	37
2.5 Використання технології Wi-Fi для передачі даних	40
3 СПОСОБИ ОТРИМАННЯ ГЕОПОЗИЦІЇ ОБ’ЄКТІВ.....	45
3.1 Використання технології GPS для визначення геолокації	47
3.2 Використання технології Wi-Fi для визначення геолокації	50
3.3 Використання технології LBS для визначення геолокації	52
4 ВИБІР АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПОБУДОВИ ІНФОРМАЦІЙНОЇ СИСТЕМ ТРЕКІНГУ ОБ’ЄКТІВ	56
4.1 Вибір апаратних засобів для побудови інформаційної систем.....	56
4.2 Вибір програмних засобів для побудови інформаційної систем.....	62
4.2.1 Вибір програмних засобів для серверної частина системи	62
4.2.2 Вибір програмних засобів для програмування трекеру.....	70
5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ТРЕКІНГУ ОБ’ЄКТІВ	73
5.1 Схема роботи інформаційної систем трекінгу об’єктів.....	73

	12
5.2 Розробка макету GPS-трекера.....	75
5.3 Розробка серверу для моделі.....	78
ВИСНОВКИ	86
ПЕРЕЛІК ПОСИЛАНЬ.....	87
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	89

ВСТУП

Інтернет речей знаходить застосування у всіх сферах, де задіяний транспорт - доставка вантажів, виробництво, роздрібна торгівля, сільське господарство, будівництво та багато інших. Технологія забезпечує плавність, прозорість та безперервність процесу доставки.

Одне з основних завдань логістики – моніторинг автопарку та товарів у всій тривалості перевезення.

Як правило, робота логістичних центрів будується за такою схемою:

Моніторинг та збирання даних у режимі реального часу (безпосередньо на місці відстеження – автомобіль або стаціонарний об'єкт).

Індикатори станів та звіти (робоче місце диспетчера у програмі).

Відстеження транспорту – можливість моніторингу товарів по всьому ланцюжку поставок підвищить безпеку. Використання сучасних GPS-стеження за автотранспортними засобами та RFID-датчиків можна використовувати для збору даних про стан, місцезнаходження активів, температуру та інші форми даних про конкретні елементи. Це допоможе менеджерам ланцюга постачання впровадити більш ефективний контроль якості, забезпечити своєчасне постачання, мінімізувати крадіжки та дати загальне уявлення про функціонування ланцюга постачання, щоб допомогти у прийнятті рішень.

Управління парком. Сучасні рішення для транспортної телематики та датчики GPS-позиціонування можуть використовуватися для збору даних у режимі реального часу для моніторингу та аналізу характеристик автомобіля та поведінки водія, а також для відстеження транспортних засобів та навантаження. У мобільному додатку

менеджер транспорту може вказати, де певне місце розташування, швидкість та напрямок - керувати водіями від несанкціонованих дій та некоректної поведінки на дорозі та за її межами. IoT також дає менеджеру чітку видимість всього процесу поставки.

1 АНАЛІЗ СФЕР ВИКОРИСТАННЯ ТА АРХІТЕКТУР ІНФОРМАЦІЙНОЇ СИСТЕМ ТРЕКІНГУ ОБ'ЄКТІВ

1.1 Використання інформаційної систем трекінгу об'єктів в сфері логістики

Сфера логістики є пріоритетною для запровадження Інтернету речей. На даний момент фінансування розвитку інтелектуальних транспортних технологій в автомобілях, дозволяє зменшити кількість аварій, збільшити екологічні показники. Крім того, використання Іот-технологій підвищує ефективність контролю трафіку що для бізнесу являється одним з найголовніших факторів.

Для управління трафіком руху застосовуються дані з вулиць, звітів користувачів транспортних засобів, камер зовнішнього спостереження, детекторів дорожнього руху та пристрої для відстеження геолокації. Виходячи з цього система налаштовує оптимальний розклад роботи транспорту, внаслідок чого підвищується якість послуг.

Впровадження інтернету речей дозволяє направляти транспортний потік, планувати місця для паркування та інше. Транспорт, який підключений до Інтернету та віддалений моніторинг автомобільного парку допомагають скоротити витрати завдяки оптимізації ремонту та обслуговування техніки.

Дані ІоТ можуть використовуватися для оптимізації планування маршрутів та їх перепланування, управління витратами палива та скорочення викидів для збільшення показників зеленої діяльності компанії. Інші сфери застосування – виявлення несправностей для планування технічного обслуговування, підвищення безпеки дорожнього руху та забезпечення дотримання законодавства.

Управління запасами та прогнозування. Датчики IoT можуть використовуватися для відстеження запасів та надання даних, які допоможуть в аналізі тенденцій прогнозування майбутніх потреб у запасах. Це допоможе уникнути ситуацій із недостатнім запасом та надлишковим запасом.

У галузі логістики IoT може створити інтелектуальну систему керування місцезнаходженням, яка дозволить компаніям легко відстежувати дії водія, місцезнаходження транспортного засобу та статус доставки.

Таке рішення є незамінним помічником у плануванні поставок, складанні та перегляді розкладів. Усі зміни миттєво виявляються і відбиваються у часі. Таким чином, технологія IoT може бути успішно використана для покращення керування розташуванням та оптимізації бізнес-процесів.

Таблиця 1 – Области застосування IoT у транспортуванні та зберіганні вантажів

Області застосування	Плюси
Підключений до інтернету транспорт	Прозорість усієї протяжності ланцюжка поставок Контроль режиму руху Підвищення дисципліни
Технології для логістики на основі інтернету речей	Супутникова геолокація «Хмарні» обчислення Переміщення трасою на автопілоті Повністю автономні системи

Таблиця 1 – Области застосування IoT у транспортуванні та зберіганні вантажів

Области застосування	Плюси
Забезпечення безпеки	<p>Аутентифікація «свій-чужий»</p> <p>Попередження неправильної експлуатації транспорту</p> <p>Легкий доступ до даних для розслідування подій</p> <p>Управління парком транспортних коштів</p>
Управління парком транспортних коштів	<p>Оптимізація сервісу в залежності від умов експлуатації</p> <p>Механічний оперативний контроль</p> <p>Злиття із системами ERP</p>
Автоматизація складів	<p>Стелажі з вантажами, що перевозяться роботами</p> <p>Автоматизовані системи складів</p> <p>Автонавантажувачі без водіїв</p> <p>Відстеження цілісності трубопроводу та його елементів</p> <p>Моніторинг навантаження на мости і тунелі, а також ступінь їх зносу</p>
Моніторинг активів	<p>Відстеження переміщення та дислокації вантажів</p> <p>Миттєвий опис вантажу в тарі, складу та і т.д.</p> <p>Доступ дислокації вантажу по всьому маршруту</p>
Спостереження за активами	<p>Моніторинг основних показників</p> <p>Миттєвий відгук у випадках зміни стану</p> <p>Прозорість даних під час перевезень</p>

Вантажівки – це рятувальний круг будь-якої логістичної компанії. Тільки в США понад 70% всіх вантажів перевозяться вантажівками. Фактично, близько 95% всіх

вироблених товарів в один момент транспортуються вантажівками. Тому компанії з логістики та автопарку потребують систем, які можуть допомогти їм керувати операціями вантажівок.

Рішення для керування розташуванням та маршрутами IoT для логістичної галузі є досить популярним. Це рішення дозволяє менеджеру з логістики відстежувати місцезнаходження товару в режимі реального часу. Використовуючи системи відстеження GPS та методи геозони можна також контролювати тоставку вантажів з віддалених місць. Це також допомагає логістичним компаніям відстежувати діяльність водіїв і забезпечувати своєчасну доставку вантажу.

Крім того, система оповіщення в реальному часі для відстеження транспортних засобів сигналізує менеджерам про будь-які аномалії, такі як гроза або аварія на автостраді, за допомогою push-повідомлень, які можуть вплинути на статус відправлення.

Ці функції діють являються помічниками для логістичних компаній і допомагають планувати та керувати графіками доставки. Перешкоди, миттєво визначаються та пом'якшуються, що призводить до оптимізації бізнес-процесів і задоволеності клієнтів.

Розумні контейнери, як і будь-які інші «розумні речі», оснащені підключеними до Інтернету гаджетами для визначення навколишнього середовища. Ці пристрої дозволяють їм керувати різними аспектами транспортування без будь-якої взаємодії з людиною.

Датчики відстеження контейнерів все ще є однією з найбільш широко використовуваних технологій IoT експедиторами та судноплавними лініями. Завдяки розумним трекарам можна перевірити місцезнаходження контейнера в будь-який момент і отримувати оновлення про статус відправлення в режимі реального часу.

Наприклад, розумні контейнери можуть стежити за температурою або освітленням і повідомляти про зміни. Також автоматично налаштуйте його на задані рівні, якщо це необхідно.

Вони можуть записувати стан відкритих/закритих дверей з точним часом. А також відстежувати будь-які інші незвичайні дії.

Ви можете відстежувати весь контейнер, а також відстежувати відправлення окремого предмета, поки він не буде доставлений до пункту призначення.

1.2 Аналіз архітектур побудови інформаційної систем трекінгу об'єктів

Типова архітектура IoT-систем складається з наступних 3-х рівнів:

Кінцеві пристрої (Things) – датчики, сенсори, контролери та інше периферійне обладнання для вимірювання необхідних показників та передачі цих даних у мережу за проводовими або бездротовими протоколами (WiFi , Bluetooth, LPWAN, Zigbee, Z-Wave та ін.). Оскільки кожна «порція» цієї інформації є невеликою за обсягом, такі дані називають малими (Little Data).

Мережеві шлюзи та хаби (Network) – роутери, які об'єднують та підключають кінцеві пристрої до хмари, він може бути вбудований в кінцевий пристрій.

Хмара (Cloud) – віддалений сервер у датацентрі, що обробляє, аналізує та надійно зберігає інформацію. Саме тут малі дані перетворюються на Big Data, коли консолідується безліч інформаційних потоків з різних пристроїв. Так інтернет речей стає «інтелектуальним», оскільки підключаються засоби аналізу даних, зокрема. із

використанням методів машинного навчання. Це дозволяє ефективно та віддалено керувати технікою, на якій встановлені кінцеві пристрої.

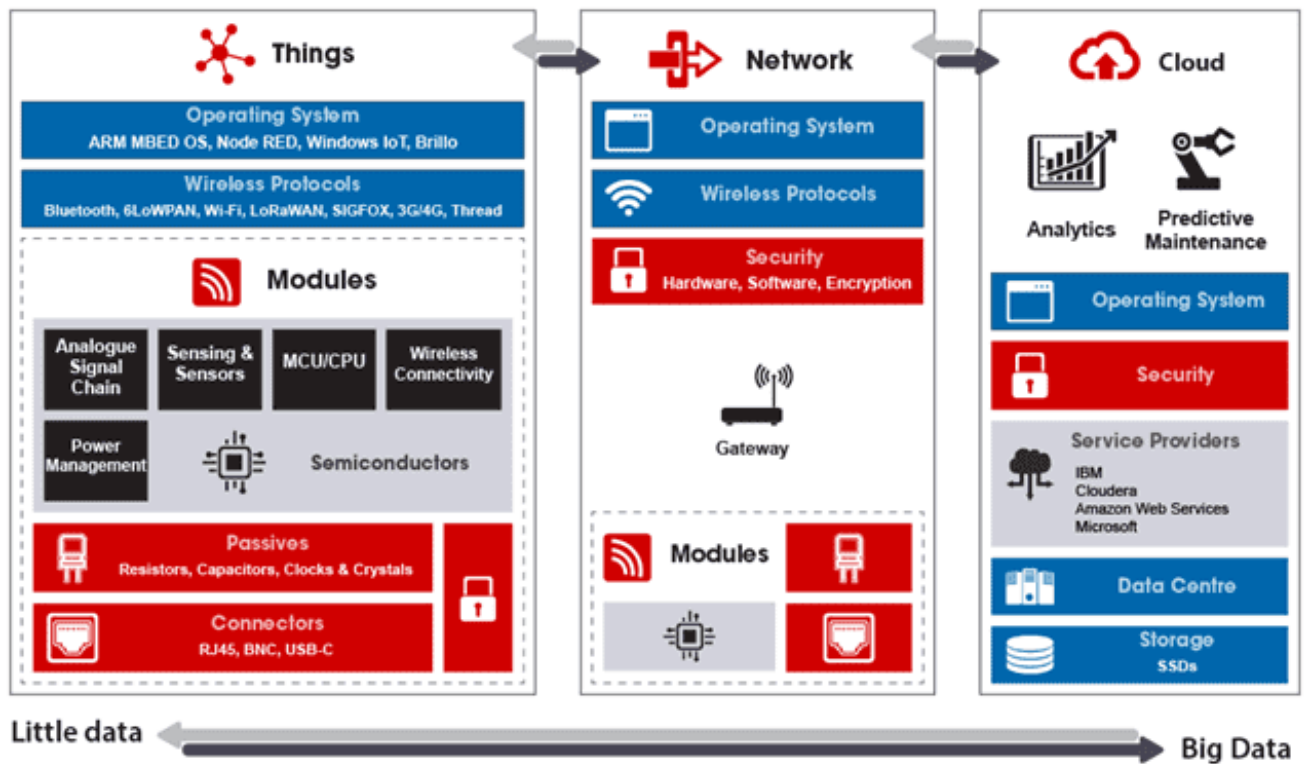


Рисунок 1.1 – Складові IoT-систем

Для організації автоматизованої системи трекінгу об'єктів, необхідно визначитись з загальною архітектурою системи на рівні пристроїв, вона може бути централізованою та децентралізованою.

Децентралізована, заснована на тому, що всі вузли в системі рівноправні, кожен елемент яких отримує сигнал для обробки, має інтелектуальні функції.

Централізована, передбачає підключення модулів до центрального контролера, в такому випадку кожний сигнал з модуля буде направлятися до контролера.

1.2.1 Централізована архітектура системи

Суть схеми роботи централізованої системи управління будинком у цьому, що має місце програмування одного центрального елемента (логічного модуля). Найчастіше це контролер із багатьма виходами. Для цього контролера пишеться програма, за допомогою якої здійснюється управління виконавчими пристроями та системами. Ця схема дає можливість використовувати мультизадачні сценарії з використанням широкого вибору обладнання. Централізовані системи автоматизації можуть бути як дротові, і бездротові.

Переваги централізованої системи автоматизації:

- Єдиний інтерфейс для управління всіма системами,
- Можливість створення мультизадачних сценаріїв,
- Підключення до системи практично будь-якого обладнання.

Слід врахувати деякі особливості при встановленні централізованих систем:

- Вразливість. Уся система управління залежить від центрального модуля. Модуль виходить з ладу – завмирає вся система.
- Людський фактор. Якщо втрачено зв'язок із програмістом, який спочатку писав код для вашої системи, то внести зміни в систему іншій людині буде дуже важко. Швидше за все, доведеться писати програму заново.

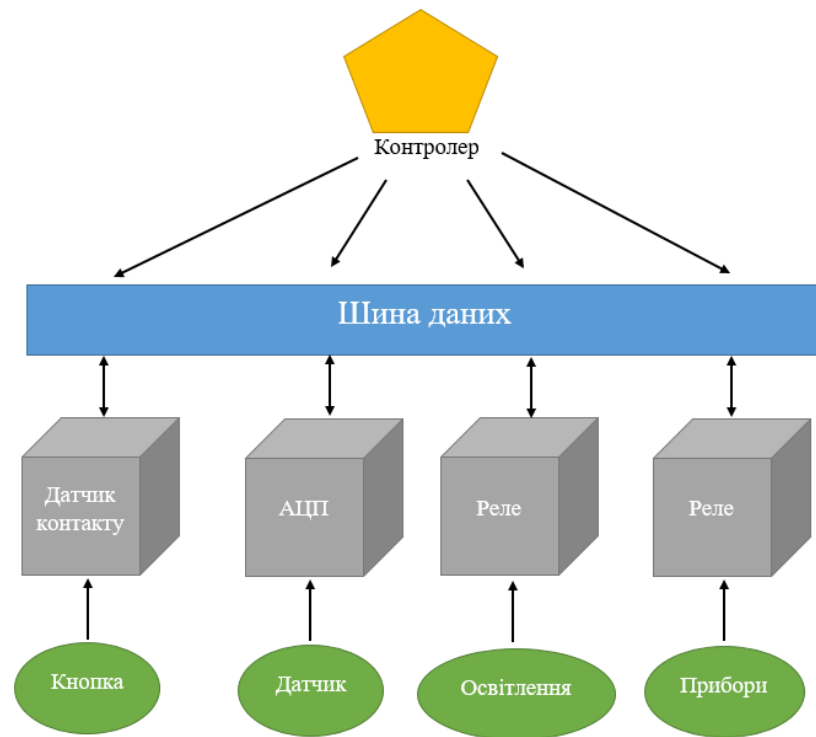


Рисунок 1.2 – Структура централізованої автоматизованої системи управління

Централізована система може бути побудована на основі топологій «Зірка», та «Загальна шина».

Топологій «Зірка» - топологія в якій усі хости підключені до центрального пристрою, відомого як пристрій-концентратор, за допомогою з'єднання «точка-точка». Тобто між хостами і концентратором існує з'єднання «точка-точка».

Концентратор діє як єдина точка відмови. Будь-який зв'язок між хостами відбувається тільки через концентратор. Ця топологія не є дорогою, оскільки для підключення ще одного хоста потрібен лише один кабель, а конфігурація проста.

Переваги:

- вихід з ладу однієї робочої станції не відбивається на роботі всієї мережі загалом;
- гарна масштабованість мережі;
- легкий пошук несправностей та обривів у мережі;

- висока продуктивність мережі (за умови правильного проектування);
- гнучкі можливості адміністрування.

Недоліки:

- вихід з ладу центрального концентратора обернеться непрацездатністю мережі (або сегмента мережі) загалом;
- для прокладки мережі часто потрібно більше кабелю, ніж більшість інших топологій;
- кінцева кількість робочих станцій у мережі (або сегменті мережі) обмежена кількістю портів у центральному концентраторі.

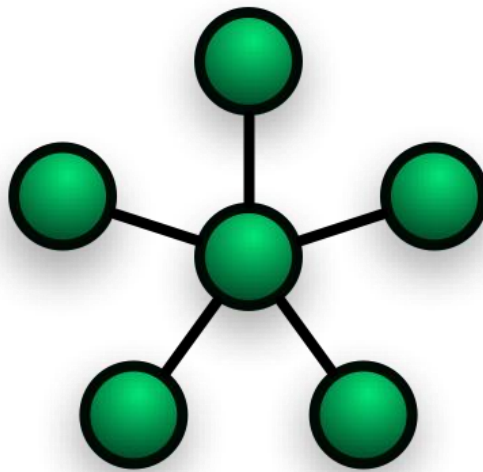


Рисунок 1.3 – Схема з'єднання вузлів в топології виду «Зірка»

Топологій «Загальна шина» - топологія являє собою основний кабель, і всі пристрої підключені до цього основного кабелю за допомогою відводних ліній. Оскільки всі дані передаються по основному кабелю, існує обмеження на кількість ліній і відстань, яку може мати основний кабель.

Переваги:

- Невеликий час встановлення мережі;
- Дешевизна (потрібно менше кабелю та мережевих пристроїв);

- Простота налаштування;
- Вихід з ладу робочої станції не відбивається на роботі мережі.

Недоліки:

- Проблеми в мережі, такі як обрив кабелю і вихід з ладу термінатора, повністю блокують роботу всієї мережі;
- Складна локалізація несправностей;
- З додаванням нових робочих станцій знижується продуктивність мережі.

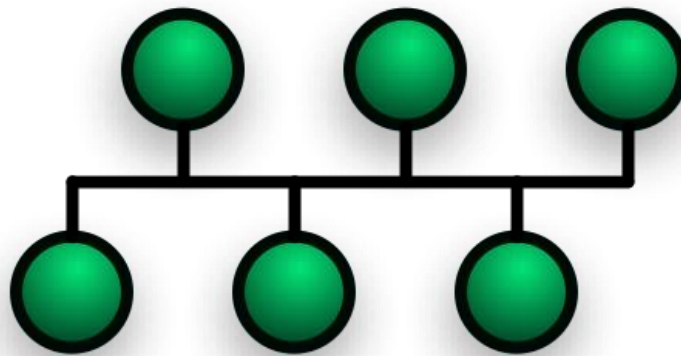


Рисунок 1.4 – Схема з'єднання вузлів в топології виду «Загальна шина»

1.2.2 Децентралізовані архітектура системи

У децентралізованій системі кожен виконавчий пристрій у будинку містить у собі мікропроцесор, який у свою чергу містить у собі енергонезалежну пам'ять. У цьому полягає досить висока надійність даних систем.

Відповідно при поломці одного пристрою вся система працює коректно, звичайно, за винятком тих приладів, які були підключені до пристрою, що вийшов з ладу.

Переваги децентралізованих систем:

1. Через відсутність центрального контролера, вихід з ладу одного або декількох модулів, істотно не вплине на роботу системи в цілому. З чого випливає, децентралізована система володіють підвищеною надійністю.
2. Децентралізовані системи прості в розширенні. На наявну шину легко додаються нові модулі, що підтримують протокол передачі даних який використовує шина.

Недоліки децентралізованих систем:

1. Висока вартість подібних систем, щодо централізованих.
2. Досить низька швидкість роботи децентралізованих систем, в наслідок проведення процедури обробки даних в різних модулях.
3. Виникнення технічних складнощів і збільшення розміру системи в силу того, що модулі володіють власними контролерами обробки даних.

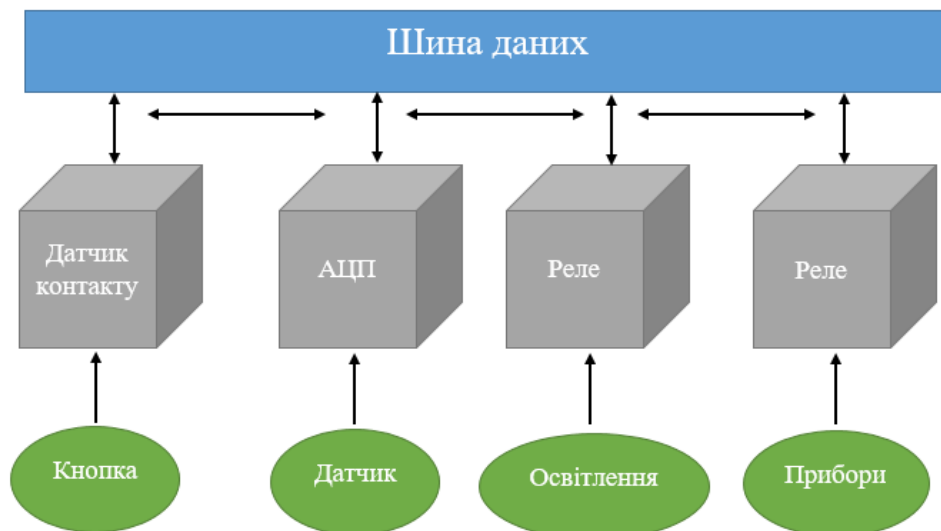


Рисунок 1.5 – Структура децентралізованих автоматизованої системи управління

Децентралізована система може бути побудована на основі «Повнозв'язкова топологія», та «Сітчастої топології».

«Повнозв'язкова топологія» - топологія комп'ютерної мережі, в якій кожна робоча станція підключена до всіх інших. Цей варіант є громіздким та неефективним,

незважаючи на свою логічну простоту. Для кожної пари має бути виділена незалежна лінія, кожен комп'ютер повинен мати стільки комунікаційних портів скільки комп'ютерів у мережі.

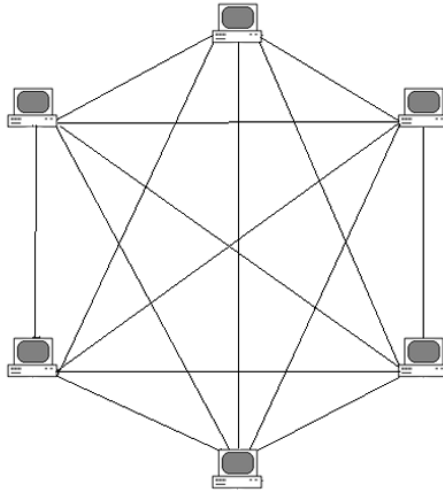


Рисунок 1.6 – Схема з'єднання вузлів в повнозв'язній топології

«Сітчаста топологія» - базова топологія комп'ютерної мережі, у якій кожна робоча станція мережі з'єднується з кількома іншими робочими станціями цієї мережі. Характеризується високою стійкістю до відмов, складністю налаштування. Кожен пристрій в такій підключений за такою топологією має багато можливих шляхів з'єднання з іншими пристроями

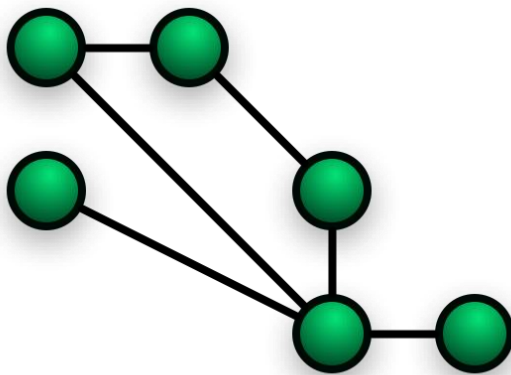


Рисунок 1.7 – Схема з'єднання вузлів в «Сітчастій топології»

1.2.3 Способи взаємодії між IoT пристроями

“Пристрій – пристрій” - два або більше пристроїв IoT можуть зв'язуватися між собою безпосередньо, а не через посередника або сервіс. Приклад: смарт-годинник з'єднується зі смартфоном.

“Пристрій – шлюз” - пристрій IoT з'єднується із сервісом або шлюзом, який виконує роль повнофункціональної точки обслуговування. Наприклад, локальний домашній шлюз може з'єднуватися з різними пристроями розумного будинку (термостат, система безпеки та інших.) і керувати ними автономному режимі. Також шлюз може забезпечувати можливість віддаленого керування такими пристроями, наприклад зі смартфона через Інтернет.

“Пристрій – Інтернет” - пристрої IoT можуть безпосередньо з'єднуватися з програмою, що працює у хмарі, для обміну даними та отримання повідомлень, що управляють. Наприклад, камери відеоспостереження можуть з'єднуватися з хмарою через точку доступу, яка не є точкою обслуговування або пунктом призначення, а виконує суто посередницьку роль.

2 АНАЛІЗ ТЕХНОЛОГІЇ ДЛЯ ПЕРЕДАЧІ ДАНИХ МІЖ ПРИСТРОЯМИ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ТРЕКІНГУ ОБ'ЄКТІВ

2.1 Використання технології Bluetooth для передачі даних

Bluetooth – це технологія бездротового зв'язку короткого діапазону, яка дозволяє таким пристроям, як мобільні телефони, комп'ютери та периферійні пристрої, передавати дані або бездротово на невеликій відстані.

Основне призначення Bluetooth – забезпечення економічного (з точки зору споживаного струму) та дешевого радіозв'язку між різними типами електронних пристроїв, причому чимале значення надається компактності електронних компонентів, що дає можливість застосовувати Bluetooth у малогабаритних пристроях.

Працюючий на частоті 2.4 ГГц приймач, яким є Bluetooth-чип, дозволяє в залежності від ступеня потужності встановлювати зв'язок в межах 10 або 100 метрів.

Bluetooth працює за принципом FHSS (Frequency-Hopping Spread Spectrum). Коротко це можна пояснити так: передавач розбиває дані на пакети і передає їх за псевдовипадковим алгоритмом стрибкоподібної перебудови частоти (1600 разів на секунду) або шаблоном, складеним з 79 підчастот. "Зрозуміти" один одного можуть тільки ті пристрої, які налаштовані на той самий шаблон передачі - для сторонніх приладів передана інформація буде звичайним шумом.



Рисунок 2.1 – Чіп Bluetooth

В даний час існує два типи пристроїв з підтримкою Bluetooth:

- Bluetooth Classic (BR/EDR)
- Bluetooth Low Energy (BLE)

Таблиця 2 - Порівняння Bluetooth Classic та BLE

Bluetooth Classic	BLE
Використовується для потокових програм, таких як трансляція аудіо та передача файлів	Використовується в сенсорах, керуванні пристроями та додатках, що не потребують передачі великих обсягів даних

Таблиця 2 - Порівняння Bluetooth Classic та BLE

Bluetooth Classic	BLE
Не оптимізований для низького енергоспоживання, але підтримує велику швидкість передачі (максимум 3 Мбіт/с, у той час як BLE 5 має максимум 2 Мбіт/с)	Призначений для застосування в пристроях, що мало споживають, з великими інтервалами між передачею даних
Для передачі даних можуть бути використані асиметричний (721 Кбіт/сек в одному напрямку та 57,6 Кбіт/сек в іншому) та симетричний методи (432,6 Кбіт/сек в обох напрямках).	Швидкість передачі може становити 1 Мбіт/с як у попередніх версіях або 2 Мбіт/с при використанні високошвидкісної передачі. Але пропускна спроможність BLE обмежена фізичною пропускною спроможністю радіоканалу, тобто швидкістю, з якою дані передаються по радіоканалу.
Обсяг даних, що передаються, не мають значення	BLE підходить для випадків, коли пристрій передає невеликі обсяги даних

Більшість пристроїв BLE можуть організовуватися в різні види мереж типу, просту мережу з топологією «крапка- крапка» (p2p) або mesh-мережа.

На відміну від p2p, mesh-мережа дозволяє кожному пристрою взаємодіяти з будь-яким іншим пристроєм в мережі, без центрального сервера і навіть без безпосереднього контакту. За рахунок того що кожен пристрої є ретранслятором, така мережа може мати безліч конфігурацій і значну зону покриття при високій стійкості до збоїв і помилок в передачі даних вузла.

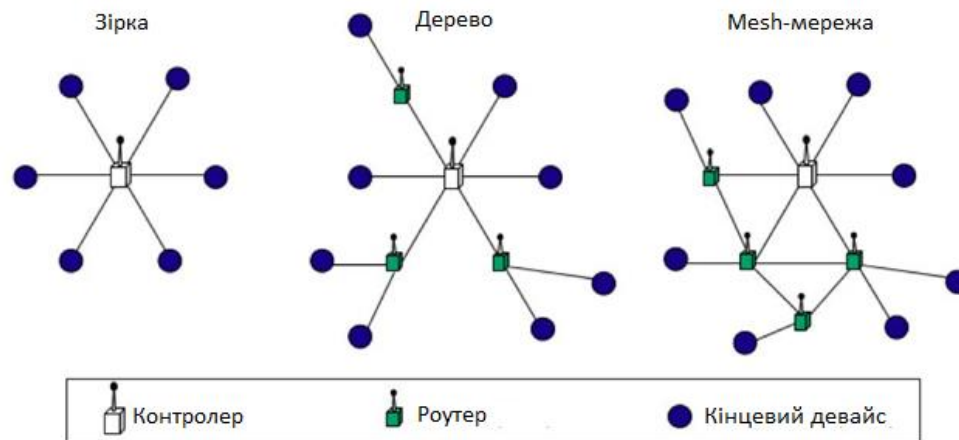


Рисунок 2.2 – Типи топологій підключення

2.2 Використання технології Zigbee для передачі даних

Zigbee – стандарт для набору високорівневих протоколів зв'язку, що використовують невеликі малопотужні цифрові трансівери, що базується на стандарті IEEE 802.15.4-2006 для бездротових персональних мереж. Технологія визначається специфікацією Zigbee, розробленою з наміром бути простіше та дешевше, ніж інші персональні мережі, такі як Bluetooth. Zigbee призначений для радіочастотних пристроїв, де потрібна тривала робота від батарейок та безпека передачі даних по мережі.

Мета технології Zigbee - це створення недорогої мережі, що самоорганізується, з пористою топологією призначеної для вирішення широкого кола завдань.



Рисунок 2.3 – Чіп Zigbee

Існують три різні типи пристроїв Zigbee:

- Координатор Zigbee (ZC) - найбільш відповідальний пристрій, що формує шляхи дерева мережі і може зв'язуватися з іншими мережами. Кожна мережа має один координатор Zigbee. Він і запускає мережу від початку. Він зберігає інформацію про мережу, постає як довірений центр і зберігає ключі безпеки.
- Маршрутизатор Zigbee (ZR) - може виступати як проміжний маршрутизатор, передаючи дані з інших пристроїв. Він також може запускати функцію програми.
- Кінцевий пристрій Zigbee (ZED) - його функціональна навантаженість дозволяє йому обмінюватися інформацією з материнським вузлом (або координатором або маршрутизатором), він не може передавати дані з інших пристроїв. Таке відношення дозволяє вузлу левову частину часу перебувати в сплячому стані, що дозволяє заощаджувати енергоресурс батарей.

На малюнку нижче – приклад перепідключення. Пристрій з під номером 3 перепідключається як пристрій 4, а потім як пристрій 5. Щоразу воно приєднується до

іншого маршрутизатора і отримує адресу з наявного в розпорядженні маршрутизатора діапазону адрес, що приєднає.

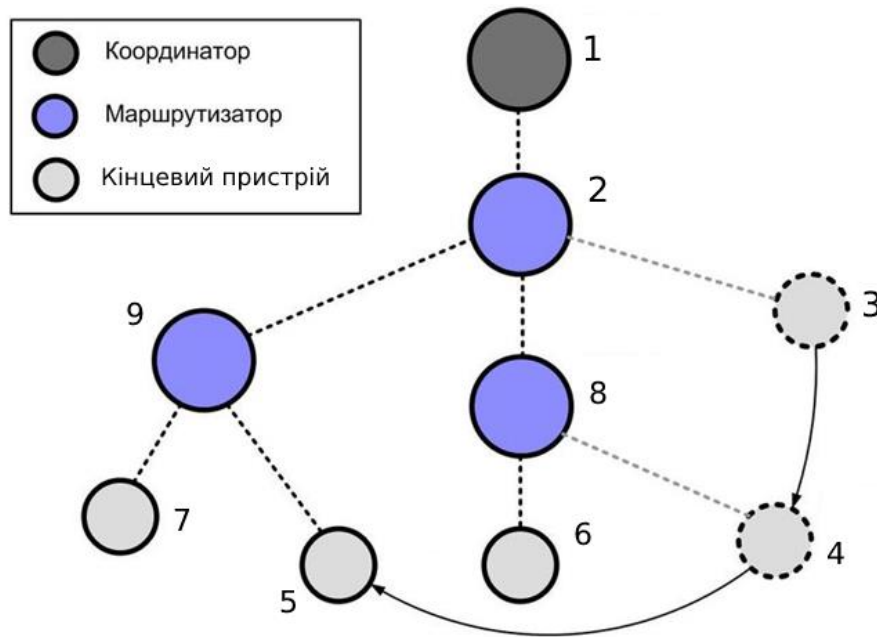


Рисунок 2.4 – Схема підключення пристроїв за допомогою Zigbee

Дуже низьке енергоспоживання Zigbee-модулів досягається за рахунок спеціального режиму "сну", коли пристрій не використовується, і низької пропускної здатності передачі даних до 250 Кбіт / с, хоча середня швидкість передачі корисних даних, залежно від завантаження мережі та кількості ретрансляцій, становить від 5 до 40 Кбіт/с.

Відстань між робочими станціями мережі становить десятки метрів усередині приміщень та сотні метрів на відкритому повітрі. За рахунок ретрансляцій зона, що покривається мережею, може бути дуже значною: до декількох тисяч квадратних метрів у приміщенні і до кількох гектарів на відкритому просторі. Більше того, мережа ZigBee у будь-який момент може бути розширена додаванням нових елементів або, навпаки, розбита на кілька зон простим призначенням відповідного числа нових

конфігураторів мережі. Це корисно зниження навантаження і підвищення швидкості передачі даних.

В плані безпеки ZigBee може запропонувати 128-бітовий алгоритмом AES, використовуваним для шифрування даних і аутентифікації, і трьома типами ключів, використовуваних для управління безпекою, кінцевим користувачам на перший погляд не варто турбуватися.

2.3 Використання технології LPWAN для передачі дани

LPWAN — бездротова технологія передачі невеликих даних на далекі відстані, розроблена для розподілених мереж телеметрії, міжмашинної взаємодії та інтернету речей. LPWAN є однією з бездротових технологій, що забезпечують середовище збору даних з різного обладнання: датчиків, лічильників та сенсорів.

В основі принципу передачі даних за технологією LPWAN фізично лежить властивість радіосистем - збільшення енергетики, а значить і дальності зв'язку при зменшенні швидкості передачі. Чим нижче бітова швидкість передачі, тим більше енергії вкладається в кожен біт і тим легше виділити його на тлі шумів у приймальній частині системи. Таким чином, низька швидкість передачі даних дозволяє досягти більшої дальності їхнього прийому.

Швидкість передачі даних LPWAN-радіоканалу залежить від конкретної задачі, що вирішується і, як правило, становить від декількох біт до декількох сотень біт в секунду. Економія на обсязі інформації, що передається, дозволяє збільшити енергетику радіосигналу і, як наслідок, значно підвищити енергетичний потенціал каналу зв'язку. Висока енергетика використовується для збільшення дальності передачі сигналу та підвищення його проникаючої здатності.

В результаті дальність передачі даних у мережі LPWAN у кілька разів перевищує дальність дії стільникової мережі та становить від кількох кілометрів до кількох десятків кілометрів. Виграш у дальності дії однієї базової станції дозволяє охоплювати більші площі меншою кількістю станцій, що, своєю чергою, скорочує витрати на інфраструктуру та обслуговування мереж LPWAN.

Пристрої здатні працювати в мережах LPWAN протягом багатьох років. Велика автономність досягається, з одного боку, за рахунок використання високоемних акумуляторів, не схильних до саморозряду, а з іншого — за рахунок низького енергоспоживання.

У процесі передачі даних пристрій споживає близько 50 мА протягом кількох секунд. Цього часу достатньо передачі даних у мережу. Решту часу прилад знаходиться в сплячому режимі, споживаючи кілька мікроампер.

Час затримки під час передачі сигналу не є ключовою вимогою в LPWAN-мережах і становить кілька секунд.

Бездротовий сегмент LPWAN-мережі використовує топологію «зірка», тобто всі пристрої надсилають дані безпосередньо на базові станції.

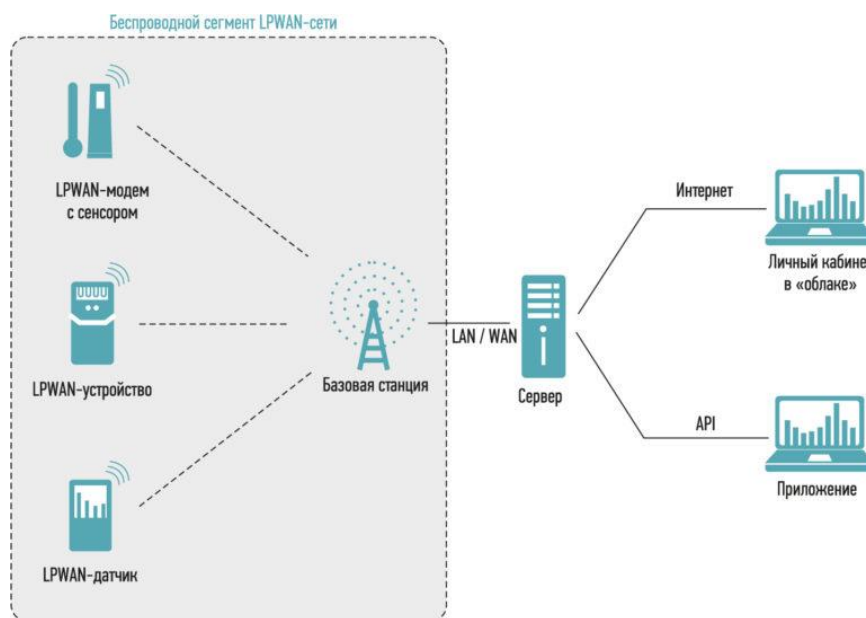


Рисунок 2.5 – Схема підключення пристроїв за допомогою топології «зірка» в мережі LPWAN

Топологія "зірка" дає незаперечну перевагу для "розумної" мережі: таким чином навмисно виключаються складнощі, пов'язані з використанням mesh-протоколу, який, крім низької надійності, провокує додаткову витрату енергії на передачу даних між пристроями.

Також LPWAN-мережа може будуватися за принципом «зірка із зірок». Кожна базова станція приймає дані різних пристроїв і передає їх на сервер для подальшої обробки.

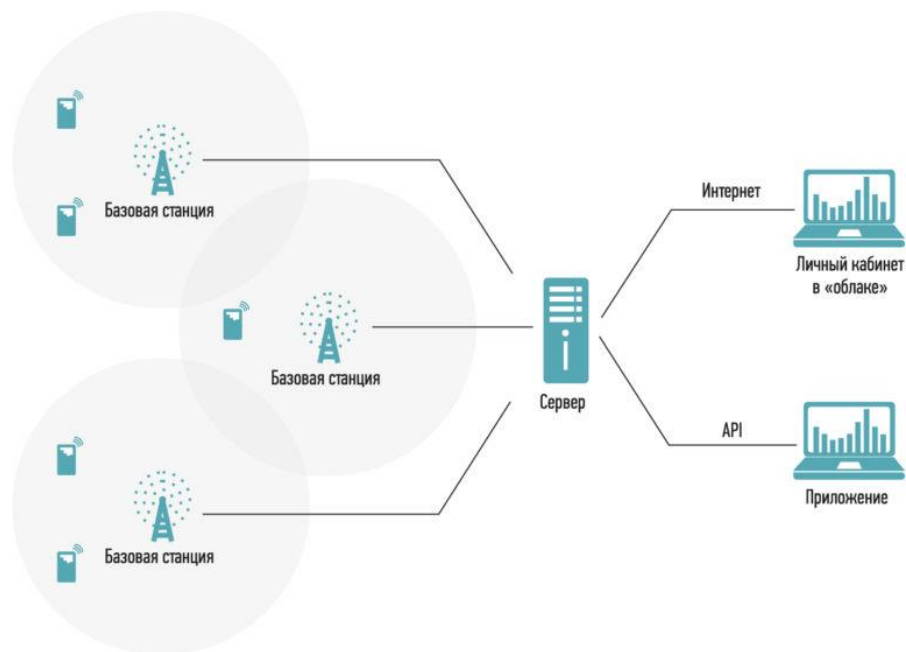


Рисунок 2.6 – Схема підключення пристроїв за допомогою топології «зірка із зірок» в мережі LPWAN

Переваги LPWAN

- Велика дальність передачі радіосигналу в порівнянні з іншими бездротовими технологіями, досягає 10-15 км.
- Низьке енергоспоживання кінцевих пристроїв, завдяки мінімальним витратам енергії на передачу невеликого пакету даних.

- Висока здатність радіосигналу в міській забудові при використанні частот суб-гігагерцового діапазону.
- Висока масштабованість мережі на великих територіях.
- Відсутність необхідності отримання частотного дозволу та плати за радіочастотний спектр, внаслідок використання неліцензованих частот

Недоліки LPWAN

- Відносно низька пропускна здатність внаслідок використання низької частоти радіоканалу. Варіюється в залежності від технології передачі даних, що використовується на фізичному рівні, становить від декількох сотень біт/с до декількох десятків кбіт/с.
- Затримка передачі даних від датчика до кінцевої програми, пов'язана з часом передачі радіосигналу, може досягати від кількох секунд до кількох десятків секунд.
- Відсутність єдиного стандарту, який визначає фізичний шар та керування доступом до середовища для бездротових LPWAN-мереж.

2.4 Використання технології Z-Wave для передачі даних

Z-Wave — це протокол бездротового зв'язку, який використовується в основному в розумних домашніх мережах, що дозволяє розумним пристроям підключатися та обмінюватися командами керування та даними один з одним.

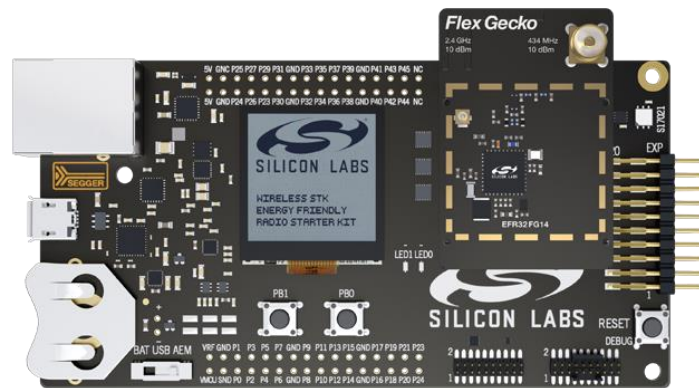


Рисунок 2.7 – Чіп Z-Wave 700

Мережа Z-Wave складається з пристроїв Інтернету речей і основного контролера, також відомого як центр розумного дому. Центральний центр розумного дому, який ви використовуєте для керування пристроями Z-Wave, підключається до Інтернету, але самі пристрої – датчики, лампочки тощо – взагалі не мають Wi-Fi, вони просто використовують підключення Z-Wave для розмови до концентратора, і це підключення не повинно бути прямим. На відміну від Wi-Fi, де пристрої мають підключатися до центрального концентратора (зазвичай маршрутизатора або іншої точки доступу), усі пристрої Z-Wave об'єднуються разом, утворюючи сітку.

Кожна мережа Z-wave складається з двох широких категорій пристроїв;

- Контролер
- Пристрої

Контролер зазвичай служить хостом мережі Z-Wave, до якої можна підключити інші пристрої. Зазвичай він поставляється з попередньо запрограмованим ідентифікатором мережі (іноді його називають HomeID), який призначається кожному підпорядкованому пристрою (який не постачається з попередньо запрограмованим ідентифікатором), коли вони додаються до мережі за допомогою процесу, який називається «включення». Окрім HomeID, для кожного пристрою, доданого до мережі Z-wave, контролер зазвичай призначає ідентифікатор, який називається NodeID.

NodeID є унікальним у кожній мережі (для кожного HomeID), тому він використовується для адресації та розпізнавання кожного пристрою в певній мережі.

Включення подібне до того, як маршрутизатор призначає IP-адреси пристроям у своїй мережі, з єдиною різницею, що головні контролери мають сітчасте з'єднання із підлеглими в мережі. Для видалення вузлів із мережі Z-Wave виконується процес під назвою «Виключення». Під час виключення з пристрою видаляються Home ID та Node ID. Пристрій скидається до заводського стану за замовчуванням (контролери мають свій власний Home ID, а підпорядковані не мають Home ID).

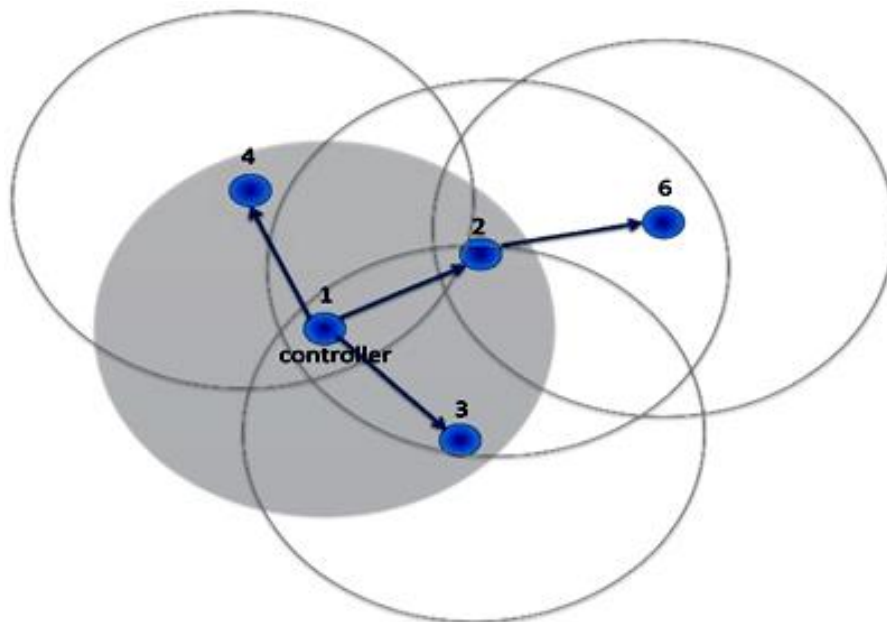


Рисунок 2.8 - Схема підключення пристроїв в мережу Z-Wave

Використовуючи технологію мережевої мережі з маршрутизацією джерела, сигнали Z-Wave можуть стрибати через інші пристрої Z-Wave, щоб досягти пристрою, яким користувач має намір керувати. Кожна мережа Z-Wave вміщує максимум чотири стрибки.

Протокол Z-Wave працює в діапазоні низьких частот 908,42 в США і 868,42 МГц в Європі. Хоча можливі перешкоди в роботі іншої побутової електроніки, наприклад, бездротових телефонів, протокол дозволяє уникнути перешкод у діапазоні 2,4 ГГц, де працюють Wi-Fi і Bluetooth.

Z-Wave пропонує швидкість передачі невеликих пакетів даних зі швидкістю 9,6 Кбіт/с, 40 Кбіт/с або 100 Кбіт/с. Він також включає шифрування AES 128, IPv6 і багатоканальну роботу.

Зв'язок між пристроями коливається від 27 до 100 метрів; серія 500 має радіус дії 40 метрів, а серія 700 — 100 метрів. Оскільки стіни та інші щільні будівельні матеріали обмежують радіус дії, загальна найкраща практика розгортання – розташовувати пристрої Z-Wave на відстані 40 метрів або менше один від одного для максимальної потужності сигналу.

Великий плюс Z-Wave полягає в тому, що її пристрої повністю взаємозв'язані. Усі без винятку пристрої Z-Wave працюють з іншими пристроями Z-Wave – і це завдяки тому, що Z-Wave Alliance підтримується приватною організацією. Зараз Z-Wave Alliance налічує понад 700 членів, а також понад 3300 сертифікованих продуктів цих брендів у всьому світі.

Щоб мати бренд Z-Wave, продукти повинні пройти сертифікацію Z-Wave. Це передбачає виконання вимог і, головне, взаємодію з усіма іншими сертифікованими пристроями Z-Wave.

2.5 Використання технології Wi-Fi для передачі даних

Wi-Fi – це радіосигнал, який надсилається з бездротового маршрутизатора на сусідній пристрій, який перетворює сигнал у дані, які ви можете бачити та використовувати.

Технологія Wi-Fi бере свій початок з рішення Федеральної комісії зв'язку США від 1985 року, яке випустило діапазони радіочастот на частотах 900 мегагерц (МГц), 2,4 гігагерц (ГГц) і 5,8 ГГц для неліцензованого використання будь-ким.

Підключення до Інтернету відбувається через бездротовий маршрутизатор. Коли ви отримуєте доступ до Wi-Fi, ви підключаєтеся до бездротового маршрутизатора, який дозволяє вашим Wi-Fi-сумісним пристроям взаємодіяти з Інтернетом.

Wi-Fi – це високошвидкісне підключення до Інтернету та мережеве підключення без використання будь-яких кабелів чи проводів. Бездротова мережа керує трьома основними елементами: радіосигналом, антеною та маршрутизатором. Радіохвилі – це ключі, які дозволяють працювати в мережі Wi-Fi. Комп'ютери та мобільні телефони готові з картами Wi-Fi. Сумісність з Wi-Fi використовувала нове створення для створення на землі, пов'язаної з мережею громади

Щоразу, коли комп'ютер отримує сигнали в межах 100-150 футів для маршрутизатора, він негайно підключає пристрій.

Діапазон Wi-Fi залежить від навколишнього середовища, діапазону дії в приміщенні або на вулиці. Карти Wi-Fi зчитують сигнали та створюють інтернет-з'єднання між користувачем і мережею. Швидкість пристрою, який використовує з'єднання Wi-Fi, збільшується, коли комп'ютер наближається до основного джерела, а швидкість зменшується, комп'ютер віддаляється.

Наразі це чотири основних типи технологій WIFI.

- Wi-Fi-802.11a
- Wi-Fi-802.11b
- Wi-Fi-802.11g
- Wi-Fi-802.11n
- Wi-Fi- 802.11ac
- Wi-Fi- 802.11ax

Wi-Fi-802.11a

802.11a - підтримує пропускну здатність до 54 Мбіт/с і сигнали в регульованому частотному спектрі близько 5 ГГц.

Плюси 802.11a - швидка максимальна швидкість; регульовані частоти запобігають перешкоджанню сигналу від інших пристроїв

Мінуси 802.11a - найвища вартість; сигнал меншої дальності, який легше перешкоджати

Wi-Fi-802.11b

802.11b є однією з серії бездротових технологій. 802.11b підтримує пропускну здатність 11 Мбіт/с. Сигнал в нерегульованому частотному спектрі близько 2,4 ГГц. Це низька частота в порівнянні з Wi-Fi-802.11a, що означає, що він працює на більшій відстані.

Плюси 802.11b - найнижча вартість; Дальність сигналу хороша і її не легко перешкодити

Мінуси 802.11b - найнижча максимальна швидкість; побутова техніка може створювати перешкоди в нерегульованому діапазоні частот

Wi-Fi-802.11g

Wi-Fi-802.11g - розроблений у 2002 і 2003 роках на ринку з'явилися продукти WLAN, які підтримують новий стандарт під назвою 802.11g. 802.11g намагається поєднати найкраще з 802.11a і 802.11b. 802.11g підтримує пропускну здатність до 54 Мбіт/с і використовує частоту 2,4 ГГц для більшого діапазону. 802.11g зворотно сумісний із 802.11b, а це означає, що точки доступу 802.11g працюватимуть з адаптерами бездротової мережі 802.11b і навпаки.

Плюси 802.11g - швидка максимальна швидкість, хороша дальність сигналу, сигнал стійкий до перешкод.

Мінуси 802.11g - коштує більше ніж 802.11b, прилади можуть створювати перешкоди на нерегульованій частоті сигналу

Wi-Fi-802.11n

802.11n - він був розроблений для покращення стандарту 802.11g. Пропускна здатність підтримується за рахунок використання кількох бездротових сигналів і

антен замість однієї. Він підтримує пропускну здатність 100 Мбіт/с і підвищену інтенсивність сигналу.

Плюси 802.11n - найшвидша максимальна швидкість і найкращий діапазон сигналу, більш стійкий до перешкод сигналу від зовнішніх джерел

Мінуси 802.11n - стандарт ще не доопрацьований; коштує більше 802.11g; використання кількох сигналів може сильно завадити роботі найближчих мереж 802.11b/g

Wi-Fi- 802.11ac

802.11ac - з'явився приблизно в 2014 році і працює виключно на частоті 5 ГГц. 802.11ac зворотно сумісний з 802.11n (і, отже, з іншими також), з n у діапазоні 2,4 ГГц і ac у діапазоні 5 ГГц. Він менш схильний до перешкод і набагато швидше, ніж його попередники, забезпечуючи максимальну швидкість 450 Мбіт/с для одного потоку, хоча реальні швидкості можуть бути нижчими. Як і 802.11n, він дозволяє здійснювати передачу в кількох просторових потоках — до восьми, за бажанням. Іноді його називають 5G через його частотний діапазон, іноді Gigabit WiFi через його потенціал перевищувати гігабіт на секунду на кількох потоках, а іноді і дуже високою пропускну здатністю з тієї ж причини.

Плюси 802.11ac – забезпечує покращену пропускну здатність і гнучкість завдяки одночасному підключенню; зворотна сумісність дозволяє використовувати існуючі технології

Мінуси 802.11ac – подвійні діапазони означають збільшення вартості, все ще схильний до перешкод на частоті 2,4 ГГц

Wi-Fi- 802.11ax

802.11ax, також відомий як WiFi 6, з'явився в 2019 році. Цей стандарт розширює можливості 802.11ac кількома ключовими способами. Перш за все, нові маршрутизатори забезпечують ще більшу швидкість потоку даних, до 9,2 Гбіт/с. WiFi 6 також дозволяє виробникам встановлювати набагато більше антен на одному маршрутизаторі, приймаючи кілька підключень одночасно, не турбуючись про

перешкоди та сповільнення. Деякі нові пристрої також підключаються до вищого діапазону 6 ГГц, що приблизно на 20 відсотків швидше, ніж 5 ГГц в ідеальних умовах.

Плюси 802.11ac – забезпечує покращену пропускну здатність, новий діапазон частот який забезпечує більшу швидкість передачі даних.

Мінуси 802.11ac – все ще схильний до перешкод на частоті 2,4 ГГц

Найпоширеніший стандарт бездротового шифрування, Wired Equivalent Privacy (WEP Encryption). Зашифровує повідомлення, надіслані через бездротові мережі, так що їх неможливо легко прочитати. Існують різні форми шифрування, але остання версія — Wi-Fi Protected Access II (WPA2), оскільки вона сильніша за інші версії, такі як Wired Equivalent Privacy (WEP) і WPA.

3 СПОСОБИ ОТРИМАННЯ ГЕОПОЗИЦІЇ ОБ'ЄКТІВ

Геолокація відноситься до будь-якого типу технології, яка здатна визначити географічне розташування пристрою. Знаходячи пристрій відстеження активів у реальному часі, ви можете знайти важливий актив, такий як контейнер, причіп, піддон тощо. Часто пристроєм відстеження є мобільний телефон або пристрій, підключений до Інтернету (Internet of Things).

Для автоматичного позиціонування доступні різні технології. Тип обраної технології зазвичай залежить від двох основних факторів.

- Точність розташування: деякі технології можуть відстежувати переміщення об'єктів аж до воріт. Це корисно для тих об'єктів, переміщення яких потрібно відстежувати у простих випадках, наприклад, при надходженні товару на склад або його відправлення зі складу. Інші технології можуть відстежувати розташування об'єкта з великою точністю (до метра або навіть менше). Це корисно у тих ситуаціях, коли потрібне точне визначення місцезнаходження.
- Швидкість поновлення. Отримання інформації в режимі реального часу не потрібне для всіх рішень. Наприклад, якщо товар був просканий при проходженні через ворота складу і тепер знаходиться на стелажі, постійне оновлення не потрібно. Однак інші рішення можуть надавати оновлені дані про місцезнаходження щогодини або в режимі реального часу, наприклад, у межах хвилин, секунд або менше. Прикладом може бути рішення, яке потребує постійного оновлення при відстеженні розташування транспортного засобу в міру того, як воно рухається.

Існують різні підходи для генерації інформації про місцезнаходження об'єктів:

- Обчислення. Включає тристороннє вимірювання на основі тимчасової різниці прийому сигналу, кута прийому, значень потужності сигналу. Зазвичай досягається за рахунок довкілля, де об'єкти повинні бути розташовані разом з інфраструктурою, прикладом такого способу може бути триангуляція в стільниковому зв'язку.
- Наявність. Використання порталів та радіомаяків для відстеження приходу та догляду, визначення розташування на основі найближчого маяка або екстраполяції між двома маяками.
- Рух. Використання пристроїв, що переміщуються, які можуть встановлюватися на предмет для визначення розташування предмета з міткою.

Критерії, які необхідно враховувати при виборі технології для визначення глокації:

- Діапазон. Який діапазон об'єктів, що відстежуються? Вони розкидані по всьому світу / чи знаходяться на об'єкті / у приміщенні?
- Перешкодостійкість. Залежить від наявності перешкод, а також часу/кута/сигналу
- Точність координат. Як точно потрібно визначити місце розташування об'єкта
- Частота поновлення інформації. Як часто потрібно оновлювати інформацію про місцезнаходження
- Інфраструктура. Існуюча (Wi-Fi), зовнішня (стільникова/супутникова), проста чи складна
- Час роботи від батареї. Щоденна підзарядка, щорічна заміна батарей, заміна міток кожні 5–10 років
- Вартість. Вартість міток, інфраструктури, впровадження та обслуговування

			
	GPS	WiFi	LBS
Ідеальне Середовище	Буть-де	В приміщенні	Буть-де
Спроживання енергії	Високе	Середнє	Низьке
Точність позиціонування	10-50 м.	15-700 м.	500-5000 м.
Необхідна інфраструктура	Ні	Так, але вже існує	

Рисунок 3.1 – Таблиця з порівнянням технологій для визначення геолокації

3.1 Використання технології GPS для визначення геолокації

Глобальна система позиціонування (GPS) — це навігаційна система, що використовує супутники, приймач і алгоритми для синхронізації даних про місцезнаходження, швидкість і час для повітряних, морських і наземних подорожей.

Супутникова система складається з сузір'я з 24 супутників у шести орбітальних площинах з центром Землі, кожна з яких має чотири супутники, які обертаються на висоті 20 000 км над Етом і рухаються зі швидкістю 14 000 км/год.

Хоча нам потрібні лише три супутники, щоб визначити місцезнаходження на поверхні Землі, четвертий супутник часто використовується для перевірки інформації з трьох інших. Четвертий супутник також переміщує нас у третій вимір і дозволяє розрахувати висоту пристрою.

GPS працює за допомогою техніки, яка називається трилатерація. Трилатерація, що використовується для розрахунку розташування, швидкості та висоти, збирає сигнали від супутників для виведення інформації про місцезнаходження. Його часто помилково приймають за триангуляцію, яка використовується для вимірювання кутів, а не відстаней.

Супутники, що обертаються навколо Землі, надсилають сигнали для зчитування та інтерпретації за допомогою пристрою GPS, розташованого на поверхні землі або поблизу неї. Щоб розрахувати місцезнаходження, пристрій GPS повинен мати можливість зчитувати сигнал щонайменше з чотирьох супутників.

Кожен супутник в мережі обертається навколо Землі двічі на день, і кожен супутник посилає унікальний сигнал, орбітальні параметри та час. У будь-який момент GPS-пристрій може зчитувати сигнали з шести або більше супутників.

Один супутник передає мікрохвильовий сигнал, який вловлюється пристроєм GPS і використовується для обчислення відстані від пристрою GPS до супутника. Оскільки пристрій GPS дає лише інформацію про відстань від супутника, один супутник не може надати багато інформації про місцезнаходження. Супутники не видають інформацію про кути, тому місцезнаходження пристрою GPS може бути в будь-якому місці поверхні сфери.

Коли супутник надсилає сигнал, він створює коло з радіусом, вимірним від пристрою GPS до супутника. Коли ми додаємо другий супутник, він створює друге коло, а розташування звужується до однієї з двох точок, де кола перетинаються. За допомогою третього супутника можна нарешті визначити місцезнаходження пристрою, оскільки пристрій знаходиться на перетині всіх трьох кіл.

Тим не менш, ми живемо в тривимірному світі, а це означає, що кожен супутник створює сферу, а не коло. Перетин трьох сфер дає дві точки перетину, тому вибирається найближча до Землі точка.

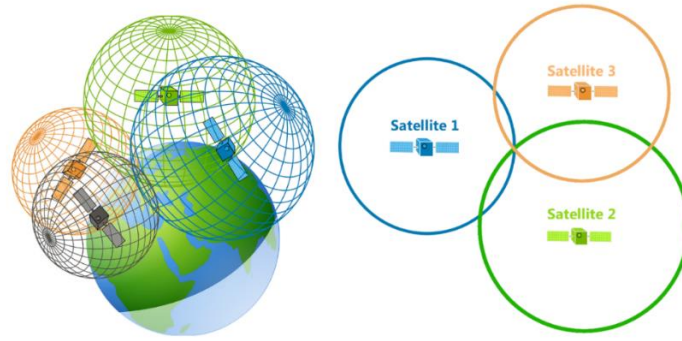


Рисунок 3.2 – Схема роботи супутників GPS

Коли пристрій рухається, радіус (відстань до супутника) змінюється. Коли радіус змінюється, утворюються нові сфери, що дає нам нове положення. Ми можемо використовувати ці дані в поєднанні з часом від супутника, щоб визначити швидкість, розрахувати відстань до пункту призначення та приблизний час прибуття.

Великим плюсом GPS є його точність. Він може визначити місцезнаходження на відстані до п'яти метрів з точністю або навіть краще за допомогою дводіпазонних приймачів GPS. Точність залежить від багатьох факторів, також важливо враховувати час, необхідний для визначення позиції, час виправлення.

Найкращі результати досягаються, коли антена GPS має чисте небо. Навпаки, коли сигнал GPS відбивається або перешкоджає, це може зайняти набагато більше часу або навіть стати неможливим, щоб отримати точне положення. Чим більше часу потрібно, щоб отримати фіксацію GPS, тим більше енергії споживається.

Існує також важлива відмінність між холодною і гарячою корекцією. Холодне виправлення — це визначення позиції, коли супутникові дані недоступні, наприклад, після періоду бездіяльності. Якщо пристрій відстеження вже отримав позиції супутника, оскільки він постійно відстежує, він може знайти точне положення набагато швидше, це гаряче виправлення.

Ще одна перевага полягає в тому, що GPS працює скрізь на відкритому повітрі і не вимагає спеціальної інфраструктури. Недоліком є те, що ця технологія геолокації вимагає досить багато енергії, оскільки вона має прослуховувати кілька супутників, а це може зайняти деякий час, особливо у випадку холодного виправлення. Іншими недоліками є те, що зв'язок на великій відстані може бути перервана через погодні умови та неможливість працювати в приміщенні.

Переваги технологій GPS

- Точне розташування на відкритому повітрі (до п'яти метрів)
- Працює майже скрізь на відкритому повітрі
- Додаткова інфраструктура не потрібна

Недоліки технологій GPS

- Високе споживання енергії
- Не працює в приміщенні
- Ризик перерви через погодні ситуації

3.2 Використання технології Wi-Fi для визначення геолокації

Досить новий метод визначення розташування пристрою - це прослуховування сигналів Wi-Fi навколо нього.

Завдяки Wi-Fi ви можете покладатися на легкодоступну інфраструктуру для визначення геолокації об'єкта і не турбуватися про розрядження акумулятора, як у

випадку з GPS. Але вам знадобиться велика кількість точок доступу, щоб точно визначити позицію, і ви не зможете визначити позицію поза зоною покриття Wi-Fi.

Позиціонування Wi-Fi підключається до бездротових локальних мереж (WLAN), які є мережами пристроїв, які підключаються до певної радіочастоти, як правило, 2,4 ГГц або 5,0 ГГц. Пристрій Wi-Fi передає сигнали на діапазон до ста метрів, що означає, що Wi-Fi може охоплювати як внутрішні, так і зовнішні майданчики. Пристрій відстеження шукає найближчі точки доступу Wi-Fi (AP). Визначаючи унікальний ідентифікатор точки доступу, наприклад, MAC-адресу, можна визначити позицію. Локальні або загальнодоступні бази даних забезпечують зв'язок між спостережуваними MAC-адресами та геолокацією.

Пристрої відстеження нюхають лише сигнали Wi-Fi, їм не потрібно підключатися до Wi-Fi. Тому позиціонування Wi-Fi також використовує мережі Wi-Fi, якими ви не володієте або не можете отримати доступ. Наприклад, коли комерційний трейлер проїжджає через міський центр, він проїжджатиме через сотні чи тисячі мереж Wi-Fi.

Приємним плясом являється те, що має низьке Wi-Fi енергоспоживання. Точність залежить від двох основних факторів: щільності точок доступу та точності бази даних. Існує ризик помилкових записів у базі даних, що призведе до помилкових позицій. Wi-Fi не вимагає додаткової інфраструктури.

Переваги технологій Wi-Fi

- Більша точність ніж у GPS
- Низьке споживання енергії

Недоліки технологій Wi-Fi.

- Може знадобитися локальна інфраструктурна мережа
- Не для місцевості, де немає інфраструктури Wi-Fi

3.3 Використання технології LBS для визначення геолокації

Location-based service (LBS) – сервіс визначення об'єкта шляхом прив'язки до орієнтирів з бази даних постачальника послуг зв'язку. Іншими словами, розташування об'єкта визначається за координатами базових станцій GSM. Точність у LBS менша, але сигнал зловити простіше, а заглушити важче.

Місцезнаходження LBS фіксується на основі даних веж стільникового зв'язку, які отримані від пристрою. Зазвичай пристрій знаходить найближчу вежу стільникового зв'язку та показує її розташування поблизу.

Точність розташування залежить від щільності базових станцій. У містах і розвинених районах точність визначення місцезнаходження може бути до 50-100 м, а в сільській і безлюдній місцевості похибка може досягати кількох кілометрів. Розширені системи також можуть намагатися обчислити відстань між GSM-модулем і станцією. Ці системи зазвичай вимагають модифікації встановленої інфраструктури і є досить дорогими. Звичайний метод пошуку називається E-OTD (Enhanced Observed Time Difference).

Він заснований на вимірюванні та порівнянні різниці в часі надходження сплесків з найближчих базових станцій. Затримки та відоме розташування базових станцій можна використовувати для триангуляції розташування модуля GSM. Цей метод пошуку забезпечує набагато кращу точність навіть у віддалених районах приблизно 125 метрів, але він вимагає, щоб базові станції мали встановлені одиниці вимірювання місцезнаходження.

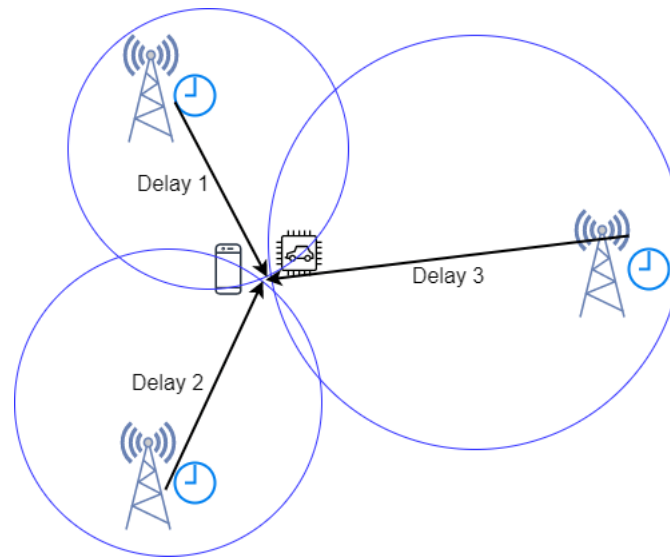


Рисунок 3.3 – Схема роботи E-OTD

LBS вимагає п'яти основних компонентів: програмного додатка постачальника послуг, мобільної мережі для передачі даних і запитів на обслуговування, постачальника контенту для надання кінцевому користувачеві геоспецифічної інформації, компонента позиціонування (див. GPS) і мобільного пристрою кінцевого користувача. пристрій.

Перевага позиціонування LBS є зручною, оскільки воно розташоване за мобільним телефоном, що має велике значення для енергоспоживання пристрою та тривалого часу очікування. Теоретично, якщо розрахувати різницю сигналів між трьома базовими станціями, можна визначити місцезнаходження мобільного телефону. Таким чином, поки мобільний телефон користувача знаходиться в зоні дії мережі мобільного зв'язку, місце розташування можна знайти в будь-який час без впливу погоди, високих будівель, розташування тощо. Позиціонування LBS підпадає під дію двох обмежень:

1. Значення координат позиції, отримане шляхом розрахунку різниці сигналу базової станції, очевидно, поступається точності позиціонування GPS (точність позиціонування базової станції LBS коливається в межах 500-5000 метрів);
2. Сфера використання вузька. Хоча на LBS не вплине погода, віадуки або багатоповерхові будинки, якщо діапазон обслуговування мобільного телефону перевищено, або кількість базових станцій, на яких розташований мобільний телефон, недостатня, позиціонування LBS неможливо виконати;

Переваги Технології LBS:

- Енергозбереження; енергоспоживання пристрою можна значно знизити завдяки спеціальному дизайну та алгоритму; оскільки модуль GPS в пристрої GPS-позиціонування споживає набагато більше енергії, ніж модуль GSM (режим очікування модуля GSM може мати струм 1-2 мА); (а робочий струм GPS 20-50 мА)
- Підходить для розміщення в приміщенні, особистого позиціонування; тому що розташування персоналу – це переважно робота в приміщенні, життя, навчання та вбудований акумулятор для живлення пристрою. Якщо пристрій GPS-позиціонування використовується в приміщенні, виникає проблема, що супутник неможливо знайти, а модуль GPS не припиняє працювати для супутникового сигналу, що ще більше збільшує споживання батареї.

Недоліки технології LBS:

- Погана точність позиціонування, міська зона 50-200 метрів; приміський радіус позиціонування 1000-2000 метрів;
- Дані базової станції будуть змінені, і оператор мобільного зв'язку налаштує, тимчасово збільшить або зменшить розподіл базової станції в міру необхідності, особливо наявність ретранслятора (за відсутності сигналу

втягти сигнал з інших базові станції до Зона використовується для впливу на точність позиціонування LBS, база даних системи позиціонування не оновлюється вчасно, що вплине на правильність позиціонування;

- Дані базової станції LBS є таємницею оператора мобільного зв'язку і, як правило, не є загальнодоступними, або для отримання потрібна авторизація або оплата. Тому використовувати його не так зручно і просто, як GPS-позиціонування.

4 ВИБІР АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПОБУДОВИ ІНФОРМАЦІЙНОЇ СИСТЕМ ТРЕКІНГУ ОБ'ЄКТІВ

4.1 Вибір апаратних засобів для побудови інформаційної систем

В якості центрального контролера був обраний Arduino, тому що це недорога друкована плата з мікро контролером, що володіє відкритою архітектурою. Платформа Arduino здатна зчитувати вхідні дані у вигляді напруги на своїх аналогових контактах. Якщо до певних входів пристрою підключити датчики, то вона програмним способом зчитує інформацію з цих контактів.

Платформа Arduino призначена для створення електронних пристроїв, що працюють за заданим алгоритмом і з можливістю реагувати на зовнішні сигнали. Arduino - це платформа з відкритою принциповою схемою. Дана платформа доступна як для підключення інших пристроїв, так і для зміни самої електричної схеми пристрою.

Порти введення-виведення мікроконтролерів оформлені у вигляді штирьових лінійок. Мікроконтролери живляться від 5 В. Відповідно порти мають такий же розмах допустимих вхідних і вихідних напруги. Програмісту доступні деякі спеціальні можливості портів введення\виведення мікроконтролеру, наприклад широтно-імпульсна модуляція (ШІМ), аналогово-цифровий перетворювач (АЦП), інтерфейси UART, SPI, I2C.

Таблиця 3 – Характеристики Arduino UNO

Мікроконтролер	Atmega328
Тактова частота	16 МГц
Робоча напруга	5В
Напруга живлення (рекомендований)	7-12В
Цифрові входи / виходи	14 (з них 6 можуть використовуватися в якості шім-виходів)
Аналогові входи	6
Максимальний струм одного виведення	40 мА
Максимальний вихідний струм виводу	3.3v 50 мА
Flash-пам'ять	32 Кб з яких 0.5 Кб виділяється під завантаження
Оперетивна пам'ять	2 Кб
Енерго незалежна пам'ять	1 Кб



Рисунок 4.1 – Платформа Arduino UNO

GPS NEO-6M

Модуль GPS NEO-6M забезпечує високу точність визначення позиціонування за допомогою технології GPS.



Рисунок 4.2- Модуль GPS NEO-6M

Технічні характеристики:

- 50-канальний приймач
- Час до першого виправлення - холодний або теплий запуск 27 секунд, гарячий старт - 1 секунда
- Відстеження та навігація - 161dBm
- Повторне отримання -160dBm
- Холодний запуск (без допомоги) -147 дБм
- Гарячий старт -156 дБм
- Максимальна частота оновлення - 1 Гц
- Точність горизонтального положення - 2,5 м
- Точність швидкості - 0,1 м/с

- Точність курсу - 0,5 градуса

Позначення виходів модуля:

- GND - земля
- Vcc - підключити до 5 В або 3,3 В
- TX - вихід для прийому цифрового сигналу
- RX - вихід для передачі цифрового сигналу

ESP8266 NodeMCU V3

Модуль Wi-Fi на базі системи ESP8266. Він завантажив програмне забезпечення NodeMCU, яке забезпечує програмування мовою сценаріїв Lua. Система має 10 контактів GPIO (кожним з них можна керувати за допомогою сигналу ШІМ), шину I2C, 1-Wire, 10-розрядний аналого-цифровий перетворювач і антену для друкованої плати. Крім того, на платі є конвертер USB-UART (CH340), який дозволяє програмувати безпосередньо через порт USB за допомогою середовища Arduino IDE.

Lua — це мова сценаріїв, яка може працювати самостійно або як розширення можливостей інших програм. Він був реалізований як бібліотека мови C. Він відрізняється простим кодом, високою продуктивністю та можливістю перенесення між платформами.

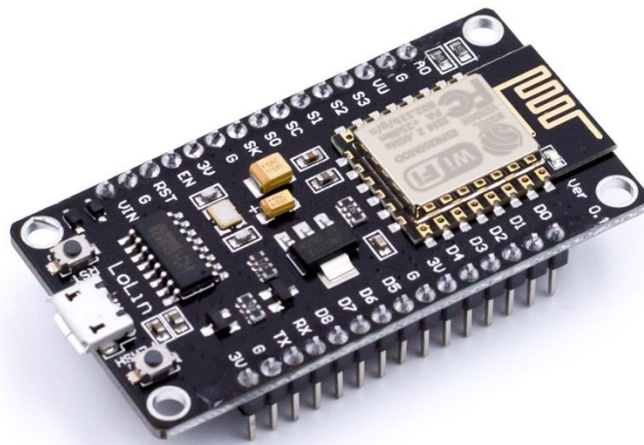


Рисунок 4.3- Модуль ESP8266 NodeMCU V3

Технічні характеристики:

- Напруга комунікаційного інтерфейсу: 3,3 В.
- Тип антени: вбудована антена на друкованій платі.
- Бездротовий стандарт 802.11 b/g/n
- WiFi на частоті 2,4 ГГц, підтримка режиму безпеки WPA / WPA2
- Підтримка трьох режимів роботи STA/AP/STA + AP
- Вбудований стек протоколів TCP/IP для підтримки кількох підключень TCP-клієнтів (5 МАКС.)
- D0 ~ D8, SD1 ~ SD3: використовується як GPIO, PWM, ІС тощо, потужність драйвера порту 15 мА
- AD0: 1 каналний АЦП
- Вхідне живлення: 4,5 В ~ 9 В (10 В), живлення від USB

- Струм: безперервна передача: $\approx 70\text{mA}$ (200mA MAX), режим очікування: $< 200\mu\text{A}$
- Швидкість передачі: 110-460800 біт/с
- Підтримка інтерфейсу передачі даних UART / GPIO
- Віддалене оновлення мікропрограми (OTA)
- Розмір Flash: 4 МБ

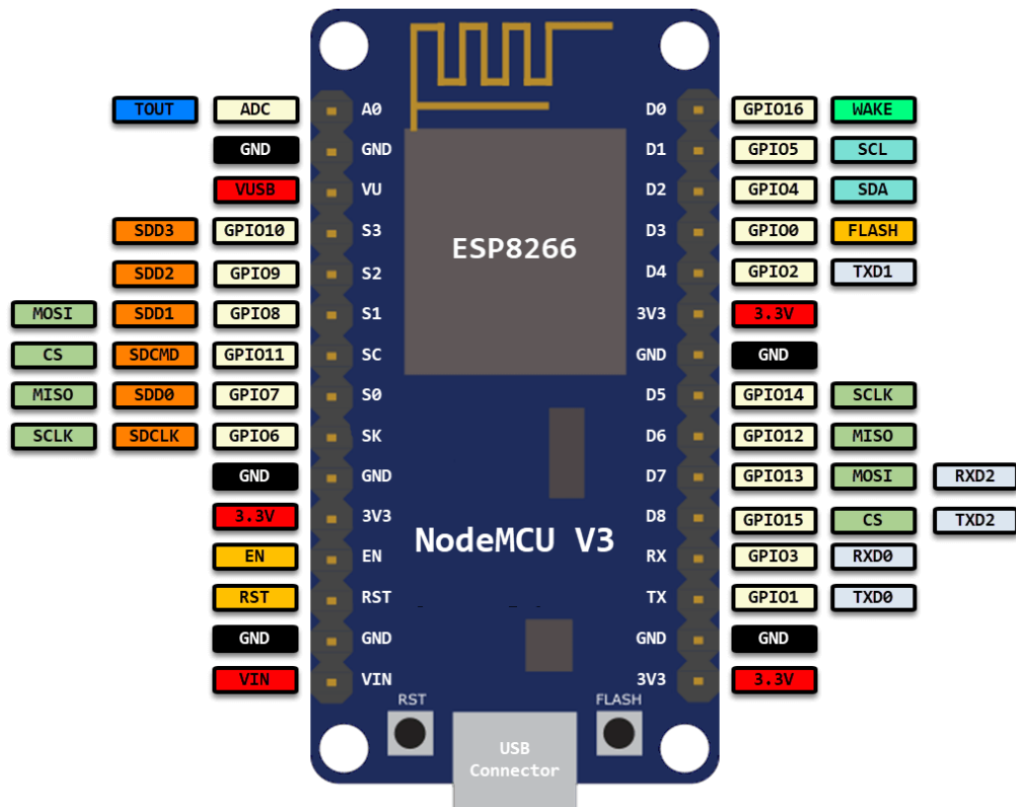


Рисунок 4.4- Позначення виходів модуля ESP8266 NodeMCU V3

4.2 Вибір програмних засобів для побудови інформаційної систем

4.2.1 Вибір програмних засобів для серверної частина системи

Взаємодія між трекерами і сервером буде відбуватися через API по технології REST.

API або інтерфейс прикладного програмування — це набір правил, які визначають, як програми чи пристрої можуть підключатися та спілкуватися один з одним. REST API — це API, який відповідає принципам проектування REST або архітектурному стилю передачі стану репрезентації. З цієї причини API REST іноді називають API RESTful.

API REST спілкуються через HTTP-запити для виконання стандартних функцій бази даних, таких як створення, читання, оновлення та видалення записів (також відомих як CRUD) у ресурсі. Наприклад, REST API використовуватиме запит GET для отримання запису, запит POST для його створення, запит PUT для оновлення запису та запит DELETE для його видалення. Усі методи HTTP можна використовувати у викликах API.

Дані можна надати клієнту практично в будь-якому форматі, включаючи нотацію об'єктів JavaScript (JSON), HTML, XML, Python, PHP або звичайний текст. JSON популярний, оскільки його читають як люди, так і машини, і він не залежить від мови програмування.

Заголовки та параметри запитів також важливі у викликах REST API, оскільки вони містять важливу інформацію про ідентифікатор, таку як метадані, авторизації, уніфіковані ідентифікатори ресурсів (URI), кешування, файли cookie тощо. Заголовки

запитів і відповіді, а також звичайні коди статусу HTTP використовуються в правильно розроблених API REST.

- Для того, щоб API вважався RESTful, він повинен відповідати наступним критеріям:
- Архітектура клієнт-сервер, що складається з клієнтів, серверів і ресурсів, із запитам, які керуються через HTTP.
- Зв'язок клієнт-сервер без стану, тобто інформація про клієнта не зберігається між запитам на отримання, і кожен запит є окремим і не підключеним.
- Дані з кешуванням, які спрощують взаємодію клієнт-сервер.
- Єдиний інтерфейс між компонентами, щоб інформація передавалася у стандартній формі. Для цього потрібно:
 - запитувані ресурси є ідентифікованими та відокремленими від представлень, надісланих клієнту.
 - Клієнт може маніпулювати ресурсами через представлення, яке вони отримують, оскільки подання містить достатньо інформації для цього.
 - самоописні повідомлення, що повертаються клієнту, мають достатньо інформації, щоб описати, як клієнт повинен її обробити.
 - гіпертекст/гіпермедіа доступний, що означає, що після доступу до ресурсу клієнт повинен мати можливість використовувати гіперпосилання, щоб знайти всі інші доступні на даний момент дії, які він може виконати.
- Багатошарова система, яка організовує кожен тип серверів (відповідальних за безпеку, балансування навантаження тощо), передбачала отримання запитуваної інформації в ієрархії, невидимі для клієнта.

Код на вимогу (необов'язково): можливість надсилати виконуваний код із сервера клієнту за запитом, що розширює функціональні можливості клієнта.

Всі данні які будуть передаватися між компонентами моделі будуть в форматі JSON.

JSON – це текстовий формат даних, який відповідає синтаксису об’єкта JavaScript. Незважаючи на те, що він дуже нагадує літеральний синтаксис об’єкта JavaScript, його можна використовувати незалежно від JavaScript, і багато середовищ програмування мають можливість читати (розбирати) та генерувати JSON.

```
{
  "firstName": "Jonathan",
  "lastName": "Freeman",
  "loginCount": 4,
  "isWriter": true,
  "worksWith": ["Spantree Technology Group", "InfoWorld"],
  "pets": [
    {
      "name": "Lilly",
      "type": "Raccoon"
    }
  ]
}
```

Рисунок 4.5 - Приклад JSON об’єкта

Під такі критерії найкраще підходять дві технології на яких можливо створити сервер Node.js та PHP, далі розглянемо кожен з них детальніше.

Node.js

Node.js — це середовище виконання JavaScript з відкритим вихідним кодом та міжплатформним. Це популярний інструмент практично для будь-яких проектів.

Node.js запускає движок JavaScript V8, ядро Google Chrome, поза браузером. Це дозволяє Node.js бути дуже продуктивним.

Програма Node.js виконується в одному процесі, не створюючи новий потік для кожного запиту. Node.js надає в стандартній бібліотеці набір асинхронних примітивів

вводу-виводу, які запобігають блокуванню коду JavaScript, і, як правило, бібліотеки в Node.js написані з використанням неблокуючих парадигм, що робить поведінку блокування скоріше винятком, ніж нормою.

Переваги Node.js

- Можливість застосовувати одну мову на клієнті та сервері. Якщо програміст прокачався в JavaScript, йому буде легше вивчити "надбудову", ніж технологію, що кардинально відрізняється.
- Загальний код. Коду, який використовується і на клієнті, і на серверній стороні, небагато, але він присутній. Головне — розуміти, що часто об'єкти з однаковими назвами можуть виконувати різні функції в браузері і на бекенді.
- Розроблявся спеціально для web. Вільно взаємодіє з найпопулярнішими базами даних, допомагає отримати низькорівневий доступ (http, udp, https, tcp).
- Швидкість. Створення робітника, що справляється з навантаженням, прототипу не забере багато часу. Перший етап, коли програміст формує кістяк майбутнього продукту, відбувається дуже швидко. Якщо архітектура добре продумана, то надалі труднощів для того, щоб розширювати сайт на Node JS теж не з'явиться.
- Велике та ком'юніті. Оскільки відкритий код, веб-розробники можуть писати різні модулі та пакети та ділитися ними. Найчастіше модулі добре комбінуються.
- Синтаксис JavaScript. Це забезпечує відносну простоту та доступність.
- Вбудована бібліотека. З кожним оновленням вона розширюється, але навіть у першій версії були широкі можливості.
- Постійний розвиток екосистеми. Кількість готових модулів та зовнішніх бібліотек безперервно зростає, у чому сприяє використання npm.

- V8. Він економно витрачає пам'ять, непогано оптимізований, дає функціонал із профілювання процесора та пам'яті.

Технологія швидко покращується, над цим працюють тисячі програмістів по всьому світу. Процес координує фонд Node. Важко недооцінити роль ком'юніті, адже це допомога зі складнощами, велика кількість навчальних матеріалів, оперативно знайдені помилки у функціонуванні системи, можливість обговорити ідеї або знайти команду. З "живою" спільнотою розвиток технології відбувається набагато швидше, і все завдяки відкритому коду. Популярність та переваги Node js не залишилися непоміченими, його вже обрали такі компанії, як Google, Ebay/PayPal, Netflix, Microsoft, Uber, LinkedIn, Yahoo, Walmart та інші.

Завдяки цьому інструменту можна створити як програму для Інтернету, так і для Linux, OS X і Windows. Використовуючи бібліотеки, також можна сформувати API. Додаються шляхи синхронізації мобільного з комп'ютером - при написанні тексту на телефоні, він буде видно в Інтернеті та на ПК/ноутбуку. На відміну від звичайного JavaScript, Node вміє взаємодіяти з глобальними об'єктами, включаючи document і window, при створенні програм для Windows. Це відкриває шлях до вінчестера та файлової системи користувача. Крім цього, у поле доступу потрапляють бібліотеки та програмні рішення, які вже є на комп'ютері.

PHP

PHP — це мова сценаріїв на стороні сервера. який використовується для розробки статичних або динамічних веб-сайтів чи веб-додатків. PHP означає Hypertext Pre-processor. PHP-скрипти можна інтерпретувати лише на сервері, на якому встановлено PHP.

Сценарій – це набір програмних інструкцій, які інтерпретуються під час виконання.

Мова сценаріїв — це мова, яка інтерпретує сценарії під час виконання. Скрипти зазвичай вбудовуються в інші програмні середовища. Метою скриптів зазвичай є підвищення продуктивності або виконання рутинних завдань для програми. Скрипти на стороні сервера інтерпретуються на сервері, тоді як сценарії на стороні клієнта інтерпретуються клієнтською програмою.

Файл PHP також може містити такі теги, як HTML, і клієнтські скрипти, такі як JavaScript. HTML є додатковою перевагою під час вивчення мови PHP. Ви навіть можете вивчати PHP, не знаючи HTML, але рекомендується принаймні знати основи HTML.

Системи управління базами даних СУБД для програм, що працюють на базі даних. Для більш складних тем, таких як інтерактивні програми та веб-сервіси, вам знадобляться JavaScript та XML.

Наведена нижче блок-схема ілюструє основну архітектуру веб-програми PHP і те, як сервер обробляє запити.

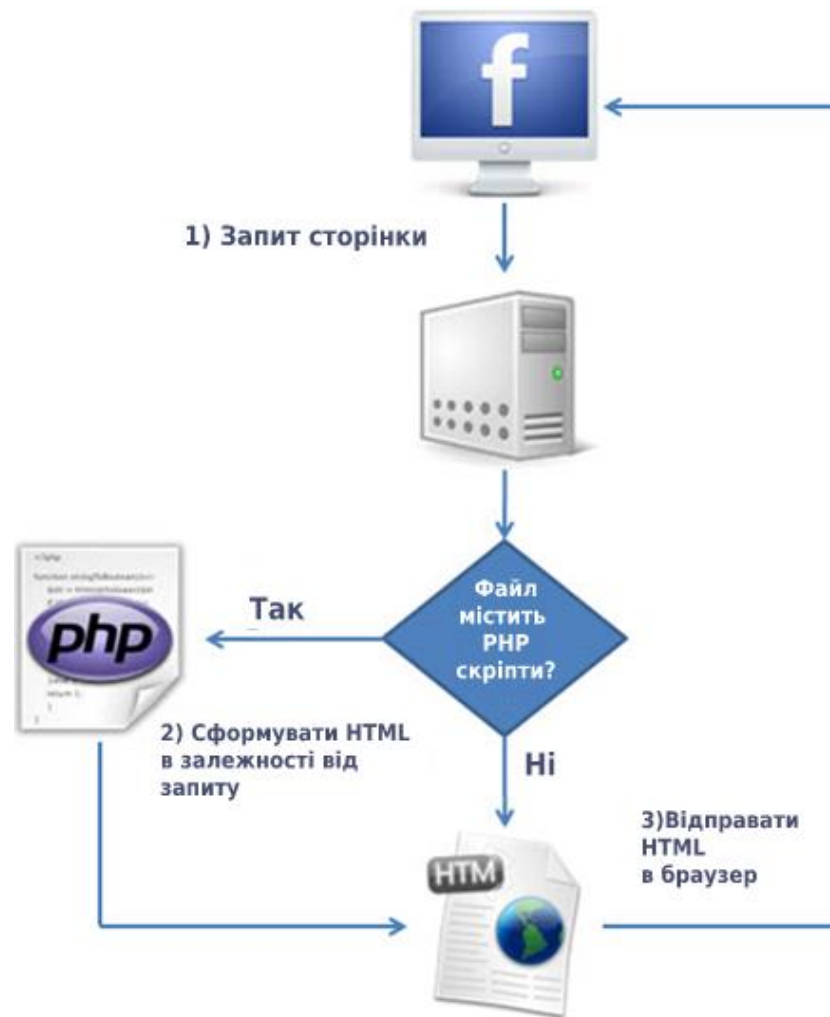


Рисунок 4.6 – Схема роботи PHP на сервері

Нижче наведено кілька причин через які можна вибрати PHP для серверу:

- PHP є відкритим вихідним кодом і безкоштовним.
- Коротка крива навчання в порівнянні з іншими мовами, такими як JSP, ASP тощо.
- Більшість серверів веб-хостингу за замовчуванням підтримують PHP, на відміну від інших мов, таких як ASP, які потребують IIS. Це робить PHP економічно ефективним вибором.

- Іншою перевагою, яку ви отримуєте від PHP, є те, що це мова сценаріїв на стороні сервера; це означає, що вам потрібно лише встановити його на сервері, а на клієнтських комп'ютерах, які запитують ресурси від сервера, не потрібно інстальювати PHP; було б достатньо лише веб-браузера.
- PHP має вбудовану підтримку для роботи рука об руку з MySQL; це не означає, що ви не можете використовувати PHP з іншими системами керування базами даних. Ви все ще можете використовувати PHP з Postgres, Oracle, MS SQL Server, ODBC тощо.
- PHP є кросплатформним.

Вибір був зроблений на користь Node.js те що він використовує синтаксис JavaScript, являється простішим, при цьому не уступаючи в функціональності, швидшим, а також на його основі простіше розробити REST API.

В якості бази данни буде використовуватися MongoDB яка являється програмою керування базами даних NoSQL з відкритим вихідним кодом. NoSQL використовується як альтернатива традиційним реляційним базам даних. Бази даних NoSQL досить корисні для роботи з великими наборами розподілених даних. MongoDB — це інструмент, який може керувати документально-орієнтованою інформацією, зберігати або отримувати інформацію.

MongoDB використовує записи, які складаються з документів, які містять структуру даних, що складається з пар полів і значень. Документи є основною одиницею даних у MongoDB. Документи подібні до нотації об'єктів JavaScript, але використовують варіант під назвою Binary JSON (BSON). Перевага використання BSON полягає в тому, що він вміщує більше типів даних. Поля в цих документах подібні до стовпців у реляційній базі даних. Містяться значення можуть бути різних типів даних, включаючи інші документи, масиви та масиви документів, відповідно до посібника користувача MongoDB. Документи також включатимуть первинний ключ як унікальний ідентифікатор.

Набір документів називаються колекціями, які функціонують як еквівалент таблиць реляційної бази даних. Колекції можуть містити будь-які типи даних, але обмеження полягає в тому, що дані в колекції не можуть бути розподілені між різними базами даних.

4.2.2 Вибір програмних засобів для програмування трекеру

Програмування Arduino ведеться цілком через власну програмну оболонку (IDE), безкоштовно доступну на сайті Arduino, яка працює на всіх актуальних платформах (Windows, Mac OS і Linux). За допомогою неї завантаження нової програми в Arduino стає справою одного кліка, потрібно тільки лише підключити плату до комп'ютера через USB. У цій оболонці є текстовий редактор, менеджер проектів, препроцесор, компілятор і інструменти для завантаження програми в мікроконтролер. Програми для Arduino пишуться на звичайному C ++, доповненим простими і зрозумілими функціями для керування введенням / виведенням на контактах.

C ++ - об'єктно-орієнтована мова програмування, забезпечує модульність, роздільну компіляцію, обробку винятків, абстракцію даних. Так само є одним з найбільш широко використовуваних і поширених мов програмування. Використовується не тільки в розробці програмного забезпечення, але і при розробці драйверів різноманітних пристроїв.

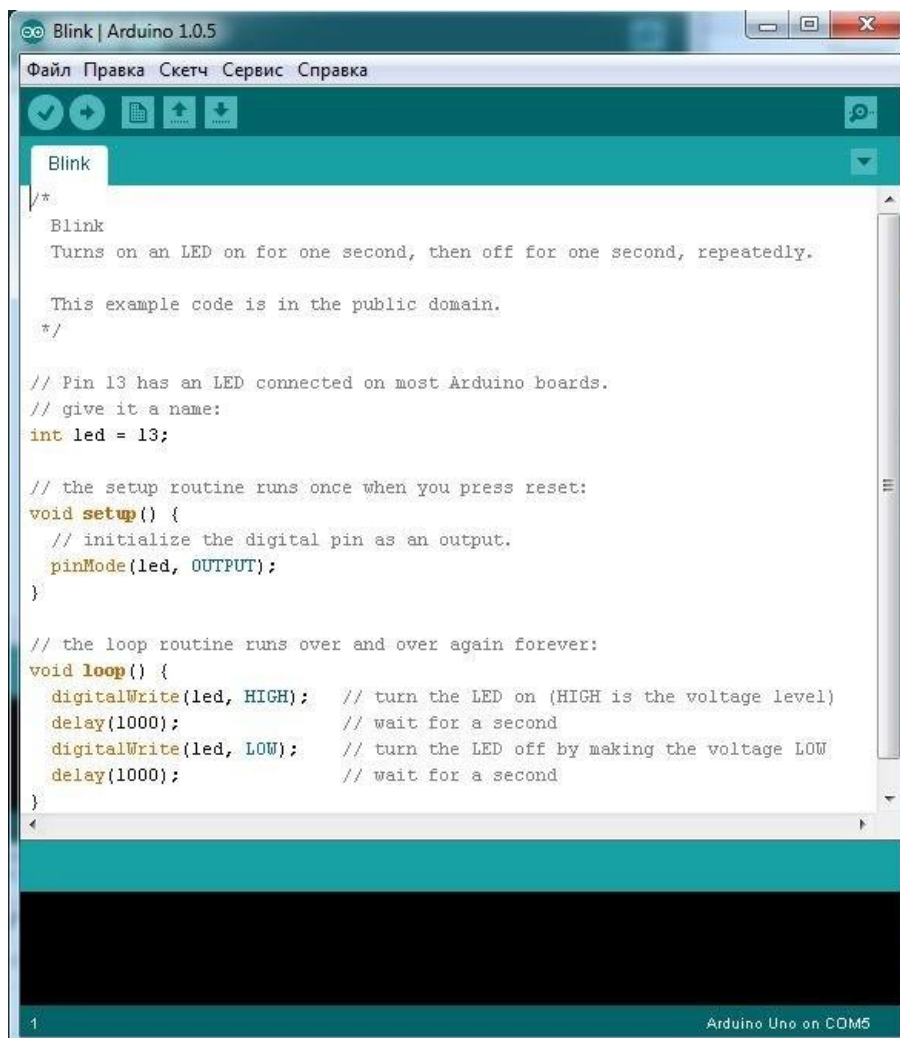
Плюсами даної програми є :

- Підсвічування коду
- Швидка заливка скетчу в плату Arduino.

- Для того, що б позначити певний пін порту як вхід вихід досить написати функцію `pinmode` (ім'я порту, `output` / `input`).
- Можна швидко привласнити номер порту до певної змінної (так би мовити спорудити маску).
- Так само можна швидко позначити стан порту (`low` або `high`).
- Є велика кількість корисних вбудованих функцій.

Мінуси:

- Робота тільки з платами Arduino.
- Відсутність `hex` файлів (точніше не зручно їх витягати з тимчасової папки).
- Необхідний певний `bootloader`, який буде розпізнає плату підключену через USB як COM-порт.
- Деякі вбудовані функції не завжди зручні, так як використовують деяку апаратну частину мікроконтролера.
- Дане середовище розробки підтримує тільки певні мікроконтролери `atmel`.
- Відсутність будь-якого віртуального симулятора.

The image shows a screenshot of the Arduino IDE software. The window title is "Blink | Arduino 1.0.5". The menu bar includes "Файл", "Правка", "Скетч", "Сервис", and "Справка". Below the menu bar is a toolbar with icons for saving, undo, redo, and other functions. The main text area contains the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

The bottom status bar shows "1" on the left and "Arduino Uno on COM5" on the right.

Рисунок 4.7 – Інтерфейс середовища програмування Arduino IDE.

5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ТРЕКІНГУ ОБ'ЄКТІВ

5.1 Схеми роботи інформаційної системи трекінгу об'єктів

Датчики, встановлені на вантажних, транспортних контейнерах, транспортних засобах або суднах, збирають телематичні дані та дані про місцезнаходження протягом усього шляху через GPS. Ці дані передаються в хмару за допомогою Wi-Fi мережі. Після того, як ці дані знаходяться в хмарі або сервері, будь-яка уповноважена сторона може отримати до них доступ з будь-якого місця та в будь-який час.

Чудова особливість IoT у логістиці полягає в тому, що він забезпечує майже в реальному часі картину розташування, стану та оточення відвантажених товарів. Можливість доступу до цих даних робить ланцюг поставок повністю прозорим. Немає необхідності чекати, поки вантаж досягне контрольного пункту, щоб переглянути його статус. В результаті постачальники можуть передбачити, виправити, а іноді й запобігти проблемам до їх ескалації завдяки набагато швидшому часу реагування. Крім того, більша прозорість означає меншу ймовірність порушення ланцюга поставок.

Нарешті, аналіз даних IoT надзвичайно корисний для оптимізації маршрутів доставки, покращення безпеки управління флотом, оцінки рівня продуктивності перевізників, аналізу вуглецевих слідів логістики та мінімізації пов'язаних з цим бізнес-ризиків.

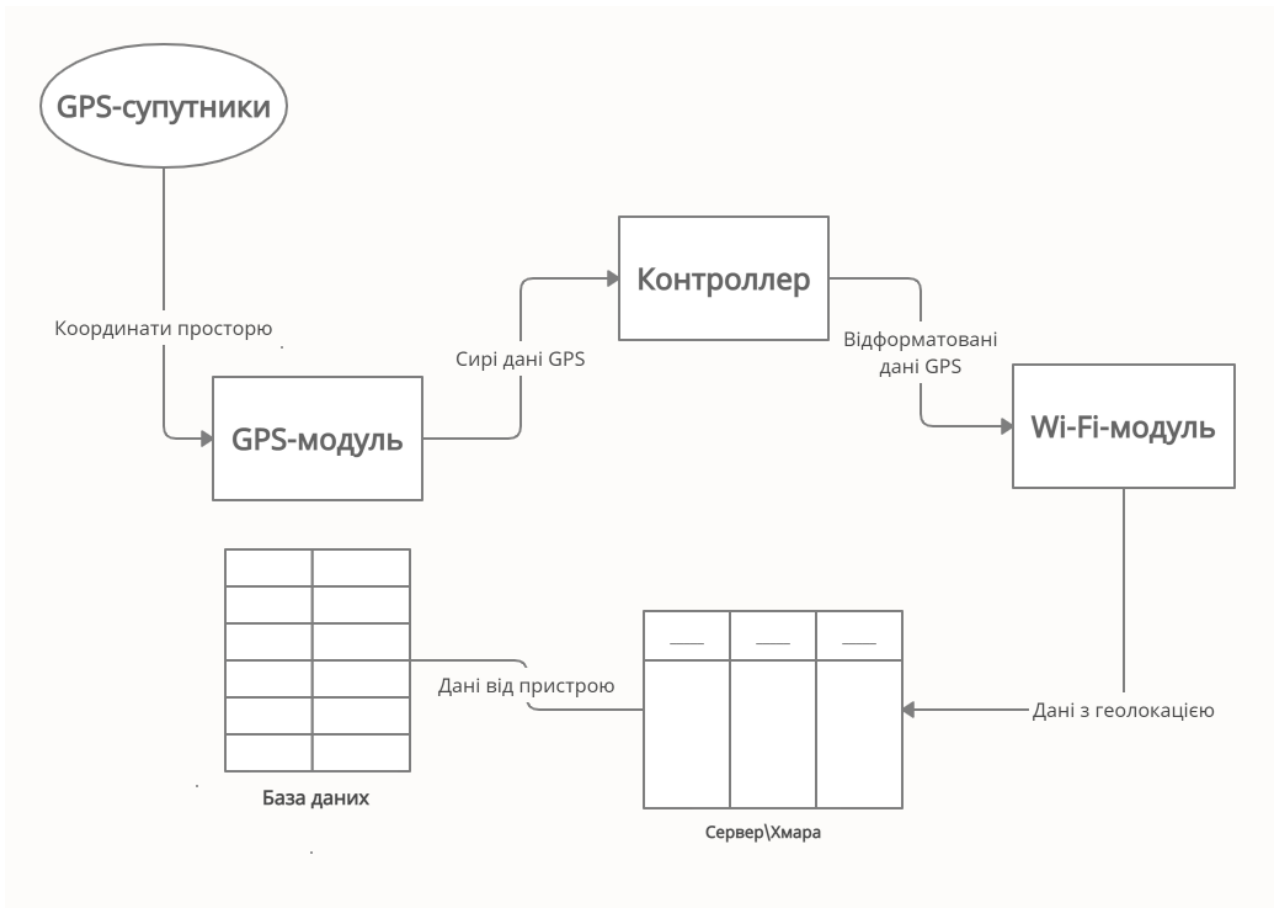


Рисунок 5.1 – Схема роботи системи

Транспортування Інтернету речей також створює більше можливостей для інших сфер, таких як експлуатація терміналів, безпека транспортування, моніторинг і маршрутизація парку, прогнозне обслуговування, управління терміном служби продукту тощо. Перехід організації від систем на основі контрольних точок вручну до безперервного потоку даних з підтримкою IoT майже відразу зробить ланцюг поставок ефективнішим і прозорим. Однак цей перехід буде нелегким.

5.2 Розробка макету GPS-трекера

Розробка макету почалась з створення схеми.

Для схеми були вибрані наступні компоненти:

- Arduino Uno – для програмування всього макету, вона буде виконувати всі обчислення та віддавати команди елементам макету.
- Wi-Fi модуль ESP8266 NodeMCU V3 – для передачі даних з GPS модуля на сервер.
- GPS модуль GPS NEO-6M – для отримання геопозиції.
- З'єднувальні кабелі – для з'єднання всіх елементів схеми.

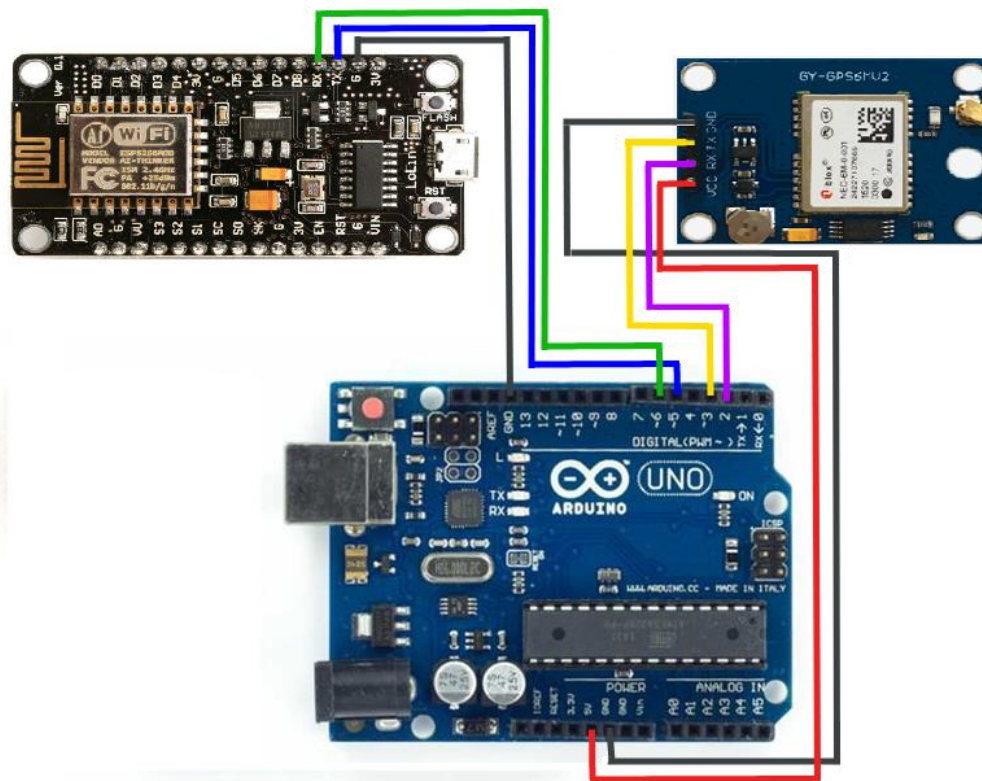


Рисунок 5.2 – Схема комутації елементів макету GPS-трекера

Алгоритм нормальної роботи схеми:

1. GPS модуль під'єднується до супутників, отримує координати, час та дату
2. GPS модуль передає дані на Arduino Uno
3. Arduino парсить дані з GPS модуля, формує їх необхідний для передачі вигляд та передає по цифровому каналу на Wi-Fi модуль
4. Wi-Fi модуль формує запит до серверу та відправляє його

Алгоритм роботи схеми при першому включенні:

1. Після включення схеми Wi-Fi модуль створить точку доступу з назвою GPS_Tracker
2. Після підключення до цієї точки доступу в браузері відкриється web-сторінка яка дозволяє під'єднатися до вибраної Wi-Fi точки
3. Необхідно натиснути кнопку Configure WiFi після чого відкриється вкладка з доступними Wi-Fi точками
4. Після вибору точки і введення паролі необхідно натиснути на кнопку Save після чого налаштування є закінченим а Wi-Fi модуль запам'ятає вибрану Wi-Fi точку
5. Почнеться нормальна робота схеми

WiFiManager

GPS_Tracker

Configure WiFi

Info

Exit

Update

Рисунок 5.3 - Перша вкладка web-сторінки для вибору Wi-Fi точки

Oksan	🔒 📶
UKrtelecom_6Q32HW	🔒 📶
umid	🔒 📶
netis_D7D376	🔒 📶
TP-Link_48A2	🔒 📶
UKrtelecom_5NC9Q9	🔒 📶
Dava	🔒 📶
DeniDo	🔒 📶

SSID

UKrtelecom_6Q32HW

Password

Save

Refresh

Рисунок 5.4 - Друга вкладка web-сторінки для вибору Wi-Fi точки

5.3 Розробка серверу для моделі

Як вже було описано вище сервер буде створено з використанням Node.js та MongoDB в якості бази даних. Він повинен мати декілька основних можливостей, а саме:

- При GET запиті без вказання ідентифікатора пристрою зчитати з БД дані всіх пристроїв та віддати їх клієнту
- При GET з вказаним ідентифікатором пристрою зчитати з БД дані цього пристрою та віддати їх клієнту
- Приймати POST запит з координатами та додатковою інформацією і записувати її до бази даних
- При DELETE запиті з вказаним ідентифікатором пристрою видалити з БД всі дані цього пристрою

Для розробки будуть використані фреймворк Express який легко дозволить створити та зконфігурувати сервер, та бібліотека mongoose яка в свою чергу дозволить спростити взаємодію з базою даних.

Почнемо з створення серверу та підключенням його до БД.

```

const express = require("express");
const mongoose = require("mongoose");
const app = express();

mongoose.connect("mongodb://localhost/gps_arduino_db",
  useNewUrlParser: true,
});
const db = mongoose.connection;

db.once("open", () => {
  console.log("Connected to Database");
}).on("error", (err) => {
  console.log(`Database connection error: ${err}`);
});
db.on("error", (err) => {
  console.log(`Database error: ${err}`);
});

```

Рисунок 5.5 - Створення інстансу сервера та підключення його до БД

Далі необхідно визначити заголовки запитів які сервер буде приймати.

```

app.use((req, res, next) => {
  res.append("Access-Control-Allow-Origin", ["*"]);
  res.append("Access-Control-Allow-Methods", "GET,PUT,POST,DELETE");
  res.append("Access-Control-Allow-Headers", "Content-Type");
  next();
});

```

Рисунок 5.6 - Визначення заголовків запитів

Важливою частиною являється обробки всіх помилок які можуть трапитися та відображення їх клієнту.

```
app.use((req, res, next) => {
  const error = new Error("Some Error");
  error.status = 404;
  next(error);
});

// Handle all error in app
app.use((error, req, res, next) => {
  res.status(error.status || 500);
  res.json({
    error: {
      message: error.message,
    },
  });
});

app.listen(PORT, () => {
  console.log("Server has been started...");
});
```

Рисунок 5.7 - Обробки помилок

Для взаємодії з БД визначасмо схему за якою повинні записуватися дані, та колекцію в чку вони будуть записані.


```

const mongoose = require("mongoose");

const gpsCoordsModelSchema = new mongoose.Schema({
  lat: {
    type: Number,
    required: true,
  },
  lon: {
    type: Number,
    required: true,
  },
  alt: {
    type: Number,
    required: true,
  },
});

const gpsTrackerDataModelSchema = new mongoose.Schema(
  {
    deviceId: {
      type: String,
      required: true,
    },
    date: {
      type: String,
      required: true,
    },
    time: {
      type: String,
      required: true,
    },
    coords: {
      type: gpsCoordsModelSchema,
      required: true,
    },
  },
  { collection: "gps_tracker_data" }
);

module.exports = mongoose.model("gpsTrackerData", gpsTrackerDataModelSchema);

```

Рисунок 5.8 - Модель даних для запису в БД

Для подальшої роботи необхідно визначити шлях за яким будуть робитися всі запити, в нашому випадку він буде виглядати як `http://192.168.0.108:8000/gps` в ньому вказаний IP-адрес сервера та шлях за яким і буде оброблятися запити.

Першим запитом який буде обробляться буде GET запиті без вказання ідентифікатора пристрою. З БД зчитуються всі записи які відповідають схемі яка була описана вище і увідпові при успішному зчитанні клієнт отримає JSON об'єкт з

даними всіх пристроїв які є в базі даних, а в разі помилки повідомлення з помилкою яка відбулась.

```
const express = require("express");
const router = express.Router();
const GpsTrackerData = require("../models/gpsTrackerData");

// Get all GPS tracker devices
router.get("/", async (req, res) => {
  try {
    const gpsTrackerData = await GpsTrackerData.find();
    res.status(200).json({
      status: "ok",
      result: { gpsTrackerData },
    });
  } catch (err) {
    res.status(500).json({ status: "error", message: err.message });
  }
});
```

Рисунок 5.9 - Обробка GET запиту без вказання ідентифікатора пристрою

Другим запитом який буде оброблятися буде GET запиті з вказання ідентифікатора пристрою. Посилання при цьому буде виглядати як <http://192.168.0.108:8000/gps>. З БД зчитуються дані пристрою ідентифікатор якого був вказаний увідпові при успішному зчитанні клієнт отримає JSON об'єкт з даними всіх пристроїв які є в базі даних, а в разі помилки повідомлення з помилкою яка відбулась.

```

// Get all GPS tracker devices
router.delete("/:id", getGpsTrackerByQueryDeviceId, async (req, res) => {
  try {
    await res.gpsTrackerData.remove();
    res.status(204).json({
      status: "ok",
      result: { message: "Deleted" },
    });
  } catch (err) {
    res.status(500).json({ status: "error", message: err.message });
  }
});

async function getGpsTrackerByQueryDeviceId(req, res, next) {
  let gpsTrackerData = null;
  try {
    gpsTrackerData = await GpsTrackerData.findOne({
      deviceId: req.params.id,
    });
    if (!gpsTrackerData) {
      return res
        .status(404)
        .json({ status: "error", message: "Can not find device by this ID" });
    }
  } catch (err) {
    return res.status(500).json({ status: "error", message: err.message });
  }

  res.gpsTrackerData = gpsTrackerData;
  next();
}

// Get all GPS tracker devices
router.get("/:id", getGpsTrackerByQueryDeviceId, async (req, res) => {
  res.send(res.gpsTrackerData);
});

```

Рисунок 5.10 - Обробка GET запиту з вказання ідентифікатора пристрою

Третім запитом який буде оброблятися буде POST запит з яким буде приходити і дані пристрою, з цих даних візьметься ідентифікатора пристрою і якщо такий пристрій вже існує в БД то його дані оновляться, якщо такого пристрою не існує то він створиться. У відповідь при успішному зчитанні клієнт отримає JSON об'єкт з даними які записалися, а в разі помилки повідомлення з помилкою яка відбулась.

```

// Create new GPS tracker device
router.post("/", async (req, res) => {
  const reqBody = req.body;

  let gpsTrackerData = null;

  try {
    // search for existing device
    gpsTrackerData = await GpsTrackerData.findOne({
      deviceId: reqBody.deviceId,
    });

    if (gpsTrackerData) {
      // update existing device data
      gpsTrackerData.deviceId = reqBody.deviceId;
      gpsTrackerData.date = reqBody.date;
      gpsTrackerData.time = reqBody.time;
      gpsTrackerData.coords = reqBody.coords;
      const updatedGpsTrackerData = await gpsTrackerData.save();
      res.status(200).json({
        status: "ok",
        result: { updatedGpsTrackerData },
      });
    } else {
      // create new device data and save
      const newGpsTrackerData = new GpsTrackerData({
        deviceId: reqBody.deviceId,
        date: reqBody.date,
        time: reqBody.time,
        coords: {
          lat: reqBody.coords.lat,
          lon: reqBody.coords.lon,
          alt: reqBody.coords.alt,
        },
      });
      gpsTrackerData = await newGpsTrackerData.save();
      res.status(201).json({
        status: "ok",
        result: { gpsTrackerData },
      });
    }
  } catch (err) {
    res.status(400).json({ status: "error", message: err.message });
  }
});

```

Рисунок 5.12 - Обробка POST запиту

Чвертим запитом який буде оброблятися буде DELETE запит для нього також потрібен ідентифікатора пристрою. при успішному виконанні з бази даних видалиться запис а клієнт отримає повідомлення про успішне виконання операції, а в разі помилки повідомлення з помилкою яка відбулась.

```
// Get all GPS tracker devices
router.delete("/:id", getGpsTrackerByQueryDeviceId, async (req, res) => {
  try {
    await res.gpsTrackerData.remove();
    res.status(204).json({
      status: "ok",
      result: { message: "Deleted" },
    });
  } catch (err) {
    res.status(500).json({ status: "error", message: err.message });
  }
});
```

Рисунок 5.12 - Обработка DELETE запит

ВИСНОВКИ

Інтернет речей у транспорті та логістиці відкриває безліч можливостей для всієї галузі. Це забезпечує такі бажані безперервні потоки даних, яких неможливо досягти за допомогою традиційних систем, а в поєднанні з машинним навчанням і штучним інтелектом, IoT може зробити ланцюжки поставок повністю прозорими та ефективними, як ніколи раніше. Загалом, переваги впровадження транспортної екосистеми IoT незаперечні.

У роботі було проаналізовано можливі архітектури та топології, використовувані для поєднання компонентів системи, визначені їх переваги на недоліки. Досліджено протоколи бездротової передачі даних, які можливі для використання в моделі інформаційної системи трекінгу об'єктів, визначені їх основні характеристики та особливості, визначено, для якої архітектури найбільше підходить кожен протокол. Проаналізовано технології та методи, які дозволяють визначити геолокацію пристрою, їх особливості та сфери використання. Представлено необхідні компоненти для побудови GPS-трекеру, за що відповідає кожен компонент та які функції він виконує, та надана їх коротка характеристика. Наведено коротку характеристику програмного забезпечення, в якому було розроблено макет трекеру, а також технології та мови програмування, які найкраще підходять для створення серверу в цій моделі.

Були розроблені макет GPS-трекеру, який дозволяє отримати його геолокацію та передати її на сервер, та сервер, який спроможний отримувати від пристроїв цю інформацію, записувати її в базу даних та віддавати її по запиті.

ПЕРЕЛІК ПОСИЛАНЬ

1. Інтернет речей у логістиці: практичне застосування технології та перспективи розвитку, вибір обладнання та програмного забезпечення [Електронний ресурс]: <https://itob.ru/blog/internet-veshchey-v-logistike-prakticheskoe-primenenie-tekhnologii-i-perspektivy-razvitiya-vybor-obo/>
2. Інтернет речей у логістиці [Електронний ресурс] : <https://apni.ru/article/1617-internet-veshchey-v-logistike>
3. Making Information Useful: How to Create a Continuous Data Flow in IoT Transportation [Електронний ресурс] : <https://intellias.com/iot-in-transportation/>
4. How IoT in Logistics Impacts Tracking, Warehousing and Fleet Management [Електронний ресурс] : <https://container-xchange.com/blog/iot-in-logistics/>
5. ЯК ІНТЕРНЕТ РЕЧІВ ВИКОРИСТОВУЄ BIG DATA: АРХІТЕКТУРА ІОТ-СИСТЕМ [Електронний ресурс] : <https://www.bigdataschool.ru/blog/iot-architecture-big-data.html>
6. Розумний Дім - обладнання та автоматика для дому. [Електронний ресурс] : <https://smarton.com.ua/novosti-umnogo-doma/tpost/ktu5fma9m1-umnii-dom-oborudovanie-i-avtomatika-dlya>
7. Топології мереж [Електронний ресурс] : <https://www.sites.google.com/site/informtexxim/home/5>
8. Bluetooth: технологія та її застосування [Електронний ресурс] : <https://www.ixbt.com/mobile/review/bluetooth-2.shtml>
9. Bluetooth Low Energy: докладний гайд для початківців [Електронний ресурс] : <https://habr.com/ru/post/532298/>
10. LPWAN [Електронний ресурс] : <https://ru.wikipedia.org/wiki/LPWAN>

- 11.LPWAN та інші бездротові технології [Електронний ресурс] :
https://controlengrussia.com/internet-veshhej/lpwan_iot/
- 12.Z-Wave explained: What is Z-Wave and why is it important for your smart home?
[Електронний ресурс] : <https://www.the-ambient.com/guides/zwave-z-wave-smart-home-guide-281>
- 13.Z-Wave [Електронний ресурс] :
<https://internetofthingsagenda.techtarget.com/definition/Z-Wave>
- 14.What is a WiFi Technology & How Does It Work? [Електронний ресурс] :
<https://www.elprocus.com/how-does-wifi-technology-work/>
- 15.How WiFi Works [Електронний ресурс] :
<https://computer.howstuffworks.com/wireless-network.htm>
- 16.WiFi Standards 802.11a/b/g/n vs. 802.11ac: Which is Best? [Електронний ресурс] :
<https://www.semiconductorstore.com/blog/2014/WiFi-standards-802-11a-b-g-n-vs-802-11ac-Which-is-Best/806/>
- 17.ЩО НАДАЮТЬ СОБОЮ РІШЕННЯ ДЛЯ ВИЗНАЧЕННЯ МІСЦЕЗНАХОДЖЕННЯ? [Електронний ресурс] :
<https://www.zebra.com/ru/ru/solutions/intelligent-edge-solutions/rtls/what-are-location-solutions.html>
- 18.What is GPS? [Електронний ресурс] : <https://www.geotab.com/blog/what-is-gps/>
- 19.LBS BASE STATION POSITIONING AND GPS SATELLITE POSITIONING COMPARISON [Електронний ресурс] : <https://www.roadragon.com/news/lbs-base-station-positioning-and-gps-satellite-positioning-comparison>
- 20.What is a REST API? [Електронний ресурс] :
<https://www.redhat.com/en/topics/api/what-is-a-rest-api>
- 21.MongoDB [Електронний ресурс] :
<https://searchdatamanagement.techtarget.com/definition/MongoDB>

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)