

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЗОВАНИХ СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка системи управління парковками у місті та оптимізації використання паркомісць за допомогою IoT»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

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

(підпис)

Данило МИХАЙЛОВ
Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр. ІСД- 41
Данило МИХАЙЛОВ
Ім'я, ПРІЗВИЩЕ

Керівник: Анастасія КАЗНАЧЕСЬВА
Ім'я, ПРІЗВИЩЕ
*науковий
ступінь,
вчене звання*

Рецензент: _____
Ім'я, ПРІЗВИЩЕ
*науковий
ступінь,
вчене звання*

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти бакалавр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІПЗАС

_____ Каміла СТОРЧАК

« _____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Михайлова Данила Володимировича**

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Розробка системи управління парковками у місті та оптимізації використання паркомісць за допомогою IoT

керівник кваліфікаційної роботи Анастасія КАЗНАЧЕСВА викладач

(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36

2. Строк подання кваліфікаційної роботи «31» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи:

1. Науково-технічна література з теми бакалаврської роботи стосовно систем управління паркуванням з використанням IoT.
2. Принцип функціонування «розумного міста» та інтеграція в нього систем управління паркуванням.
3. Основні принципи управління парковками, включаючи методи оптимізації та використання технологій IoT.

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

1. Технології Інтернету речей (IoT) для розробки системи управління паркуванням.

2. Протоколи та стандарти зв'язку для IoT-пристроїв (LoRaWAN, MQTT тощо).
 3. Розробка апаратного забезпечення для системи управління паркуванням з використанням мікроконтролерів та сенсорів.
 4. Розробка програмного забезпечення для збору, обробки та передачі даних з датчиків.
 5. Інтеграція системи управління паркуванням з хмарними сервісами та мобільними додатками.
 6. Аналіз ефективності запропонованої системи та оцінка її впливу на оптимізацію використання паркомісць.
5. Ілюстративний матеріал: *презентація*

6. Дата видачі завдання: «27» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	римітка
	Аналіз наявної науково-технічної літератури	27.02-05.03.2024	
	Обґрунтування актуальності роботи	06.03-11.03.2024	
	Аналіз існуючих систем управління паркуванням та їх оптимізації за допомогою IoT.	12.03-27.03.2024	
	Дослідження технологій та протоколів зв'язку для IoT-пристроїв у сфері паркування.	28.03-10.04.2024	
	Розробка апаратного забезпечення для системи управління паркуванням з використанням мікроконтролерів та сенсорів. Розробка програмного забезпечення для збору, обробки та передачі даних з датчиків паркування.	11.04-15.05.2024	
	Оформлення роботи: вступ, висновки, реферат	16.05-22.05.2024	
	Розробка демонстраційних матеріалів	23.05-24.05.2024	

Здобувач(ка) вищої освіти

_____ (підпис)

Данило МИХАЙЛОВ
(Ім'я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи

_____ (підпис)

Анастасія КАЗНАЧЕСВА
(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 63 с., 39 рис., 3 табл., 32 джерела.

IoT, СИСТЕМА УПРАВЛІННЯ ПАРКУВАННЯМ, RFID, ESP8266, NODEMCU 8266, C++, УЛЬТРАЗВУКОВИЙ ДАТЧИК, ІНФРАЧЕРВОНИЙ ДАТЧИК, СЕРВОПРИВІД, FIREBASE, ARDUINO, PLATFORMIO

Об'єкт дослідження: процес розробки та впровадження систем управління паркуванням у містах з використанням технологій Інтернету речей (IoT).

Предмет дослідження: методи та засоби оптимізації використання паркомісць за допомогою IoT, включаючи апаратні та програмні компоненти системи.

Мета роботи: дослідження проблеми паркування у містах, аналіз існуючих рішень, розробка та реалізація прототипу системи управління паркуванням з використанням IoT для оптимізації використання паркомісць.

Методи дослідження: методи аналізу та синтезу, моделювання, проектування, програмування, тестування та аналіз ефективності системи.

У роботі розглянуто:

1) Проблему паркування у містах та необхідність впровадження систем управління паркуванням.

2) Роль та можливості технологій IoT у вирішенні проблеми паркування.

3) Методи та принципи оптимізації використання паркомісць.

4) Принципи роботи та складові системи управління паркуванням з використанням IoT.

5) Розробку апаратного забезпечення системи, включаючи вибір компонентів та схему підключення.

Розробку програмного забезпечення для збору, обробки та передачі даних з датчиків, а також для взаємодії з хмарними сервісами.

6) Тестування та аналіз ефективності розробленої системи.

Результати роботи: розроблено прототип системи управління паркуванням з використанням IoT, який дозволяє збирати дані про стан паркувальних місць, передавати їх на хмарний сервер та надавати інформацію користувачам про наявність вільних місць. Система має потенціал для покращення ефективності використання паркувальних місць, зменшення заторів та покращення екологічної ситуації у містах.

ABSTRACT

Text part of bachelor's thesis: 63 p., 39 fig., 3 tables, 32 sources.

IoT, PARKING MANAGEMENT SYSTEM, RFID, ESP8266, NODEMCU 8266, C++, ULTRASONIC SENSOR, INFRARED SENSOR, SERVO, FIREBASE, ARDUINO, PLATFORMIO

Object of research: the process of developing and implementing parking management systems in cities using the Internet of Things (IoT) technologies.

Subject of research: methods and means of optimizing the use of parking spaces using IoT, including hardware and software components of the system.

Purpose of the work: to study the problem of parking in cities, analyze existing solutions, develop and implement a prototype of a parking management system using IoT to optimize the use of parking spaces.

Research methods: methods of analysis and synthesis, modeling, design, programming, testing and analysis of system efficiency.

The work considered:

The problem of parking in cities and the need to implement parking management systems.

The role and capabilities of IoT technologies in solving the parking problem.

Methods and principles of optimizing the use of parking spaces.

Principles of operation and components of a parking management system using IoT.

Development of system hardware, including component selection and connection diagram. Software development for collecting, processing and transmitting data from sensors, as well as for interacting with cloud services.

Testing and analysis of the effectiveness of the developed system. Results of the work: a prototype of a parking management system using IoT has been developed, which allows collecting data on the status of parking spaces, transmitting them to a cloud server and providing users with information about the availability of free spaces. The system has the potential to improve the efficiency of parking space use, reduce traffic jams and improve the environmental situation in cities.

ЗМІСТ

ВСТУП	9
1 ТЕХНІЧНІ ОСОБЛИВОСТІ РОЗРОБКИ ТА ЕКСПЛУАТАЦІЇ СИСТЕМ РОЗПОДІЛУ ПАРКОМІСЦЬ	10
1.1 Роль та необхідність систем управління парковками в містах	10
1.2 Вплив технологій IoT на оптимізацію використання паркомісць	13
1.3 Методи та принципи оптимізації паркування в умовах міського середовища	16
2 РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ УПРАВЛІННЯ ТА ОПТИМІЗАЦІЇ ПАРКУВАННЯ	30
2.1 Принцип дії та алгоритм роботи пристрою	30
2.2 Складові та схема підключення пристрою	31
2.3 Корпус і механічна робота пристрою	38
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ УПРАВЛІННЯ ТА ОПТИМІЗАЦІЇ ПАРКУВАННЯ	40
3.1 Налаштування середовища розробки	40
3.2 Алгоритм роботи програми	43
3.3 Лістинг програми	47
ВИСНОВОК	55
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	57

ВСТУП

Урбанізація та збільшення кількості автомобілів призводять до зростаючої проблеми паркування у містах. Недостатня кількість паркувальних місць, неефективне їх використання та складність пошуку вільного місця створюють значні незручності для водіїв, призводять до заторів та негативно впливають на екологічну ситуацію.

Технології Інтернету речей (IoT) відкривають нові можливості для вирішення проблеми паркування. Застосування датчиків, бездротових мереж та інтелектуальних систем управління дозволяє створювати "розумні" парковки, які здатні оптимізувати використання паркувальних місць, покращити процес пошуку вільного місця та забезпечити більш ефективне управління паркувальною інфраструктурою.

У цьому дипломному проекті розглядається проблема паркування у містах та пропонується рішення на основі технологій IoT. Метою проекту є розробка системи управління парковками, яка дозволить збирати дані про стан паркувальних місць, передавати їх на хмарний сервер та надавати інформацію користувачам про наявність вільних місць.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Проаналізувати існуючі системи управління паркуванням та визначити їх переваги та недоліки.
2. Дослідити технології IoT, які можуть бути використані для створення системи управління паркуванням.
3. Розробити апаратне забезпечення системи, включаючи вибір мікроконтролера, датчиків та інших компонентів.
4. Розробити програмне забезпечення для збору, обробки та передачі даних з датчиків, а також для взаємодії з хмарним сервером.
5. Провести тестування та налагодження системи.
6. Оцінити ефективність запропонованої системи та її вплив на оптимізацію використання паркомісць.

Очікується, що розроблена система управління паркуванням з використанням IoT дозволить значно покращити ситуацію з паркуванням у містах, зменшити час пошуку вільного місця, знизити завантаженість доріг та покращити екологічну ситуацію.

1 ТЕХНІЧНІ ОСОБЛИВОСТІ РОЗРОБКИ ТА ЕКСПЛУАТАЦІЇ СИСТЕМ РОЗПОДІЛУ ПАРКОМІСЦЬ

1.1 Роль та необхідність систем управління парковками в містах

В сучасному міському середовищі проблема паркування стає все більш актуальною і складною через зростання чисельності автотранспорту та обмеженість простору для паркування. Управління парковками у містах стає важливим елементом інфраструктури, оскільки воно спрямоване на ефективне використання доступних паркінгових майданчиків, зменшення транспортного затору, покращення безпеки дорожнього руху та зменшення негативного впливу на довкілля.

Компанія **IBM** провела дослідження в якому прийняли участь 8042 водії у 20 містах на шести континентах. Згідно цього глобального опитування водії в Найробі витрачали в середньому 31,7 хвилини на свій найдовший пошук місця для паркування, а водії в Бангалорі, Пекіні, Буенос-Айресі, Мадриді, Мехіко, Парижі та Шеньчжені, як повідомляється, значно перевищують середній світовий показник. Сімнадцять відсотків водіїв у Мілані та Пекіні та 16 відсотків водіїв у Мадриді та Шеньчжені витрачали від 31 до 40 хвилин на пошуки паркування. Фактично, більше половини всіх водіїв у 16 з 20 опитаних міст повідомили, що вони були настільки розчаровані, що покинули пошук місця для паркування та просто поїхали в інше місце. Наприклад, майже троє з чотирьох опитаних пасажирів у Шеньчжені (80 %), Пекіні (74 %), Найробі (76 %), Сінгапурі та Мехіко (73 %) і Мадриді (69 %) повідомили, що не дійшли до пункту призначення, тому що вони покинули пошук паркінгу. І навпаки, респонденти в Чикаго (63 %), Стокгольмі (62 %), Монреалі (58 %) і Торонто (57 %) рідко відчують таке розчарування.

IBM збрала результати опитування у свій перший в історії Індекс паркування, який оцінює емоційні та економічні витрати від паркування в 20 міжнародних містах, причому найбільше число є найобтяжливішим. Індекс показує широкий діапазон болю при паркуванні від міста до міста. Найменше проблем із паркуванням у досліджуваних містах зазнало Чикаго, за ним йдуть Лос-Анджелес і Торонто(рис.1.1)[2]



Рисунок 1.1 - Індекс паркування ІВМ

Індекс паркування **ІВМ** складається з таких ключових питань:

- 1) найдовша кількість часу в пошуках місця для паркування;
- 2) неможливість знайти місце для паркування;
- 3) розбіжності щодо місць паркування;
- 4) отримано паркувальний талон за незаконне паркування;
- 5) кількість отриманих паркувальних штрафів.

Міста отримали такі бали: Нью-Делі : 140; Бангалор 138; Пекін 124; Шеньчжень 122; Париж 122; Мілан 117; Найробі 111; Мадрид : 104; Сінгапур 97; Мехіко : 97; Стокгольм 90; Йоганнесбург 87; Лондон 86; Нью-Йорк 85; Монреаль 85; Буенос-Айрес : 80; Торонто : 77; Лос-Анджелес 61; і Чикаго 51.[2]

У вищезгаданому переліку міст зазначені з різними рівнями ефективності паркувальних систем, що відображається у їхніх балах. Міста з високими балами, такі як Нью-Делі та Бангалор, свідчать про проблематичну ситуацію з паркуванням, включаючи довгий час очікування та недостатність парковочних майданчиків. У той час як міста з низькими балами, як Торонто та Лос-Анджелес, демонструють кращу організацію паркувальної інфраструктури.

Система управління парковками виступає ключовим інструментом для розв'язання проблем паркування в містах. Шляхом впровадження передових технологій, таких як датчики та веб-платформи, системи управління парковками забезпечують ефективне використання доступних парковочних майданчиків, оптимізують процес пошуку місць, спрощують доступ до інформації для користувачів та сприяють зменшенню кількості порушень правил паркування.

Отже, на основі оцінки індексу паркування ІВМ та розгляду потенційних переваг систем управління парковками, можна зробити висновок про необхідність їх впровадження для поліпшення ситуації з паркуванням у містах.

Системи управління парковками є необхідними через ряд причин, показаних на рис. 1.2.



Рисунок 1.2 - Проблеми парковок в місті

1. Ефективне використання парковочного простору: Однією з основних проблем у містах є неефективне використання парковочних майданчиків через недостатню координацію та контроль за їх зайнятістю. Розумна парковка вирішує цю проблему шляхом впровадження системи датчиків та камер відеоспостереження, які надають інформацію про наявність вільних місць та дозволяють оптимізувати розподіл парковочного простору для максимальної його використаності.

2. Зменшення транспортних заторів: Пошук парковочного місця у центрі міста може призвести до транспортних заторів і створювати негативний вплив на рух автотранспорту та загальну мобільність містян. Розумна парковка надає користувачам можливість заздалегідь знаходити вільні парковочні місця через мобільні додатки або веб-платформи, тим самим зменшуючи час, який витрачається на пошук парковки, і сприяючи зменшенню транспортних заторів.

3. Підвищення безпеки парковочних зон: Неорганізоване паркування може призвести до створення небезпечних ситуацій, таких як перешкоди для руху транспорту, блокування виїздів або неправильне розташування автомобілів. Розумна парковка включає системи відеоспостереження та аналізу даних, які допомагають виявляти порушення правил паркування та забезпечувати вжиття відповідних заходів, щоб запобігти негативним наслідкам для безпеки дорожнього руху та зберегти порядок на парковочних майданчиках.

Додатково, можна зазначити, що системи управління парковками сприяють зменшенню викидів шкідливих речовин, а саме CO₂, у повітря, оскільки вони допомагають зменшити час, витрачений на пошук парковочного місця, та уникнути зайняття автомобілів у режимі простою. Це важливо для міст, що стикаються з проблемою забруднення повітря та потребою впровадження заходів для зменшення впливу автомобільного транспорту на довкілля.

Крім того, системи управління парковками можуть сприяти розвитку "зелених" технологій та інфраструктури. Наприклад, вони можуть інтегрувати зарядні станції для електромобілів або стимулювати використання велосипедів та громадського транспорту. Це сприяє переходу до більш сталого і екологічного міського транспорту, сприяючи зменшенню залежності від автомобілів з ДВЗ та покращенню якості середовища для мешканців міст.

1.2 Вплив технологій IoT на оптимізацію використання паркомісць

Вплив технологій Інтернету речей (IoT) на оптимізацію використання паркомісць визначається їхнім потенціалом у зборі, аналізі та використанні даних для покращення управління парковальною інфраструктурою. Інтеграція датчиків, вбудованих систем зв'язку та аналітичних платформ у паркову інфраструктуру створює нові можливості для збору інформації в режимі реального часу та аналізу даних про використання паркомісць (рис. 1.3 та рис.1.4).



Рисунок 1.3 – Звичайна парковка



Рисунок 1.4 – Розумна парковка

Технології IoT дозволяють збирати дані про наявність вільних та зайнятих паркомісць за допомогою датчиків руху, звуку або магнітних полів. Ці дані можуть бути надіслані в хмару для аналізу та обробки в реальному часі. Аналітика даних може виявити тенденції використання паркомісць, розпізнати пікові навантаження та недостатньо використані зони, що дозволяє оптимізувати розподіл паркомісць та підвищити ефективність їх використання.

Крім того, IoT може допомогти у впровадженні систем "розумного" керування парковкою. Ці системи можуть автоматично надсилати інформацію про наявність вільних місць водіям через мобільні додатки або електронні табло, а також надавати можливість бронювання місць. Більш того, системи управління можуть координувати рух транспорту для мінімізації часу пошуку парковки та зменшення заторів.

Згідно дослідження компанії "Liberium", яка надає технологічні рішення на основі Інтернету речей (IoT), використання цієї технології зменше обсяг трафіку на 8%, викиди газів на 40%, кілометри, пройдені автомобілем до паркування на 30% і час витрачений на паркування на 43%(рис.1.5). [1]

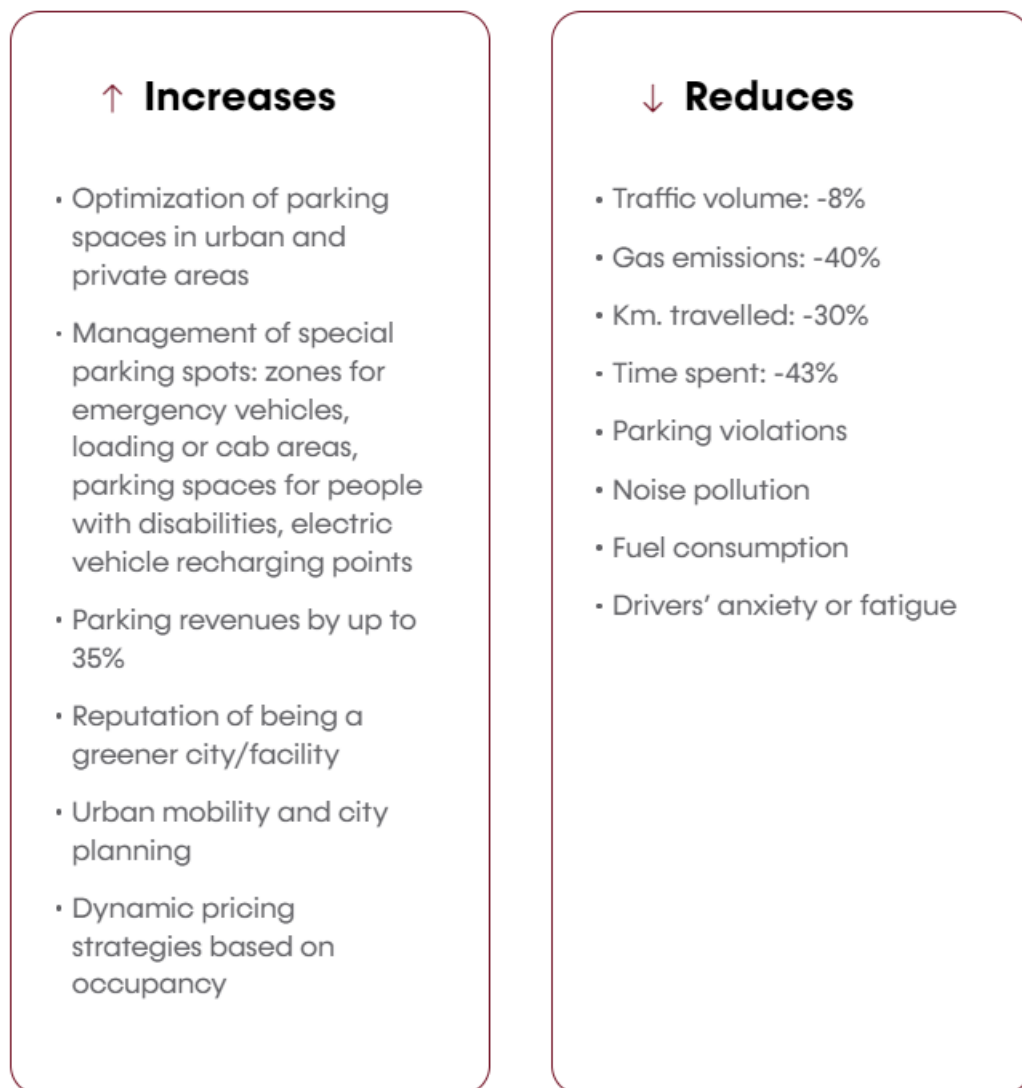


Рисунок 1.5 – Вплив системи управління парковками

Використання технологій Інтернету речей (IoT) в паркувальній інфраструктурі відображає значний вплив на різні аспекти мобільності та середовища. Дослідження компанії "Liberium", спеціалізованої на технологічних рішеннях IoT, засвідчує значні позитивні зміни в контексті обсягу трафіку, викидів газів, кількості кілометрів, пройдених автомобілем до паркування, а також часу, витраченого на процес паркування.

Перше, використання IoT дозволяє ефективно керувати рухом автотранспорту та регулювати доступ до парковочних майданчиків. Це призводить до зменшення обсягу трафіку на 8%, оскільки водії отримують доступ до актуальної інформації про вільні парковочні місця та оптимальні маршрути до них.

Друге, за рахунок оптимізації паркувального процесу та зменшення часу пошуку місця для паркування на 43%, викиди газів зменшуються на 40%. Це пояснюється тим, що менше часу, витраченого на пошук парковки, означає менше часу, коли автомобілі перебувають у режимі простою, а отже, менше споживання палива та викиди газів.

Третє, зменшення кількості кілометрів, пройдених автомобілем до паркування на 30%, також призводить до зменшення викидів газів та обсягу трафіку на дорогах, оскільки водії витрачають менше часу на проїзд до парковочних майданчиків(рис.1.6).

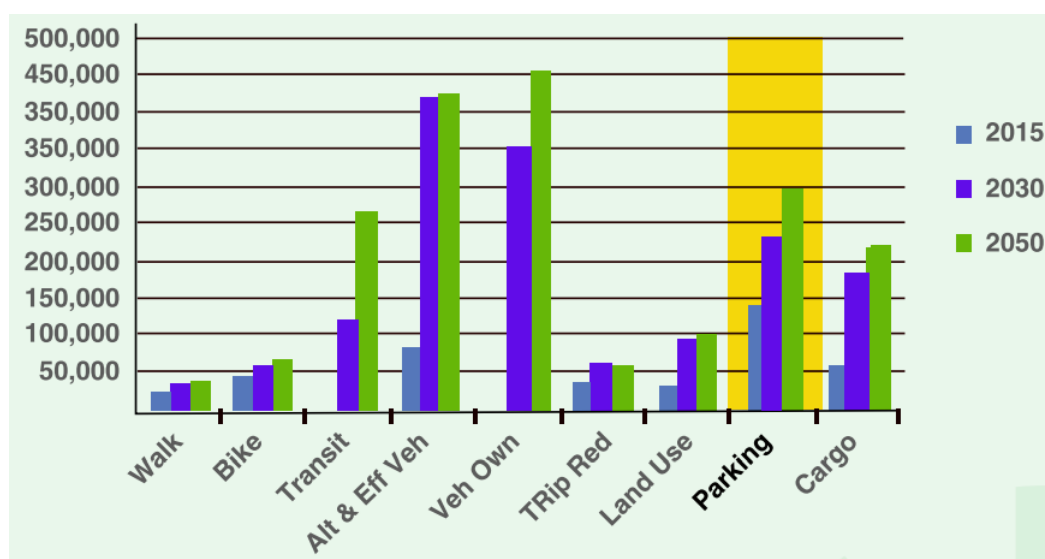


Рисунок 1.6 – Вплив системи управління парковками на зменшення викидів газу

Отже, використання технологій Інтернету речей в паркувальній інфраструктурі суттєво сприяє покращенню різних аспектів мобільності та середовища, зменшуючи обсяг трафіку, викиди газів, кількість кілометрів, пройдених автомобілем до паркування, а також час, витрачений на паркування.

1.3 Методи та принципи оптимізації паркування в умовах міського середовища

Edge Computing(рис.1.7)(також відомий як "концепція граничних обчислень" або "периферійні обчислення") - це система обробки інформації, в якій збирання та аналіз даних проводяться не в централізованому обчислювальному середовищі, такому як Центр обробки даних (ЦОД), а в місцях їхнього виникнення, наприклад, на панелях управління, датчиках чи віддалених робочих станціях.

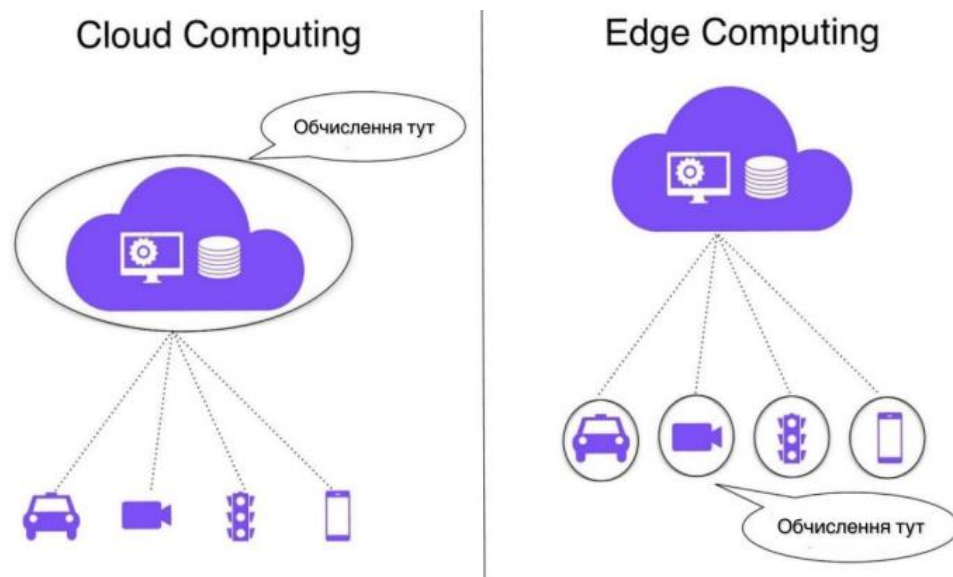


Рисунок 1.7 – Edge computing

Мета Edge Computing полягає в тому, щоб перенести обробку програм або точки надання загальної функціональності ближче до джерела дії, тобто туди, де технологія розподілених систем взаємодіє з фізичним світом. Цей підхід дозволяє знизити навантаження на мережу та ЦОД, зменшує затримки в обслуговуванні та покращує якість сервісів. Edge Computing не обов'язково потребує зв'язку з централізованою хмарою, хоча може взаємодіяти з нею за необхідності.

У більш технічному вимірі, Edge Computing можна порівняти з вимірювальним пристроєм на нафтовому родовищі або віддаленому телекомунікаційному об'єкті: вона наближає обчислювальні ресурси до місця, де збираються дані.

Edge (Fog) Computing можна розглядати як одну з форм обчислень, які відбуваються поза межами хмарового середовища, зокрема, на краї мережі, а також у точках збору даних. Головною метою цього підходу є зближення обчислювальних ресурсів та сервісів із централізованих точок хмарових технологій, таких як Центри Обробки Даних (ЦОД), до користувачів, а також до місць, де відбувається генерація даних.

Edge Computing дозволяє розміщувати додатки, дані та обчислювальні можливості в місцях, що знаходяться наближені до кінцевих користувачів, замість централізованих обчислювальних середовищ. Це допомагає оптимізувати обробку програм та функціональність, а також забезпечує зниження затримок у мережі та поліпшення якості обслуговування(рис.1.8).

From edge sensors to the centralized cloud

The edge computing ecosystem is comprised of four primary areas

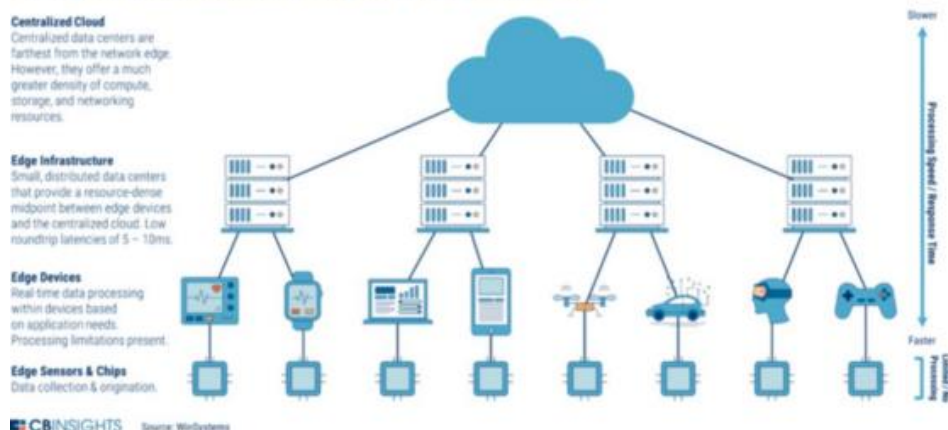


Рисунок 1.8 – Схема реалізації граничних обчислень (Edge Computing)

Методологія Edge (Fog) Computing орієнтована на наближеність до кінцевих пристроїв, географічний розподіл обчислювальних ресурсів та підтримку мобільності. Вона дозволяє перенести обчислювальні ресурси до точок, де вони найбільше потрібні, забезпечуючи при цьому більш ефективне використання мережевої пропускну здатності та знижуючи навантаження на централізовані системи хмарових обчислень.

Відмінність між Edge Computing та хмаровими обчисленнями полягає в тому, що хмарові обчислення зазвичай працюють з великими обсягами даних, в той час як Edge Computing спеціалізується на обробці "миттєвих даних", що генеруються датчиками або контрольованими системами.

Цей підхід до обчислень можна порівняти з використанням вимірювальних приладів на нафтових родовищах або віддалених телекомунікаційних об'єктах, коли обчислювальні ресурси наближаються до місця збору даних з метою оптимізації процесів обробки та аналізу інформації.

У сучасних міських умовах проблема паркування стає все більш актуальною та важливою для ефективного функціонування транспортної інфраструктури та забезпечення комфорту мешканців міст. Зростання чисельності автотранспорту, обмежений простір та недостатність паркувальних майданчиків у центральних районах міст спричиняють необхідність впровадження та вдосконалення систем управління паркуванням.

У даному розділі проведено порівняльний аналіз трьох відомих систем управління паркуванням: DEPS (Dynamic Electronic Parking System), Dahua та IPark. Кожна з цих систем розроблена з метою оптимізації процесу паркування та забезпечення ефективного використання паркувальних майданчиків у міських умовах.

Аналіз включає в себе детальне вивчення основних принципів функціонування кожної системи, їхніх переваг та недоліків. Також розглядаються можливості застосування кожної системи в конкретних міських умовах, враховуючи особливості та потреби міста.

Цей розділ спрямований на встановлення порівняльних характеристик та обґрунтування вибору найбільш ефективної системи управління паркуванням для конкретного міського середовища.

Автоматична парковка з візуалізацією проїзду

Ефективне керування рухом на паркувальних майданчиках у супермаркетах та торгових центрах визначається як одна з ключових проблем управління такими об'єктами. Пошук вільного місця для автомобіля може забирати значний час, досягаючи в середньому 20 хвилин, що негативно впливає на бажання відвідувачів здійснювати покупки на даному об'єкті. Неодноразові спроби знайти місце для паркування можуть призвести до роздратування потенційних клієнтів, приводячи до ситуації, коли дві третини водіїв готові відмовитись від візиту та вибрати інший торговий центр або супермаркет, де існує хоча б інформація на табло про наявність вільних паркувальних місць.

Невдоволення від процесу пошуку паркувального місця виростає не лише з відсутності інформації, а й з необхідності маневрувати автомобілем по паркінгу з метою знаходження місця. В цьому контексті, просто надання інформації про вільні місця на табло не завжди є достатньою мірою для вирішення проблеми. Замість цього, необхідна детальна карта проїзду, яка б дозволяла відвідувачам легко та без роздумів припаркувати свій автомобіль на час відвідування торгового об'єкту.

Вирішити проблему управління паркінгом може Інтернет речей. Автоматична парковка з візуалізацією проїзду здійснюється шляхом установки датчиків зайнятості паркувальних місць на базі технології LoRaWAN. LoRa smart parking працює таким чином: датчики LoRaWAN з магнітним сенсором встановлюються в асфальті, по радіоканалу 868 МГц передають інформацію про зайнятість місць на LoRaWAN шлюз, а ПЗ Thingsboard PE звідти відображає дані на табло чи у мобільному додатку.[3]

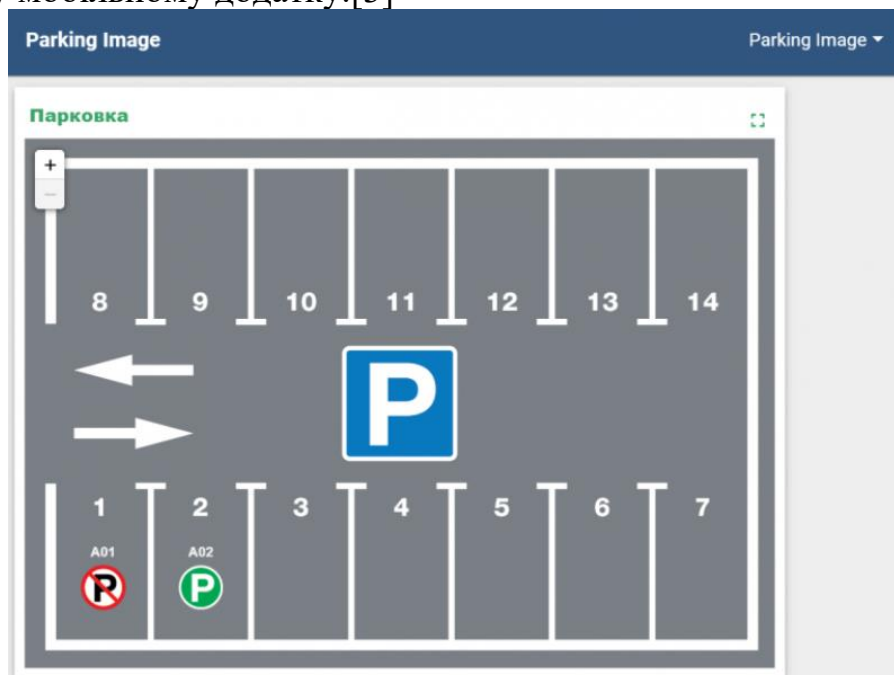


Рисунок 1.8 - Схема реалізації системи паркінгу з використанням мікроконтролерів LoRa

Технологія LoRa вирішує широкий спектр завдань, пов'язаних з передачею "телеметричної" інформації через радіоканал. Під терміном "телеметрія" розуміється передача різноманітних потоків даних, для яких достатня пропускна здатність від 0.3 до 40 кілобіт на секунду. Максимальна довжина пакета інформації обмежена 256 байтами. Більшість завдань, які вирішуються за допомогою технології телеметрії, вкладаються в ці параметри та вимоги до пропускної здатності, і навіть у випадках, коли потрібно передавати значну кількість інформації, ці обмеження вважаються задовільними.

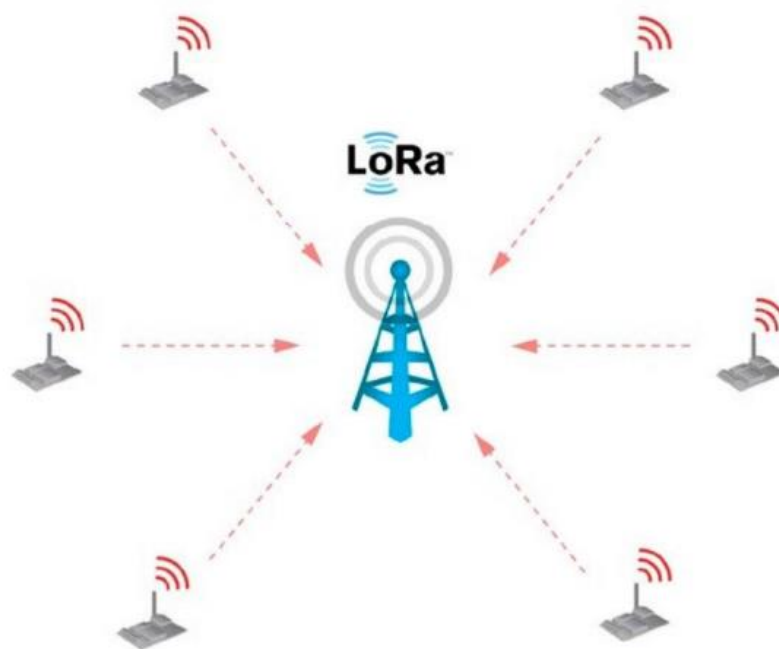


Рисунок 1.9 – Схема топології з'єднання технології LoRa

Для LoRaWAN покриття знадобиться один шлюз, до якого в подальшому можна підключити декілька тисяч різноманітних датчиків: для автоматизації освітлення, доступу до службових приміщень, обладнання протипожежної безпеки і навіть більше. Найвиразнішою особливістю лора-датчиків серед інших IoT систем можна назвати економію батареї. Через те, що більшу частину часу вони знаходяться у сплячому режимі і включаються лише для передачі невеликого пакету даних раз на визначений вами час, датчики не потребують постійного обслуговування. Контролюйте їх роботу віддалено: вмикайте, вимикайте, перепрошивайте та оновлюйте – все завдяки продуманій екосистемі на протоколі LoRaWAN.[3]

ThingsBoard PE(рис. 1.10) є платформою IoT з відкритим кодом, спрямованою на швидкий розвиток, управління та масштабування проектів Internet of Things (IoT). Основною метою цієї платформи є надання готового хмарного або локального рішення для IoT, що дозволить створити інфраструктуру на стороні сервера для ваших IoT-додатків. Вона забезпечує широкі можливості управління та

аналізу даних, забезпечуючи зручний інтерфейс для взаємодії з пристроями та збирання даних з них. ThingsBoard PE дозволяє ефективно впроваджувати та масштабувати IoT-проекти, забезпечуючи високий рівень функціональності та надійності.

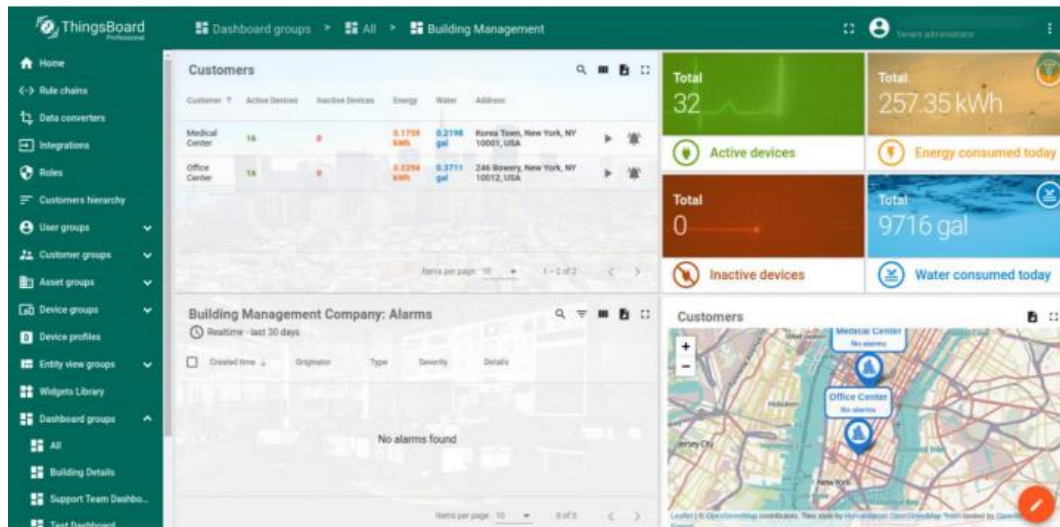


Рисунок 1.10 - Веб інтерфейс платформи ThingsBoard

Особливості платформи включають наступне:

- Управління пристроями, активами та клієнтами: Забезпечення можливості реєстрації пристроїв, активів та клієнтів у системі, а також визначення взаємозв'язків між ними.
- Збирання та візуалізація даних: Здійснення збору даних з пристроїв та активів і їх подальша візуалізація у зручному для користувача форматі.
- Аналіз телеметрії та запуск тривог: Обробка вхідної телеметрії для виконання складних аналітичних завдань та сприяння у виявленні відхилень, а також запуск тривог в разі виникнення певних подій.
- Керування пристроями через віддалені виклики процедур (RPC): Забезпечення можливості керування пристроями за допомогою віддалених викликів процедур.
- Створення робочих процесів на основі подій: Розроблення робочих процесів, які активуються на основі подій життєвого циклу пристрою, REST API або запитів RPC.
- Створення інформаційних панелей: Розроблення динамічних та адаптивних інформаційних панелей для презентації телеметрії пристроїв або активів, а також статистики користувачам.
- Налаштування ланцюжків правил: Можливість увімкнення функцій, специфічних для конкретного випадку використання, за допомогою налаштованих ланцюжків правил.
- Передача даних пристроєм в інші системи: Забезпечення можливості передачі даних, зібраних пристроєм, в інші системи для подальшої обробки чи аналізу.

Система управління паркуванням Dahua

Інноваційне рішення для паркування Dahua(рис.1.11) об'єднує ANPR і відеоаналітику, що дозволяє автоматично заїжджати/виїжджати без ручного втручання для підвищення ефективності паркування. Крім того, для покращення взаємодії з користувачем встановлено навігацію та визначення місцезнаходження автомобіля за допомогою вбудованої електронної карти. [4]

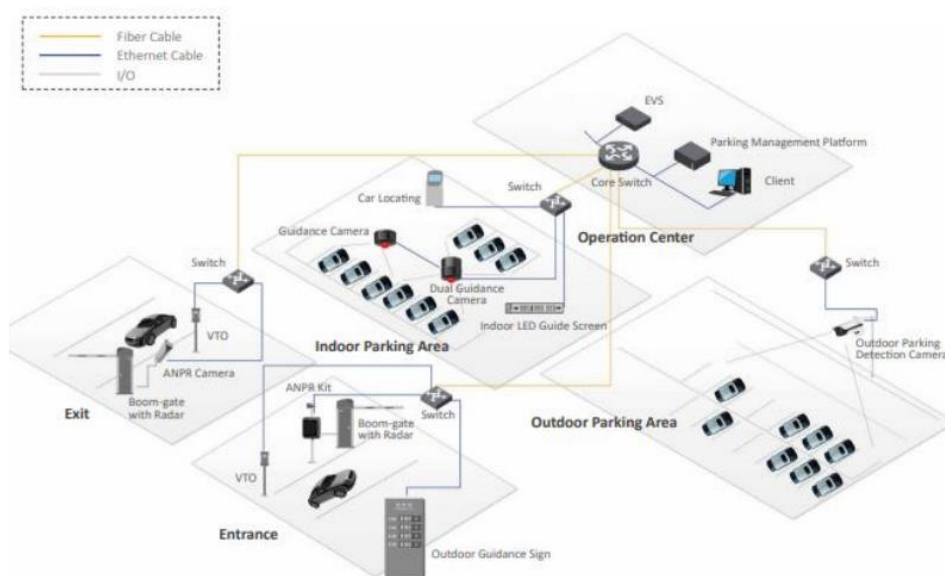


Рисунок 1.11 - Топологічно-структурна схема комунікаційної системи Dahua

Особливості рішення:

1. Легко дістатися, легко піти

- Допоможе вам знайти вільне місце для паркування
- Знайдіть свій автомобіль і сплатіть, щоб зручно виїхати

2. ANPR, відео або циклічний тригер на вибір

- Пропонуйте більш економічний вибір
- Швидкість розпізнавання вище 95%

3. Комплексне спостереження

- Забезпечте спостереження 24/7 з доступом камера та камера точкового

виявлення

- Пошук відеозапису та зображення за датою/часом/номером автомобіля

Мережеве керування

- Підтримка централізованого мережевого керування кількома паркувальними зонами та автостоянками

Відкрита система

- Відкрита система з багатьма інтерфейсами, щоб задовольнити різні вимоги клієнтів

- Простота інтеграція зі сторонніми платіжними системами

Кількість доступних паркувальних місць визначається програмним забезпеченням і відображається на зовнішньому екрані. Ця інформація відображається на багатозонному дисплеї, який дозволяє водіям чітко бачити

кількість вільних місць на світлодіодному екрані перед в'їздом на стоянку. Крім того, при в'їзді або виїзді зі стоянки водіям надається інформаційний екран з відповідною інформацією.

Dahua є впроваджувачем та розробником стандарту відеоспостереження HDCVI, який дозволяє передавати аудіо- та відеодані по коаксіальному кабелю. Цей стандарт відкриває широкі можливості для створення високоефективних систем відеоспостереження, забезпечуючи якісну передачу даних та ефективну роботу системи.

Всі ці функції досягаються за допомогою камери «3-6m Parking Entrance & Exit Camera»(рис.1.12):



Рисунок 1.12 - Камера «3-6m Parking Entrance & Exit Camera»

Її основні технічні характеристики:

- Вбудований алгоритм LPR
- Дальність захоплення 3-6 м, визначення ширини смуги 3-4 м
- Швидкість захоплення транспортних засобів $\geq 99\%$,

швидкість

розпізнавання $\geq 95\%$

- IP67, DC12V/AC24V/PoE+
- 2 МП (1920*1080), 1/2,8" CMOS, 0,002 люкс
- Моторизований варіофокальний об'єктив 3,2-10,5 мм
- 1 вбудований порт для карти TF підтримує до 128 ГБ[4]

Камера LPR True Day/Night з роздільною здатністю 2 мегапікселя дозволяє аналізувати такі низькошвидкісні процеси, такі як паркування, контроль доступу та міське вуличне спостереження. Прилад підтримує виявлення відео та розпізнає транспортний засіб в низькому положенні зі швидкістю менше 80 км/год із вбудованим алгоритмом LPR список. Ця модель надає комплексне рішення для отримання інформації з двохфокусною лінзою і потужним ІЧ-підсвічуванням на вулиці спостерігаємих цілей. Потужна система WDR (wide dynamic range) створена для середовища з прямими сонячними променями або відблисками, навіть головне світло вмикається, коли автомобіль наближається. Камера підтримує розпізнавання

петлі та відео, вона більш стабільна, коли система використовується для контролю шлагбаума або аналізу номера транспортного засобу в контролі доступу[4].

Другим варіантом відеоприладу може бути камера «Dual Parking Guidance Camera»(рис.1.13):



Рисунок 1.13 - Камера «Dual Parking Guidance Camera»

Детектор паркувальних місць може контролювати 6 паркувальних місць в режимі реального часу, ідентифікувати транспортні засоби та номерні знаки, а також відобразити стан паркувального місця з різними кольорами індикаторів (можна налаштувати 7 кольорів).

Основні характеристики:

- Виявлення в режимі реального часу 6 паркувальних місць
- 7 кольорів індикаторів (червоний/жовтий/синій/зелений/блакитний/білий/рожевий)
- Інтерактивний дизайн із зовнішнім точковим індикатором
- До 6 камер у шлейфовому мережевому кабелі для живлення та передачі
- 2 МП, об'єктив 2,8/3,6 мм
- Вбудований алгоритм LPR
- DC 12-48V [4]

Система навігації на парковці iPark

Система навігації на парковці спрямована на оптимізацію процесу знаходження вільних місць для паркування та полегшення виїзду з них з мінімальними затримками для водіїв. Вона використовує ряд датчиків та сенсорів, розташованих на паркувальному об'єкті, для систематичного моніторингу трафіку, реєстрації в'їздів та виїздів автомобілів, а також визначення завантаженості паркінгу.

Обладнання системи включає в себе фронтальні датчики різних типів, такі як Bilogy і Trilogy. Ці датчики, обладнані ультразвуковими сенсорами, забезпечують точне визначення розташування автомобілів на паркувальних місцях з високою достовірністю. Вони монтується над кожним окремим місцем для паркування на висоті від 2.1 до 3.5 метрів, забезпечуючи ефективне охоплення всієї парковки.

За допомогою цих датчиків система здійснює моніторинг наявності вільних парковочних місць та надає водіям необхідну інформацію для швидкого

знаходження місця для паркування. Це сприяє зниженню часу пошуку вільного місця, що в свою чергу підвищує пропускну спроможність парковки на 25% і сприяє зменшенню забруднення повітря чадним газом.

Окрім цього, система навігації на парковці забезпечує розподіл потоку машин в залежності від завантаженості парковки, що дозволяє збалансувати трафік та уникнути заторів. Такий підхід сприяє покращенню загального досвіду водіїв та забезпечує ефективне використання парковочного простору.

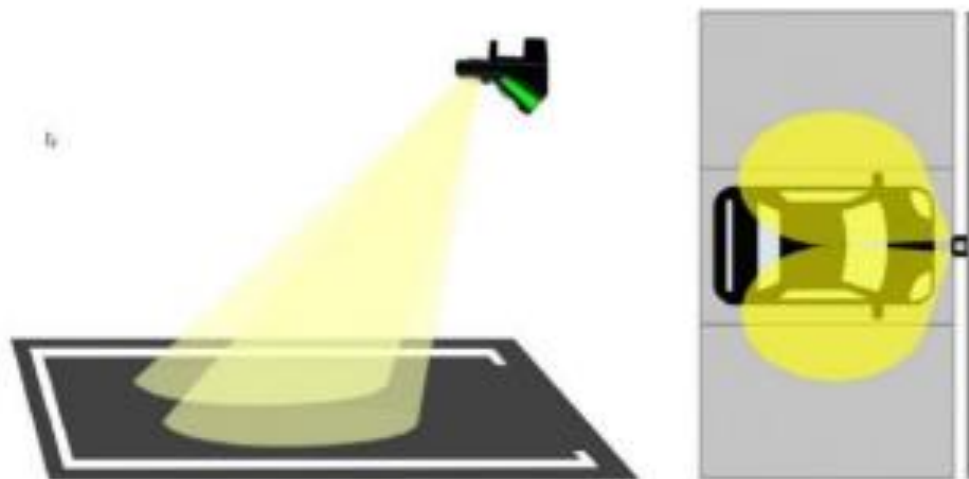


Рисунок 1.14 – Датчик Trilogy

Датчик Trilogy, відмінно від Bilogy, володіє додатковою функціональністю LED-підсвічування паркувального місця. Ця особливість дозволяє динамічно регулювати яскравість підсвічування під час паркування автомобіля, зменшуючи її після завершення паркування. Крім того, можливе вимкнення підсвічування, коли місце вільне, або, навпаки, підсвічування лише вільних місць.

Завдяки гнучкій настройці RGB LED-індикатора, можливо встановити для кожного кольору індикатора свої значення, що відображають стан паркувального місця. Наприклад:

- Червоне підсвічування - місце зайняте.
- Зелене підсвічування - місце вільне.
- Синє підсвічування - місце для інвалідів.

Кольори індикатора можна налаштовувати вручну або за певним розкладом. Це може бути корисним, наприклад, для "бронювання" місць для особливо важливих гостей, коли індикатор може бути "червоним", але місце залишається вільним. Також можна передбачити окремий колір для заброньованих місць.

Можливість налаштовувати колір LED-індикатора Trilogy та Bilogy вручну або за розкладом відкриває широкі можливості для управління паркувальним простором. Наприклад, ця функція може стати в нагоді для "бронювання" місць для особливо важливих гостей. Таким чином, індикатор може відображати червоний колір, вказуючи на те, що паркувальне місце залишається вільним, але

зарезервоване для конкретного автомобіля. Також можна встановити окремий колір для заброньованих місць, що дозволить легко відрізнити їх від інших.

Крім того, налаштовуваний колір LED-індикатора не обмежується лише червоним або зеленим. Наприклад, для різних категорій автомобілів можна встановити відмінні від цих стандартних кольори. Наприклад, для службових транспортних засобів чи великогабаритних машин можна використовувати інші кольори, такі як помаранчевий для великогабаритних машин та червоний для зайнятих місць. Це дозволяє оптимізувати розподіл паркувального простору та покращити організацію паркування на об'єкті.

Реальний приклад - у великому торгово-офісному центрі в Куала-Лумпурі, Малайзія, "рожеве" підсвічування вільного місця – паркувальна зона для жінок[5].



Рисунок 1.15 - Зовнішній вигляд фронтального датчика Trilogy

Датчики, які можна конфігурувати віддалено через програмне забезпечення CirPark Scada, відкривають широкі можливості для налаштування та управління системою паркування. За допомогою цього програмного забезпечення можна налаштувати чутливість сенсорів, програмувати роботу індикаторів або підсвічування за розкладом.

Один із часто використовуваних проектних рішень - це безкоштовний період гостьового паркування, після закінчення якого оператор отримує відповідне повідомлення. Це дозволяє ефективно використовувати паркувальний простір та забезпечує зручність для користувачів.

Центральні датчики, у порівнянні з фронтальними, мають простішу конструкцію, що призводить до зниження їх вартості та енергоспоживання, а також спрощує налаштування та управління. Ці датчики з одним ультразвуковим сенсором встановлюються над центральною частиною місця для паркування та мають двоколірний LED-індикатор стану.

У ряді CirControl доступні два типи центральних датчиків: SP3-RB з двоколірним червоно-синім індикатором та SP3-RG(рис. 1.16) з двоколірним червоно-зеленим індикатором. Ці варіанти датчиків надають можливість вибору підходящого кольорового коду для відображення стану паркувальних місць залежно від потреб та вимог конкретного паркувального об'єкту.



Рисунок 1.16 – Датчики SP3-RG та SP3-RB

Розділені датчик та індикатор присутності автомобіля. Цей варіант оснащення паркувального місця можливий у вигляді окремих датчика і сенсора. Датчики центральної установки дещо гірше видно для водіїв, внаслідок чого може незначно зростати час пошуку паркувального місця.



Рисунок 1.17 – Датчик SP3 та виносні індикатори

Ультразвуковий датчик SP3 (характеристики ті ж, як у SP3-RG\SP3-RB) встановлюється над центральною частиною місця для паркування, а індикатор виноситься на фронтальну частину. Індикатори мають червонозелене або червоно-блакитне led-підсвічування[5].

Порівняння систем паркування

Новітні технології паркування, такі як DEPS, Dahua та IPark, представляють собою системи, які впроваджуються для забезпечення ефективного управління та моніторингу паркувальних просторів. Дослідження функціональних характеристик цих систем виявляє схожість у їх базових принципах дії, але водночас розкриває їхню відмінність у ряді аспектів.

Система DEPS, розроблена для автоматизованого контролю доступу та моніторингу паркувальних зон, вирізняється широким спектром функцій, включаючи розпізнавання автомобільних номерів, інтеграцію з відеоспостереженням та системами оплати. Зокрема, DEPS надає можливість ефективного управління потоком автотранспорту, спрощуючи процеси в'їзду та виїзду.

Система Dahua пропонує подібний функціонал, однак вона акцентується на високоякісному відеоспостереженні та аналітиці, що дозволяє не лише відслідковувати рух автотранспорту, але й забезпечує захист паркувальних зон від незаконних дій.

Нарешті, система IPark спеціалізується на оптимізації управління паркувальними майданчиками через використання інноваційних технологій і Інтернету речей. Вона забезпечує віддалений моніторинг, системи резервування місць та інші зручні функції, що сприяють підвищенню ефективності використання паркувальних зон.

Для кращого уявлення про різноманіття функціоналу та характеристик цих систем була розроблена порівняльна таблиця(таблиця 1), в якій зібрана інформація про їхні можливості, технічні характеристики, а також переваги та недоліки. Такий підхід дозволяє оперативно оцінити кожен систему з точки зору відповідності конкретним потребам та умовам експлуатації.

Таблиця 1 - Порівняння характеристик систем паркування

Тип системи	DEPS	Dahua	IPark
Ціна	401\$	400-715\$	370\$
Протокол передачі даних	LoRa,LoRaWAN	Ethernet2 x10/100M self-adaptive port	Ethernet
Тип датчика виявлення т/з	Магнітний датчик	Камера	УЗ-датчики, Bilogy, Trilogy
Наявність Edge Computing	-	+	-

Висновки до Розділу 1

Актуальність дослідження в галузі систем розумного паркування, що поєднують у собі технології інтернету речей (IoT), відеоспостереження та аналітики для оптимізації управління паркувальними майданчиками – дуже велика. Аналіз цього сегмента ринку показує наростаючий інтерес до використання таких систем як DEPS, Dahua та IPark у містах та комерційних об'єктах, з огляду на їхню потенційну здатність поліпшувати розподіл паркувальних ресурсів та зменшувати транспортні затори.

Принципи роботи розглянутих систем розумного паркування виявилися схожими у певних аспектах, проте кожна система має свої унікальні особливості та функціонал, що відображається у їхній архітектурі та технічних можливостях. Наприклад, DEPS використовує LoRaWAN для збору даних про зайнятість парковочних місць та можливість інтеграції з іншими IoT-рішеннями, тоді як Dahua надає розширені можливості аналізу даних відеоспостереження та IPark спеціалізується на навігації на парковці.

Порівняльний аналіз цих систем дозволив виявити їхні переваги та обмеження в різних сценаріях застосування. Отже, прийняття рішення щодо вибору конкретної системи для конкретного застосування вимагає уважного аналізу технічних характеристик та врахування потреб користувачів.

Загалом, результати дослідження першого розділу демонструють значущість систем розумного паркування як інструмента для оптимізації управління паркувальними просторами та покращення мобільності в міських середовищах. Далі дослідження у цій галузі може спрямовуватися на вдосконалення технологічних рішень та розробку нових інновацій для подальшого підвищення ефективності та використання систем розумного паркування.

2 РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ УПРАВЛІННЯ ТА ОПТИМІЗАЦІЇ ПАРКУВАННЯ

2.1 Принцип дії та алгоритм роботи пристрою

Алгоритм має передбачати послідовність дій для забезпечення ефективного та надійного функціонування системи управління та оптимізації паркування. Починаючи зі збору даних про стан паркувальних місць, апаратний пристрій переходить до їх обробки з метою визначення кількості вільних та зайнятих місць для подальшої передачі. Опрацьовані дані надсилаються до центральної системи управління паркуванням через безпроводний чи проводний зв'язок, забезпечуючи актуальність інформації про доступність місць для паркування. Процес збору та передачі даних повторюється періодично або при зміні стану паркувальних місць з метою забезпечення постійної актуалізації інформації. Завершальним етапом алгоритму є відображення отриманої інформації про стан паркувальних місць на інформаційних табло для водіїв або інтеграція в мобільний додаток для полегшення пошуку вільних паркомісць та зручності користування системою.

Принцип дії:

Апаратний пристрій призначений для збору, обробки та передачі даних про стан парковочних місць. Його основним завданням є надання інформації про вільні та зайняті місця на паркувальному майданчику системі управління паркуванням для подальшого аналізу та оптимізації(рис.2.1).

Принцип дії апаратного пристрою для системи управління та оптимізації паркування



Рисунок 2.1 – Алгоритм та принцип дії апаратного пристрою для системи управління та оптимізації паркування

Алгоритм роботи:

1. Збір даних: Процес починається зі збору даних про стан паркувальних місць за допомогою вбудованих датчиків або сенсорів. Датчики реєструють

наявність автомобілів на майданчику та передають цю інформацію до апаратного пристрою.

2. Обробка інформації: Пристрій обробляє отримані дані для визначення кількості вільних та зайнятих парковочних місць. Ця інформація зберігається в пам'яті пристрою та підготовлюється для подальшої передачі.

3. Передача даних: Опрацьовані дані надсилаються до центральної системи управління паркуванням через безпроводний чи проводний зв'язок. Це може бути здійснено через мережу LoRaWAN, Wi-Fi, Ethernet тощо.

4. Оновлення інформації: Процес збору та передачі даних повторюється періодично або при зміні стану паркувальних місць. Це дозволяє забезпечити актуальність інформації про доступність місць для паркування.

5. Система відображення: Отримана інформація про стан паркувальних місць може бути відображена на інформаційних табло для водіїв або інтегрована в мобільний додаток для зручності користувачів.

Апаратний пристрій працює у відповідності з вищезазначеним алгоритмом, забезпечуючи надійну та ефективну роботу системи управління та оптимізації паркування.

2.2 Складові та схема підключення пристрою

Для реалізації апаратного забезпечення системи управління паркуванням пропонується наступний набір компонентів:

1. Мікроконтролер Arduino Uno(рис.2.2):

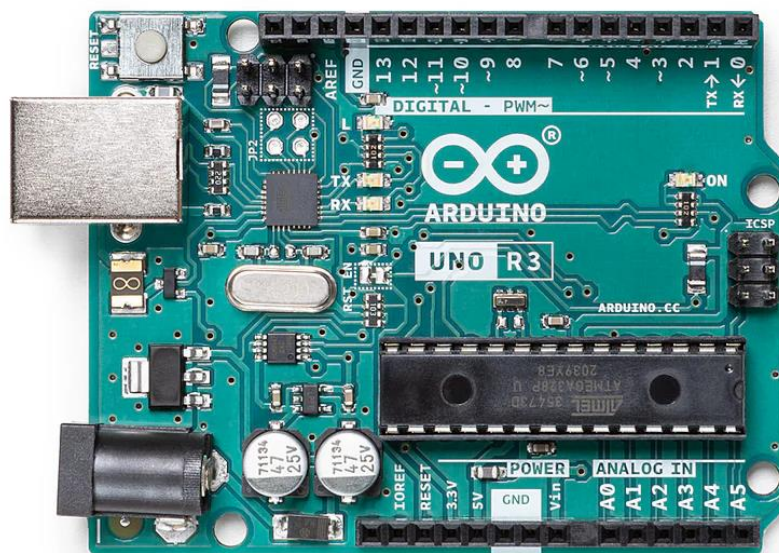


Рисунок 2.2 – Загальний вигляд Arduino Uno

Опис: Популярний вибір для проектів завдяки своїй простоті використання, доступності та широкій спільноті розробників. Забезпечує достатню

обчислювальну потужність для обробки даних з датчиків та управління периферійними пристроями.

Технічні характеристики:

- Мікроконтролер: ATmega328P
 - Робоча напруга: 5 В
 - Цифрові входи/виходи: 14 (з яких 6 можуть використовуватися як ШІМ-виходи)
 - Аналогові входи: 6
 - Flash-пам'ять: 32 КБ (ATmega328P) з яких 0.5 КБ використовується завантажувачем
 - SRAM: 2 КБ (ATmega328P)
 - EEPROM: 1 КБ (ATmega328P)
 - Тактова частота: 16 МГц
2. NodeMCU ESP8266(рис.2.3):

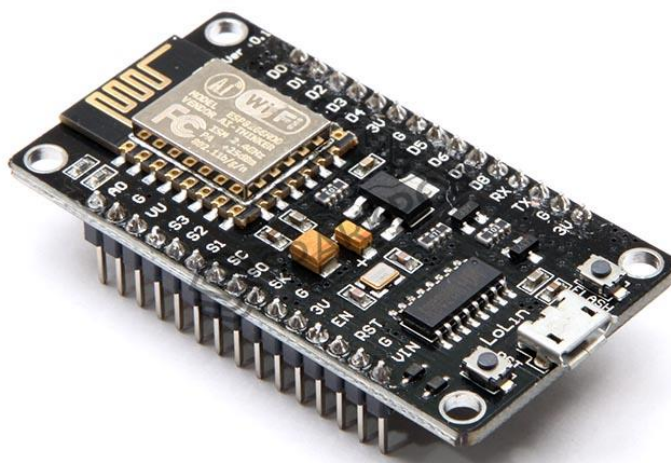


Рисунок 2.3 – Загальний вигляд NodeMCU ESP8266.

Опис: Компактний модуль з вбудованим Wi-Fi, що дозволяє легко підключати пристрої до мережі Інтернет та забезпечувати передачу даних на хмарний сервер.

Технічні характеристики:

- Мікроконтролер: ESP8266
- Робоча напруга: 3.3 В
- Цифрові входи/виходи: 11 (з яких деякі можуть використовуватися як ШІМ-виходи)
- Аналоговий вхід: 1 (максимальна напруга 3.2 В)
- Flash-пам'ять: 4 МБ
- SRAM: 80 КБ

- Wi-Fi: 802.11 b/g/n
3. RFID-модуль MFRC522(рис.2.4):



Рисунок 2.4 – Загальний вигляд RFID-модуль MFRC522 та RFID-мітки

Опис: Призначений для зчитування та запису даних на RFID-мітки. Використовується для ідентифікації користувачів та авторизації їх доступу до парковки.

Технічні характеристики:

- Робоча частота: 13.56 МГц
- Підтримувані стандарти: ISO/IEC 14443 A, MIFARE, NTAG
- Інтерфейс: SPI
- Напруга живлення: 3.3 В

4. Ультразвуковий датчик HC-SR04(рис.2.5):



Рисунок 2.5 – Загальний вигляд HC-SR04

Опис: Вимірює відстань до об'єкта за допомогою ультразвукових хвиль. Використовується для визначення наявності автомобіля на паркувальному місці.

Технічні характеристики:

- Напруга живлення: 5В

- Споживаний струм: 15мА
 - Робоча частота: 40кГц
 - Максимальна дальність вимірювання: 4 м
 - Мінімальна дальність вимірювання: 2 см
 - Кут огляду: <math><15^\circ</math>
5. Інфрачервоний датчик перешкод (наприклад, TCRT5000)(рис.2.6):

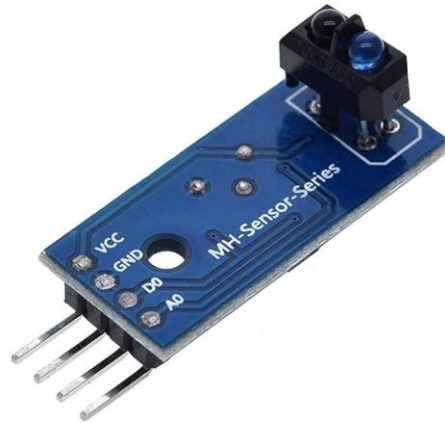


Рисунок 2.6 – Загальний вигляд TCRT5000

Опис: Виявляє наявність об'єкта перед собою за допомогою інфрачервоного випромінювання. Може використовуватися для визначення зайнятості паркувального місця або виявлення перешкод перед шлагбаумом.

Технічні характеристики:

- Напруга живлення: 5 В
 - Вихідний сигнал: цифровий (високий/низький)
 - Дальність виявлення: до 2-3 см
6. Сервопривід SG90(рис.2.7):
- 7.



Рисунок 2.7 – Загальний вигляд SG90

Опис: Керує положенням валу на заданий кут. Використовується для відкриття та закриття шлагбаумів.

Технічні характеристики:

- Напруга живлення: 4.8-6 В
 - Кут повороту: 180°
 - Споживаний струм: до 100 мА
8. Камера OV7670(рис.2.7):
- 9.



Рисунок 2.7 – Загальний вигляд OV7670

Опис: Компактна камера з можливістю передачі зображень через інтерфейс SCCB (подібний до I2C). Використовується для запису зображень автомобілів та їх номерних знаків.

Технічні характеристики:

- Роздільна здатність: до 640x480 пікселів
- Інтерфейс: SCCB
- Напруга живлення: 3.3 В

8. Світлодіодний індикатор та зумер (активний):

Опис: Використовуються для надання візуальної та звукової інформації користувачам про стан системи (наприклад, вільне/зайняте місце, відкриття/закриття шлагбауму).

Технічні характеристики:

- Світлодіод: напруга живлення 2-3 В, струм 20 мА
- Зумер: напруга живлення 5 В, струм до 50 мА

9. Блок живлення 5 В:

Опис: Забезпечує живлення всіх компонентів системи.

10. Хмарний сервер (наприклад, Firebase):

Опис: Надає сервіси для зберігання даних, аутентифікації користувачів та обміну повідомленнями між пристроями.

Принцип роботи системи(рис.2.8):

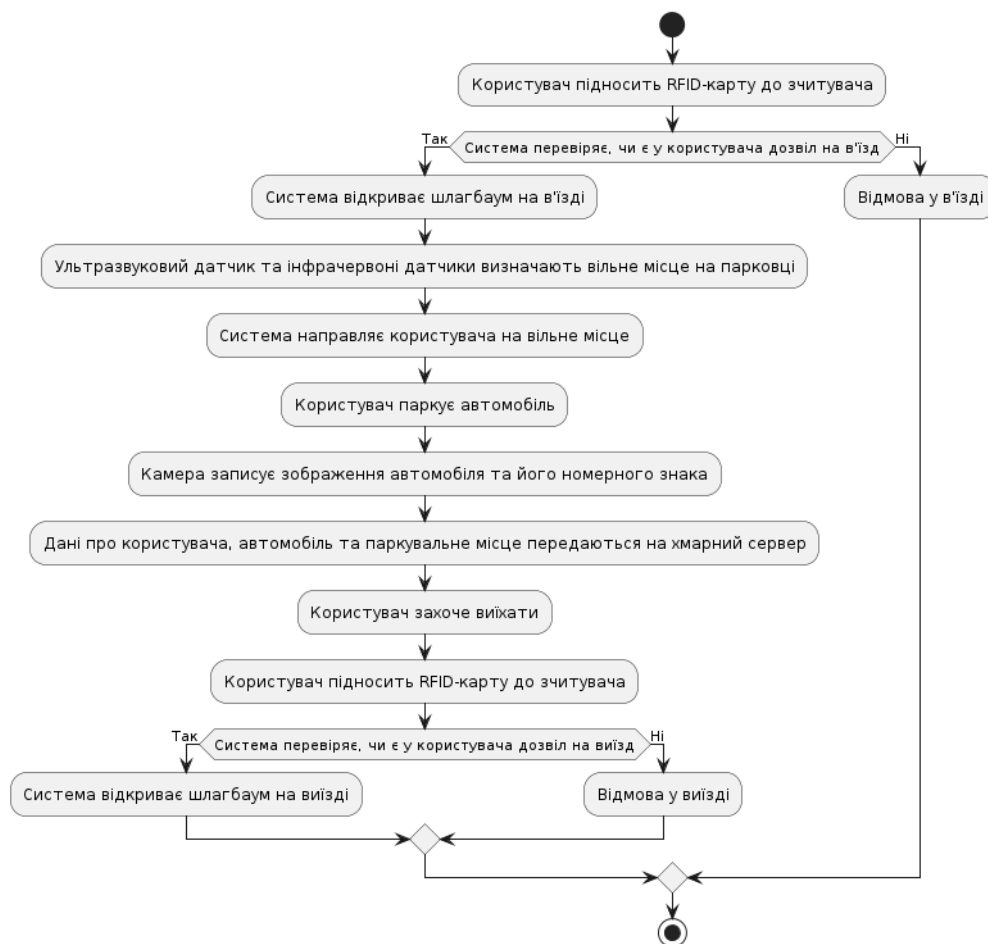


Рисунок 2.8 – Блок-схема принципу роботи системи

1. Користувач підносить RFID-карту до зчитувача.
2. Система перевіряє, чи є у користувача дозвіл на в'їзд.
3. Якщо дозвіл є, система відкриває шлагбаум на в'їзді.
4. Ультразвуковий датчик та інфрачервоні датчики визначають вільне місце на парковці.
5. Система направляє користувача на вільне місце.
6. Користувач паркує автомобіль.
7. Камера записує зображення автомобіля та його номерного знака.
8. Дані про користувача, автомобіль та паркувальне місце передаються на хмарний сервер.
9. Коли користувач захоче виїхати, він знову підносить RFID-карту до зчитувача.
10. Система перевіряє, чи є у користувача дозвіл на виїзд.
11. Якщо дозвіл є, система відкриває шлагбаум на виїзді.

На рисунку 2.9 показана блок-схема, яка ілюструє схему підключення системи. Процес починається з моменту, коли користувач підносить свою RFID-

карту до зчитувача. Система зчитує ідентифікатор картки та перевіряє його наявність у базі даних користувачів, які мають дозвіл на в'їзд. Якщо картка ідентифікована та дозвіл підтверджено, система подає сигнал на відкриття шлагбаума, дозволяючи автомобілю в'їхати на територію парковки.

Після в'їзду, система за допомогою ультразвукових та інфрачервоних датчиків визначає наявність вільних паркувальних місць. Визначивши вільне місце, система надає водієві інструкції щодо його розташування, наприклад, за допомогою світлових індикаторів або дисплеїв.

Коли автомобіль займає паркувальне місце, камера фіксує його зображення та номерний знак. Ці дані, разом з інформацією про користувача та час в'їзду, передаються на хмарний сервер для зберігання та подальшого аналізу.

Процес виїзду відбувається аналогічно. Користувач підносить RFID-карту до зчитувача, система перевіряє дозвіл на виїзд та, у разі підтвердження, відкриває шлагбаум.

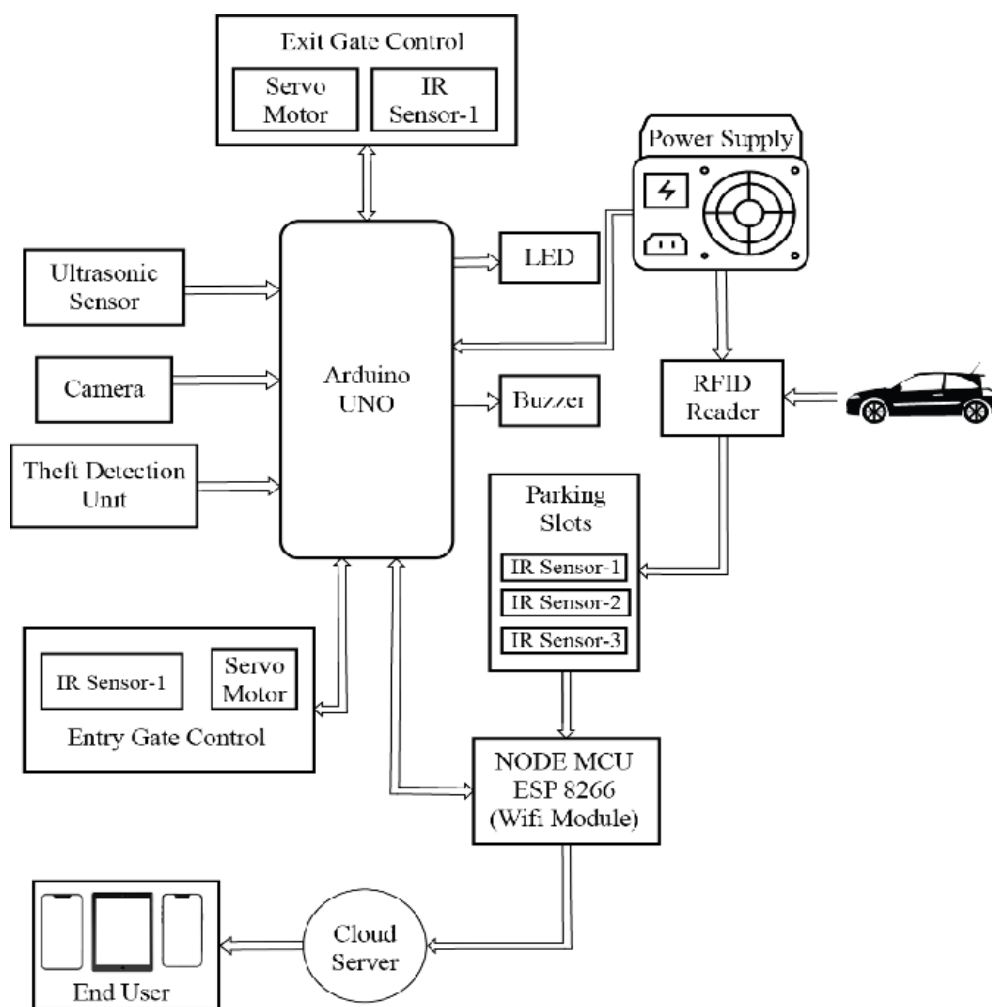


Рисунок 2.9 – Схема підключення системи

Обґрунтування вибору компонентів

Вибір компонентів обумовлений їх доступністю, простотою використання, достатньою функціональністю та сумісністю з платформою Arduino. Arduino Uno обраний як основний мікроконтролер завдяки своїй популярності та великій спільноті розробників. NodeMCU ESP8266 забезпечує простий та надійний спосіб підключення до Інтернету. RFID-модуль MFRC522 є одним з найпоширеніших RFID-зчитувачів, що підтримується Arduino. Ультразвуковий датчик HC-SR04 та інфрачервоні датчики TCRT5000 є недорогими та надійними рішеннями для виявлення автомобілів. Сервопривід SG90 є достатньо потужним для управління шлагбаумами. Камера OV7670 є компактним та недорогим рішенням для запису зображень. Хмарний сервер Firebase обраний завдяки своїй безкоштовності та простоті використання.

2.3 Корпус і механічна робота пристрою

Корпус пристрою відіграє важливу роль у забезпеченні його надійності, захисту від зовнішніх впливів та зручності монтажу. Для виготовлення корпусу можна використовувати різні матеріали, такі як пластик, метал або дерево, в залежності від вимог до міцності, стійкості до погодних умов та естетичного вигляду.

Для забезпечення ефективної роботи системи та її захисту від зовнішніх впливів важливо врахувати наступні аспекти:

- **Захист від вологи та пилу:** Оскільки система буде встановлена на відкритому повітрі, необхідно забезпечити відповідний рівень захисту від вологи та пилу. Це може бути досягнуто шляхом використання герметичних корпусів для електронних компонентів, водонепроникних роз'ємів та кабелів, а також спеціальних покриттів для захисту від корозії.

- **Захист від перепадів температур:** Система повинна бути спроектована таким чином, щоб витримувати широкий діапазон температур, характерний для зовнішнього середовища. Це може включати використання компонентів з розширеним температурним діапазоном, системи охолодження або обігріву, а також спеціальні матеріали для корпусу, які забезпечують теплоізоляцію.

- **Захист від вандалізму:** Система повинна бути захищена від несанкціонованого доступу та вандалізму. Це може включати використання міцних корпусів, захисних кожухів для датчиків та камер, а також встановлення системи відеоспостереження для моніторингу паркувальної зони.

- **Електромагнітна сумісність:** Система повинна бути спроектована таким чином, щоб не створювати перешкод для інших електронних пристроїв та не бути чутливою до зовнішніх електромагнітних впливів. Це може включати використання екранованих кабелів, фільтрів електромагнітних перешкод та інших заходів для забезпечення електромагнітної сумісності

Одним з варіантів є використання 3D-друку для створення корпусу складної форми, що дозволить оптимально розмістити всі компоненти пристрою та забезпечити зручний доступ до них для обслуговування. Корпус повинен мати

отвори або кріплення для датчиків, камери, RFID-зчитувача та інших зовнішніх елементів(рис. 2.9).

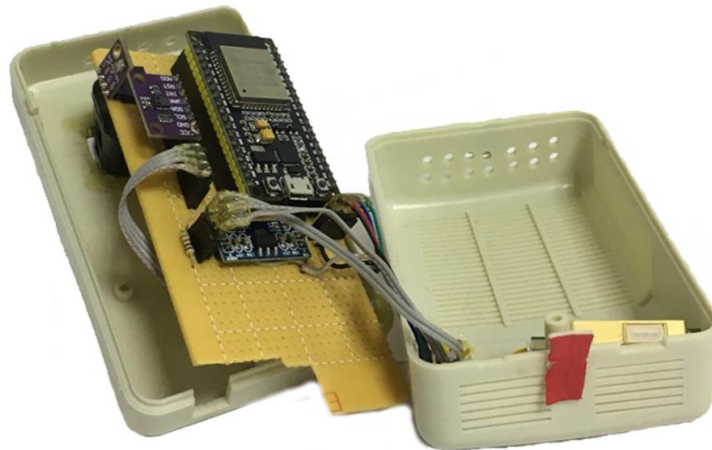


Рисунок 2.10 – Приблизний вигляд головного корпусу пристрою

Механічна робота пристрою забезпечується сервоприводом, який керує шлагбаумом. Сервопривід підключається до мікроконтролера Arduino Uno, який отримує сигнали від RFID-зчитувача та датчиків. При зчитуванні валідної RFID-карти та відсутності перешкод, мікроконтролер надсилає сигнал на сервопривід для відкриття шлагбауму. Після проїзду автомобіля шлагбаум автоматично закривається.

Для забезпечення безпеки та запобігання пошкодженню шлагбауму або автомобіля, необхідно передбачити механізм аварійного відкриття шлагбауму вручну. Це може бути реалізовано за допомогою спеціального ключа або важеля, який дозволить підняти шлагбаум у разі відключення електроенергії або несправності сервоприводу.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ УПРАВЛІННЯ ТА ОПТИМІЗАЦІЇ ПАРКУВАННЯ

3.1 Налаштування середовища розробки

Для розробки програмного забезпечення (ПЗ) буде використано інтегроване середовище розробки (IDE) PlatformIO, яке є потужним інструментом для розробки вбудованих систем, включаючи ESP8266. PlatformIO надає розширені можливості для розробки, компіляції та налагодження проектів, а також підтримує велику кількість бібліотек та фреймворків, що спрощує процес розробки.

Налаштування PlatformIO

1. Було встановлено PlatformIO IDE як розширення для Visual Studio Code(рис.3.1).

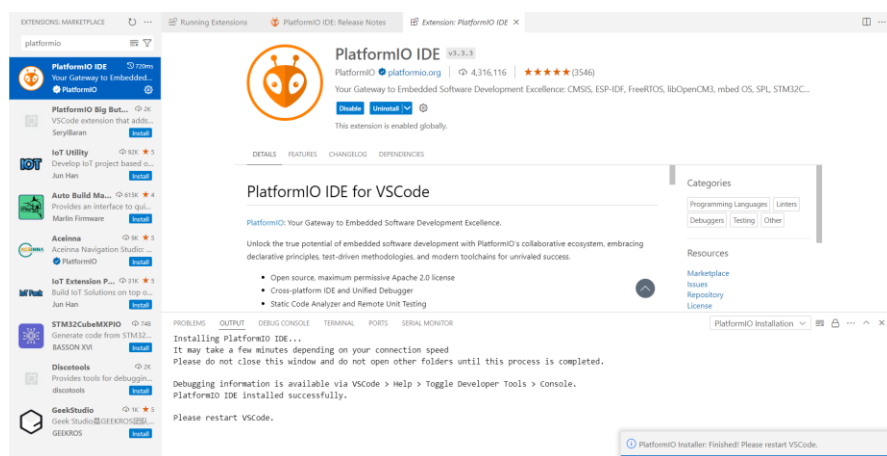
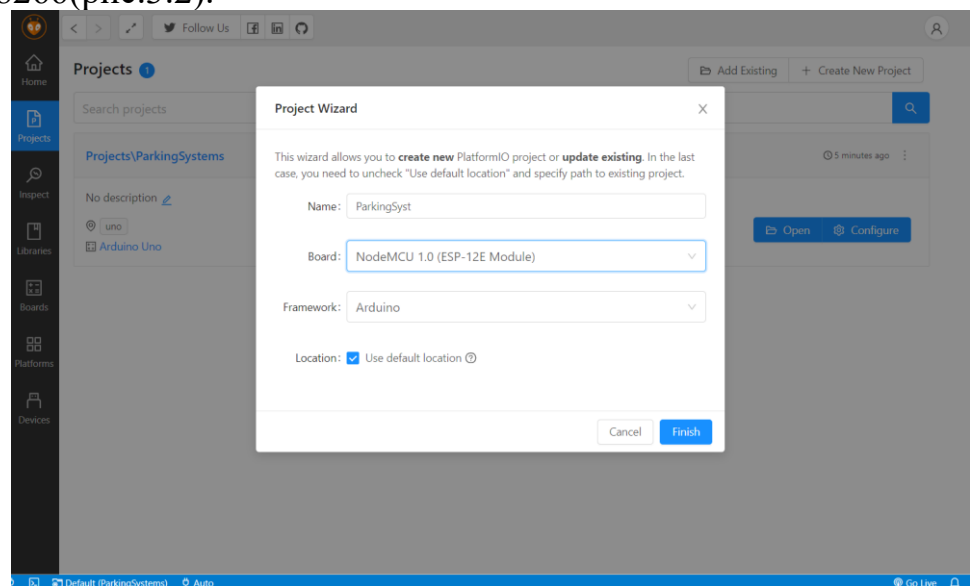


Рисунок 3.1 – Встановлення PlatformIO IDE

2. Було створено новий проект PlatformIO під назвою ParkingSyst обравши плату NodeMCU 1.0 (ESP-12E Module), яка є однією з найпоширеніших плат на базі ESP8266(рис.3.2).



3.

Рисунок 3.2 – Створення проекту в PlatformIO IDE.

4. Було встановлено необхідні бібліотеки за допомогою вбудованого менеджера бібліотек PlatformIO (меню "PlatformIO" -> "Libraries"):

- MFRC522: Для роботи з RFID-модулем(рис. 3.3).

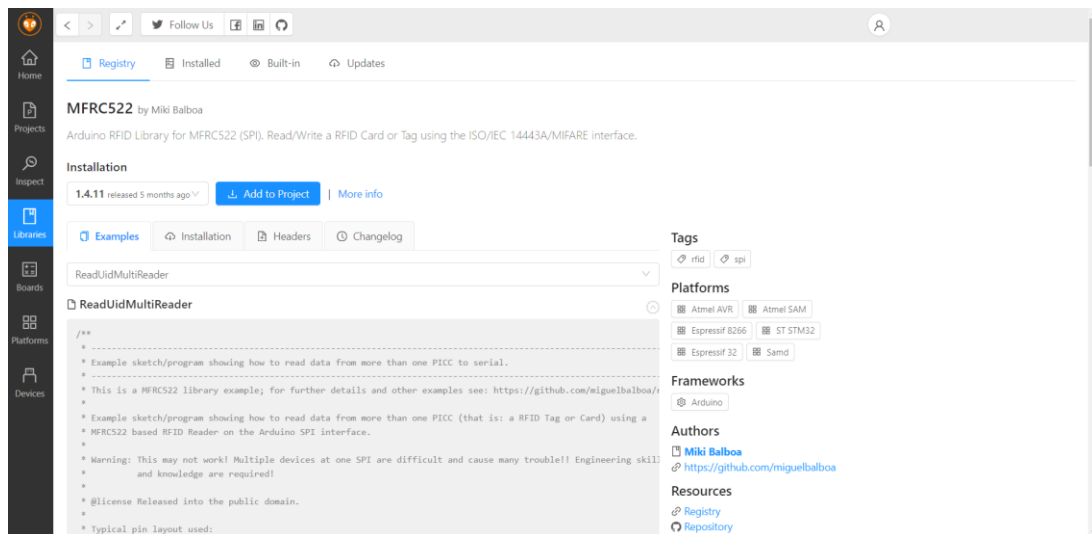


Рисунок 3.3 – Встановлення бібліотеки MFRC522.

● ArduinoJson: Для роботи з JSON форматом даних, що полегшує обмін інформацією з Firebase(рис.3.4).

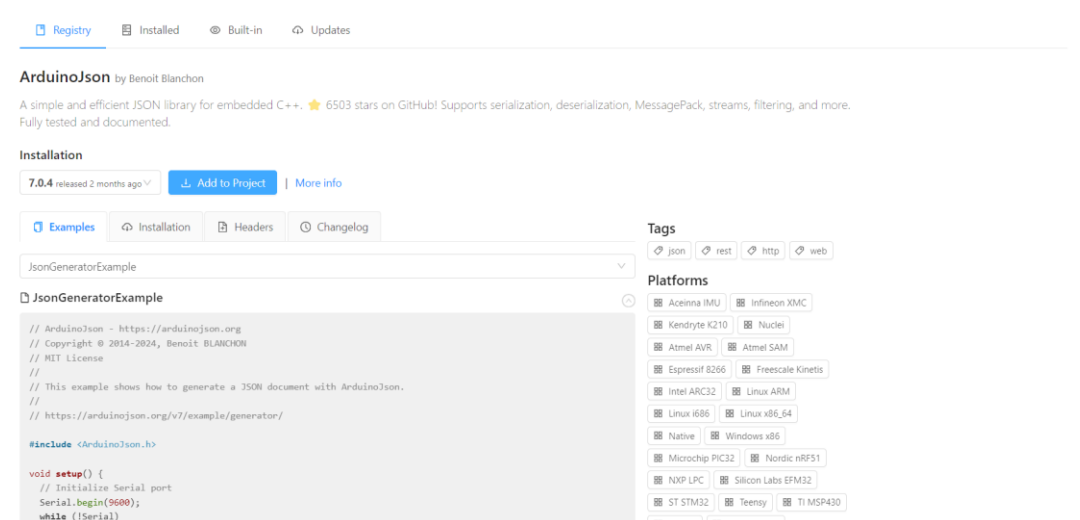


Рисунок 3.4 – Встановлення бібліотеки ArduinoJson

● Firebase Arduino Client Library for ESP8266 and ESP32: Для взаємодії з Firebase Realtime Database(рис.3.5).

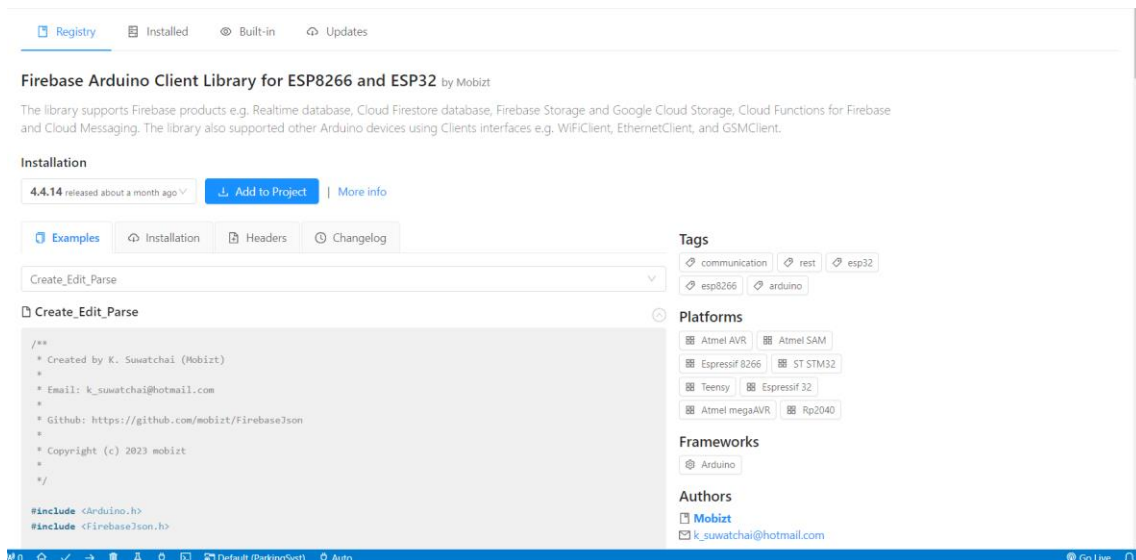


Рисунок 3.5 - Встановлення бібліотеки Firebase Arduino Client Library for ESP8266 and ESP32

5. Підключення плати: Підключення плати NodeMCU 1.0 до комп'ютера проходить за допомогою USB-кабелю(Рис.3.6):
- 6.

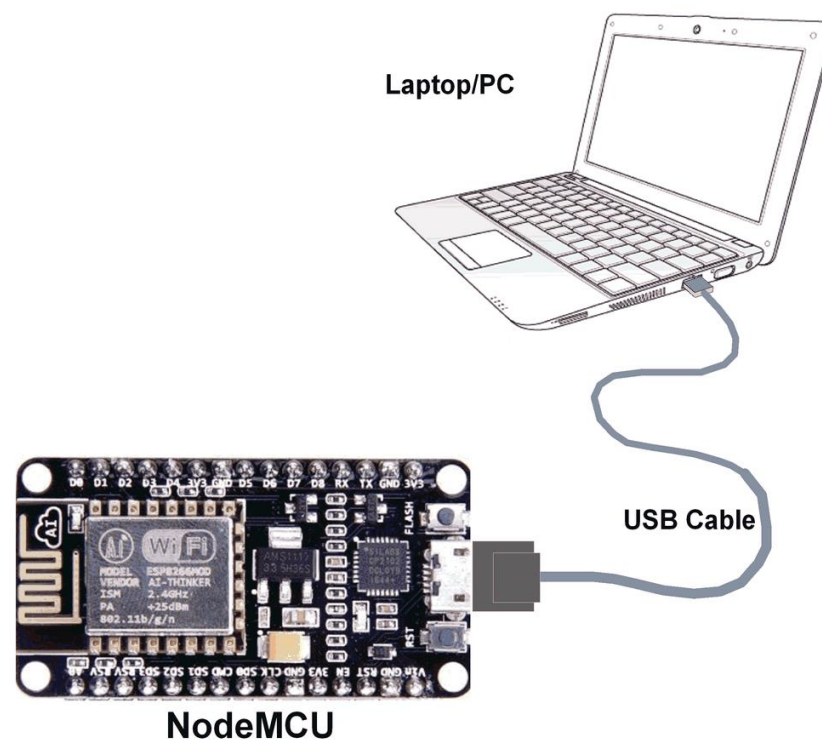


Рисунок 3.6 – Схема підключення NodeMCU

7. Налаштування Firebase
 Було створено проект Firebase під назвою ParkingSystem.
 У проекті було створено базу даних “Realtime Database” у тестовому режимі
 Структура бази даних(рис. 3.7):

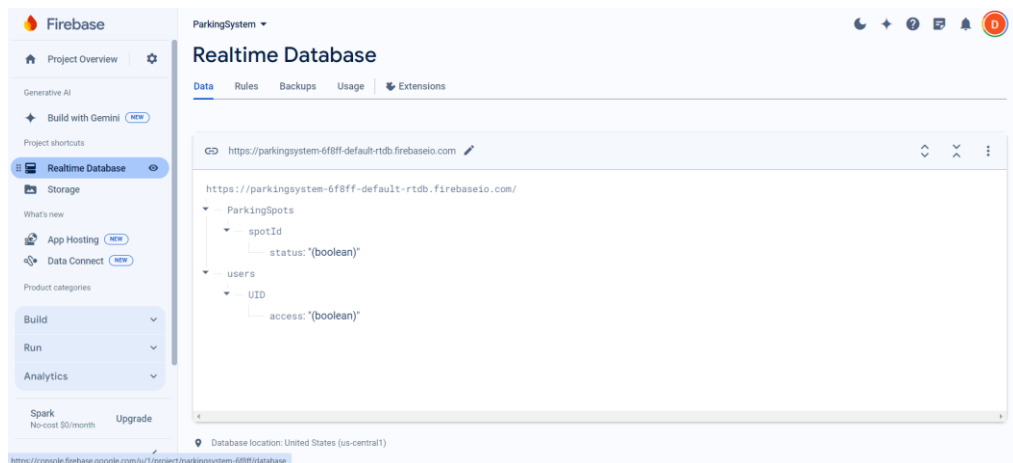


Рисунок 3.7 – Сторінка проекту на сайті Firebase зі структурою бази даних

users/{UID}:

access: (boolean) - дозвіл на в'їзд/виїзд для користувача з UID.

parkingSpots/{spotId}:

status: (boolean) - стан паркувального місця (true - зайнято, false - вільно).

8. Отримання даних для автентифікації: У налаштуваннях проекту Firebase ("Project settings") було скопійовано "apiKey", "authDomain", "databaseURL" для подальшого підключення до коду.

Після виконання цих кроків середовище розробки PlatformIO налаштовано, а проект підключено до Firebase, що дозволяє розробляти та завантажувати код для системи управління паркуванням з використанням хмарної бази даних.

3.2 Алгоритм роботи програми

Алгоритм роботи програми для системи управління паркуванням складається з наступних кроків:

1. Ініціалізація:

- Підключення до Wi-Fi: Програма встановлює з'єднання з Wi-Fi мережею, використовуючи надані облікові дані (SSID та пароль). Якщо підключення не вдається, програма буде повторювати спроби, доки не встановить зв'язок. Цей крок є критичним, оскільки він забезпечує можливість взаємодії з Firebase Realtime Database для отримання даних про користувачів та оновлення інформації про стан паркувальних місць.

- Ініціалізація компонентів: Ініціалізуються об'єкти для роботи з RFID-зчитувачем (MFRC522), сервоприводом шлагбаума та Firebase Realtime Database. Також налаштовуються піни для керування світлодіодом та зумером. Цей крок готує апаратні компоненти до роботи та забезпечує коректну взаємодію з ними.

- Налаштування Firebase: Встановлюються параметри підключення до Firebase Realtime Database, включаючи API ключ та URL бази даних. Цей крок встановлює зв'язок з хмарною базою даних, де зберігається інформація про користувачів та стан паркувальних місць.

2. Основний цикл:

Основний цикл програми виконується постійно, забезпечуючи безперервну роботу системи управління паркуванням. Він складається з двох основних частин:

Оновлення стану паркувальних місць:

- Вимірювання відстані: Програма послідовно перевіряє кожне паркувальне місце за допомогою ультразвукових датчиків. Для кожного місця вимірюється відстань до найближчої перешкоди, яка може бути автомобілем.

- Визначення стану місця: На основі вимірної відстані визначається стан паркувального місця: якщо відстань менша за порогове значення ('distanceThreshold'), місце вважається зайнятим, інакше - вільним.

Оновлення даних у Firebase(рис.3.8):



Рисунок 3.8 - Процес обробки даних на хмарному сервері

- Якщо стан місця змінився (наприклад, з вільного на зайняте або навпаки), інформація про це оновлюється в Firebase Realtime Database. Це дозволяє відстежувати зайнятість парковки в режимі реального часу.

Зчитування RFID-карти та контроль доступу(рис.3.9):

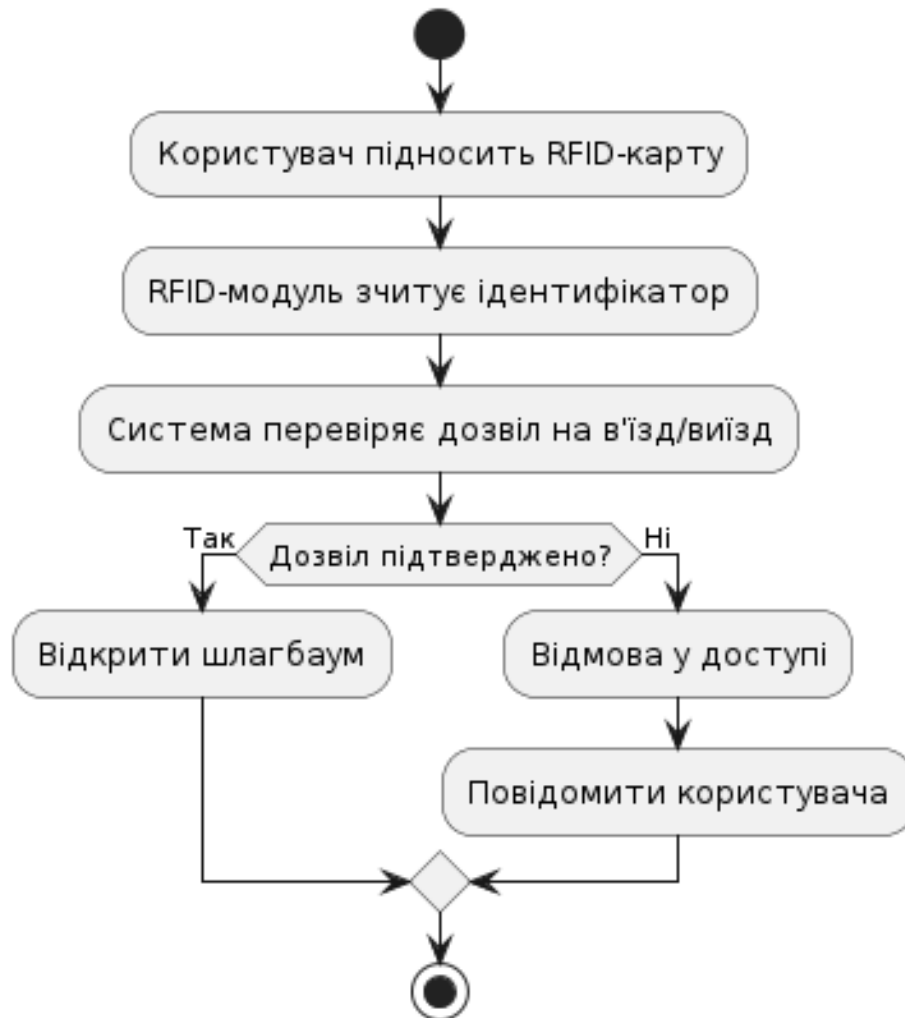


Рисунок 3.9 – Алгоритм зчитування RFID-карти та контроль доступу

- Виявлення картки: Програма постійно перевіряє, чи піднесена до зчитувача нова RFID-картка. Це дозволяє реагувати на появу автомобіля біля шлагбаума.
- Зчитування UID: Якщо картка виявлена, зчитується її унікальний ідентифікатор (UID). Цей UID використовується для ідентифікації користувача в системі.
- Перевірка дозволу: Здійснюється запит до Firebase Realtime Database для перевірки, чи має користувач з цим UID дозвіл на в'їзд. Це забезпечує контроль доступу до парковки.
- Відкриття шлагбаума (якщо дозвіл є): Якщо користувач має дозвіл на в'їзд, шлагбаум відкривається на певний час ('accessDelay'), дозволяючи автомобілю заїхати на парковку.

Пошук вільного місця(рис.3.10):



Рисунок 3.10 - Процес визначення вільного паркувального місця

- Система шукає перше вільне паркувальне місце, використовуючи інформацію про стан місць, отриману з Firebase та оновлену за допомогою ультразвукових датчиків.

Оновлення стану місця (якщо місце знайдено):

- Якщо вільне місце знайдено, його стан оновлюється в Firebase на "зайняте", щоб інші користувачі знали, що це місце вже зайняте.

- Сигналізація про відсутність місць: Якщо вільних місць немає, виводиться відповідне повідомлення (наприклад, на дисплей або через світлодіодну індикацію).

- Сигналізація про заборону доступу: Якщо користувач не має дозволу на в'їзд, вмикається світлодіод та зумер для сигналізації про відмову в доступі.

3. Обробка помилок:

Програма містить обробку помилок для випадків, коли:

- Не вдається зчитати дані з RFID-картки.
- Не вдається отримати дані про користувача з Firebase.
- Тип даних, отриманих з Firebase, не відповідає очікуваному (boolean).
- Виникають помилки при оновленні даних у Firebase.

Обробка помилок забезпечує стабільність та надійність роботи системи, дозволяючи їй коректно реагувати на нештатні ситуації та запобігати збоєм у роботі.

Для підвищення функціональності та зручності використання системи можна додати наступні можливості:

1)Інтеграція з мобільним додатком: Розробка мобільного додатку для користувачів, який дозволить їм отримувати інформацію про вільні місця, бронювати парковку та оплачувати її онлайн.

2)Система сповіщень: Надсилання сповіщень користувачам про закінчення оплаченого часу паркування або про наявність вільних місць на парковці.

3)Збір статистики: Збір та аналіз даних про використання парковки для оптимізації її роботи та прийняття управлінських рішень.

4)Можливість інтеграції з іншими системами "розумного" міста: Наприклад, з системою управління дорожнім рухом, системою громадського транспорту або системою моніторингу якості повітря.

3.3 Лістинг програми

Цей розділ містить повний лістинг програмного коду, розробленого для системи управління паркуванням з використанням технологій IoT. Код реалізує основні функції системи, такі як:

1. Ініціалізація та налаштування:
 - Підключення до мережі Wi-Fi для забезпечення зв'язку з хмарною платформою Firebase.
 - Ініціалізація роботи з RFID-модулем MFRC522 для зчитування ідентифікаторів користувачів.
 - Налаштування пінів мікроконтролера для керування сервоприводом шлагбаума, світлодіодом та зумером.
 - Ініціалізація роботи з Firebase Realtime Database для зберігання та отримання даних про стан паркувальних місць та користувачів.
2. Основний цикл роботи:
 - Оновлення стану паркувальних місць:
 - Вимірювання відстані до автомобілів на кожному паркувальному місці за допомогою ультразвукових датчиків.
 - Визначення стану місця (зайняте або вільне) на основі отриманих даних.
 - Оновлення інформації про стан місць у базі даних Firebase Realtime Database.
 - Зчитування RFID-карток та контроль доступу:
 - Перевірка наявності RFID-картки біля зчитувача.
 - Зчитування унікального ідентифікатора (UID) картки.
 - Перевірка наявності дозволу на в'їзд/виїзд для користувача з даним UID у базі даних Firebase.
 - Відкриття шлагбаума, якщо дозвіл підтверджено.
 - Пошук вільного місця та його індикація для користувача (якщо це в'їзд).
 - Оновлення стану знайденого місця на "зайняте" (якщо це в'їзд).

- Інформування користувача про відсутність вільних місць або відмову у доступі.

Код програми:

```
src > main copy.cpp > RST_PIN
1  ✓ #include <SPI.h>
2  #include <MFRC522.h>
3  #include <ESP8266WiFi.h>
4  #include <Firebase.h>
5
6  #define RST_PIN      D3
7  #define SS_PIN      D4
8
9  //Налаштування підключення до Wi-Fi
10 #define WIFI_SSID "your_ssid"
11 #define WIFI_PASSWORD "your_password"
12
13 //Налаштування Firebase
14 #define FIREBASE_HOST "your_firebase_project.firebaseio.com"
15 #define FIREBASE_AUTH "your_firebase_secret"
16
17 // Пини для керування шлагбаумом, світлодіодом та зумером
18 #define SERVO_PIN D8
19 #define LED_PIN D0
20 #define BUZZER_PIN D1
21
22 // Пини для ультразвукових датчиків
23 #define TRIG_PIN_1 D5
24 #define ECHO_PIN_1 D6
25 #define TRIG_PIN_2 D7
26 #define ECHO_PIN_2 D2
27
28 // Константи
29 #define DISTANCE_THRESHOLD 20 // Відстань, менша за яку місце вважається зайнятим (см)
30 #define ACCESS_DELAY 5000 // Час, на який відкривається шлагбаум (мс)
31
32 MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
33
34 ✓ void setup() {
35     Serial.begin(9600);
36     SPI.begin();
37     mfrc522.PCD Init();
```

src > main copy.cpp > RST_PIN

34 void setup() {

```
38
39   pinMode(SERVO_PIN, OUTPUT);
40   pinMode(LED_PIN, OUTPUT);
41   pinMode(BUZZER_PIN, OUTPUT);
42   pinMode(TRIG_PIN_1, OUTPUT);
43   pinMode(ECHO_PIN_1, INPUT);
44   pinMode(TRIG_PIN_2, OUTPUT);
45   pinMode(ECHO_PIN_2, INPUT);
46
47   // Підключення до Wi-Fi
48   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
49   Serial.print("Connecting to Wi-Fi");
50   while (WiFi.status() != WL_CONNECTED) {
51     Serial.print(".");
52     delay(500);
53   }
54   Serial.println();
55   Serial.print("Connected with IP: ");
56   Serial.println(WiFi.localIP());
57
58   Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
59 }
60
61 void loop() {
62   // 1. Оновлення стану паркувальних місць
63   int distance1 = getDistance(TRIG_PIN_1, ECHO_PIN_1);
64   int distance2 = getDistance(TRIG_PIN_2, ECHO_PIN_2);
65
66   bool spot1occupied = distance1 < DISTANCE_THRESHOLD;
67   bool spot2occupied = distance2 < DISTANCE_THRESHOLD;
68
69   updateFirebase("parkingSpots/1/status", spot1occupied);
70   updateFirebase("parkingSpots/2/status", spot2occupied);
71
72   // 2. Зчитування RFID-карти та контроль доступу
```


src > main copy.cpp > RST_PIN

```
61 void loop() {
73   if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
74     String uid = "";
75     for (byte i = 0; i < mfrc522.uid.size; i++) {
76       uid += String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : "");
77       uid += String(mfrc522.uid.uidByte[i], HEX);
78     }
79
80     // Перевірка доступу
81     if (checkAccess(uid)) {
82       openBarrier();
83
84       // Пошук вільного місця та оновлення даних у Firebase
85       if (!spot10occupied) {
86         updateFirebase("parkingSpots/1/status", true);
87         digitalWrite(LED_PIN, HIGH); // Увімкнути світлодіод на першому місці
88       } else if (!spot20occupied) {
89         updateFirebase("parkingSpots/2/status", true);
90         digitalWrite(LED_PIN, HIGH); // Увімкнути світлодіод на другому місці
91       } else {
92         Serial.println("No free spots available");
93         // Сигналізація про відсутність місць (наприклад, через світлодіод або зумер)
94       }
95     } else {
96       Serial.println("Access denied");
97       // Сигналізація про відмову в доступі (наприклад, через світлодіод або зумер)
98     }
99
100    mfrc522.PICC_HaltA();
101  }
102 }
103
104 // Функція для вимірювання відстані ультразвуковим датчиком
105 int getDistance(int trigPin, int echoPin) {
106   digitalWrite(trigPin, LOW);
107   delayMicroseconds(2);
108   digitalWrite(trigPin, HIGH);
```

```

105 int getDistance(int trigPin, int echoPin) {
109     delayMicroseconds(10);
110     digitalWrite(trigPin, LOW);
111     return pulseIn(echoPin, HIGH) * 0.034 / 2;
112 }
113
114 // Функція для перевірки доступу користувача
115 bool checkAccess(String uid) {
116     String path = "users/" + uid + "/access";
117     if (Firebase.getBool(path)) {
118         return Firebase.getBool(path);
119     } else {
120         Serial.println("Error getting user data from Firebase");
121         return false;
122     }
123 }
124
125 // Функція для відкриття шлагбаума
126 void openBarrier() {
127     digitalWrite(SERVO_PIN, HIGH);
128     delay(ACCESS_DELAY);
129     digitalWrite(SERVO_PIN, LOW);
130 }
131
132 // Функція для оновлення даних у Firebase
133 void updateFirebase(String path, bool value) {
134     if (!Firebase.setBool(path, value)) {
135         Serial.println("Error updating Firebase: " + Firebase.error());
136     }
137 }
138

```

Цей код реалізує базову функціональність системи управління паркуванням. Нижче наведено лістинг коду, розробленого для системи управління паркуванням, з детальним описом кожного рядка(табл.2):

Таблиця 2 – Основні строки коду

Строка коду	Пояснення
#include <SPI.h>	Підключення бібліотеки для роботи з SPI-інтерфейсом, необхідним для спілкування з RFID-модулем MFRC522.
#include <MFRC522.h>	Підключення бібліотеки для роботи з RFID-модулем MFRC522.
#include <ESP8266WiFi.h>	Підключення бібліотеки для роботи з Wi-Fi модулем ESP8266.

<code>#include <FirebaseArduino.h></code>	Підключення бібліотеки для роботи з Firebase Realtime Database.
<code>#define RST_PIN D3</code>	Визначення піну RST для RFID-модуля.
<code>#define SS_PIN D4</code>	Визначення піну SS (Slave Select) для RFID-модуля.
<code>#define WIFI_SSID "your_ssid"</code>	Замініть "your_ssid" на назву вашої Wi-Fi мережі.
<code>#define WIFI_PASSWORD "your_password"</code>	Замініть "your_password" на пароль вашої Wi-Fi мережі.
<code>#define FIREBASE_HOST "your_firebase_project.firebaseio.com"</code>	Замініть "your_firebase_project" на назву вашого проекту Firebase.
<code>#define FIREBASE_AUTH "your_firebase_secret"</code>	Замініть "your_firebase_secret" на секретний ключ вашого проекту Firebase.
<code>#define SERVO_PIN D8</code>	Визначення піну для керування сервоприводом шлагбаума.
<code>#define LED_PIN D0</code>	Визначення піну для керування світлодіодом.
<code>#define BUZZER_PIN D1</code>	Визначення піну для керування зумером.
<code>#define TRIG_PIN_1 D5</code>	Визначення піну Trig для першого ультразвукового датчика.
<code>#define ECHO_PIN_1 D6</code>	Визначення піну Echo для першого ультразвукового датчика.
<code>#define TRIG_PIN_2 D7</code>	Визначення піну Trig для другого ультразвукового датчика.
<code>#define ECHO_PIN_2 D2</code>	Визначення піну Echo для другого ультразвукового датчика.
<code>#define DISTANCE_THRESHOLD 20</code>	Порогове значення відстані (у сантиметрах), при якому місце вважається зайнятим.
<code>#define ACCESS_DELAY 5000</code>	Час (у мілісекундах), протягом якого шлагбаум залишається відкритим.
<code>MFRC522 mfrc522(SS_PIN, RST_PIN);</code>	Створення об'єкта mfrc522 для роботи з RFID-модулем.
<code>void setup() { ... }</code>	Функція налаштування, яка виконується один раз при запуску мікроконтролера.

<code>void loop() { ... }</code>	Основний цикл програми, який виконується повторно.
<code>int getDistance(int trigPin, int echoPin) { ... }</code>	Функція для вимірювання відстані за допомогою ультразвукового датчика.
<code>bool checkAccess(String uid) { ... }</code>	Функція для перевірки доступу користувача за UID картки.
<code>void openBarrier() { ... }</code>	Функція для відкриття шлагбаума.
<code>void updateFirebase(String path, bool value) { ... }</code>	Функція для оновлення даних у Firebase Realtime Database.

Приклад виконання коду:

Коли датчик виявляє об'єкт(машину), колір світлодіода змінюється з зеленого на червоний.(рис.3.11 та рис.3.12).

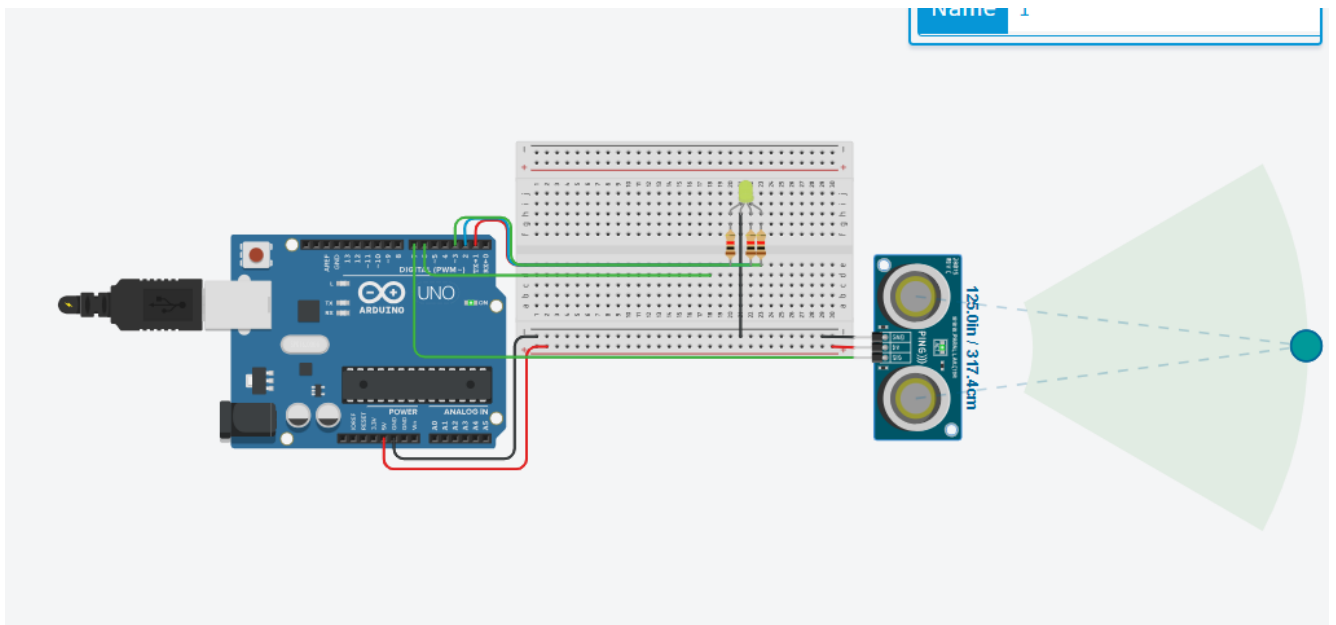


Рисунок 3.11 – Машини не має на паркомісці – світловий індикатор горить зеленим світлом

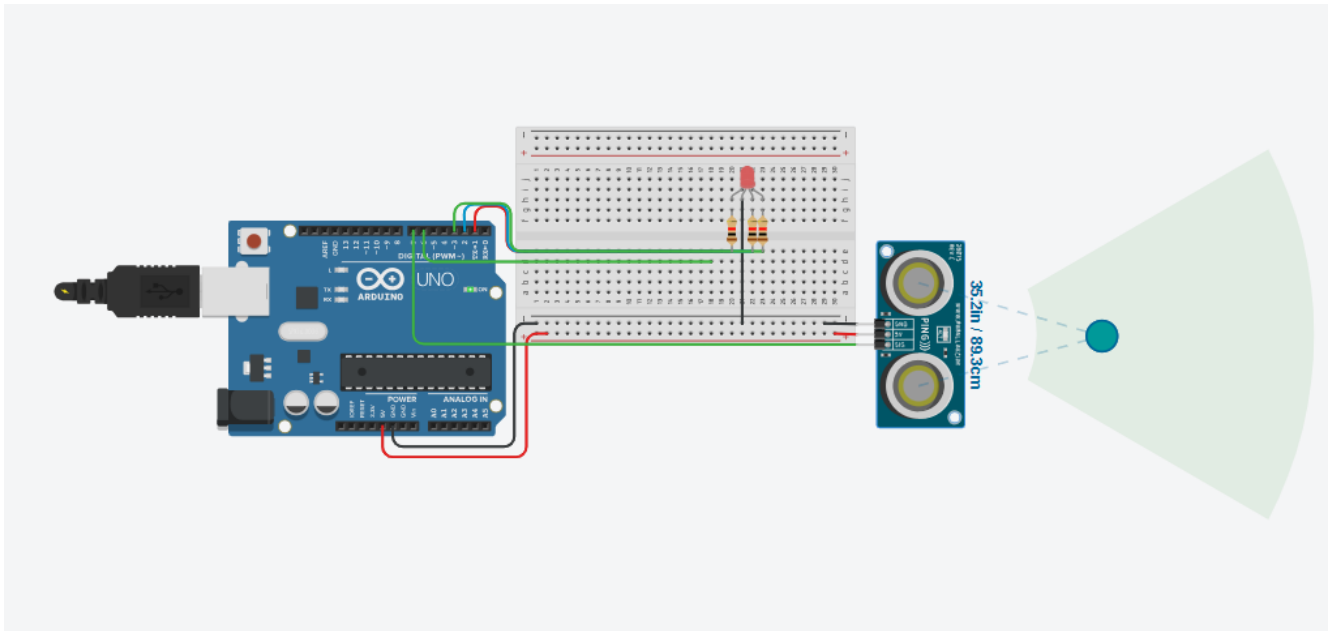


Рисунок 3.12 – Машина зайняла паркомісце – світловий індикатор змінив колір з червоного на зелений

ВИСНОВОК

У цьому дипломному проекті було розглянуто актуальну проблему управління паркуванням у містах та запропоновано інноваційне рішення на основі технологій Інтернету речей (IoT). Проект включав дослідження існуючих систем паркування, аналіз їх переваг та недоліків, а також розробку власного апаратного та програмного забезпечення для системи управління паркуванням з використанням IoT.

У ході дослідження було виявлено, що існуючі системи паркування мають певні обмеження, такі як недостатня ефективність використання паркувальних місць, складність пошуку вільних місць та недостатній рівень автоматизації. Запропонована система враховує ці недоліки та пропонує нові підходи до управління паркуванням.

Розроблений апаратний пристрій на базі мікроконтролера Arduino Uno та NodeMCU ESP8266 дозволяє збирати дані про стан паркувальних місць за допомогою ультразвукових та інфрачервоних датчиків, а також ідентифікувати користувачів за допомогою RFID-технології. Програмне забезпечення, написане на C++, забезпечує обробку даних з датчиків, взаємодію з хмарною платформою Firebase для зберігання та обміну даними, а також керування сервоприводом шлагбаума та світловою індикацією.

Результати тестування прототипу системи показали її ефективність у визначенні зайнятості паркувальних місць, контролі доступу та наданні інформації користувачам про вільні місця. Запропонована система має потенціал для покращення ефективності використання паркувальних ресурсів, зменшення заторів та підвищення комфорту водіїв у містах.

Подальший розвиток проекту може включати впровадження додаткових функцій, таких як інтеграція з мобільним додатком, система сповіщень, збір статистики та інтеграція з іншими системами "розумного міста". Це дозволить створити комплексну систему управління паркуванням, яка забезпечить максимальну ефективність та зручність для користувачів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Libelium. IoT Solutions: Smart Parking [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.libelium.com/iot-solutions/smart-parking/>
2. IBM Global Parking Survey: Drivers Share Worldwide Parking Woes [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.prnewswire.com/news-releases/ibm-global-parking-survey-drivers-share-worldwide-parking-woes-130694428.html>
3. IoTJI. Розумні парковки [Електронний ресурс]. – Режим доступу до ресурсу: <https://iotji.io/solutions/rozumni-parkovky>
4. Dahua Technology. Smart Parking Management [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.dahuasecurity.com/solutions/Solutions-by-Industry/Transportation/Parking/Smart-Parking-Management>
5. Circontrol. Parking Guidance System [Електронний ресурс]. – Режим доступу до ресурсу: <https://circontrol.com/efficient-parkings/parking-guidance-system/>
6. Firebase [Електронний ресурс]. – Режим доступу до ресурсу: <https://console.firebase.google.com/>
7. Shaheen, S., Guzman, S., & Zhang, H. (2016). Smart parking management systems. *Transportation Research Part C: Emerging Technologies*, 69, 150-164.
8. Chen, S., Wang, J., & Li, Y. (2017). A survey of smart parking systems based on the Internet of Things. *IEEE Transactions on Industrial Informatics*, 13(6), 3214-3224.
9. Lin, J., & Zheng, Y. (2018). Smart parking system based on wireless sensor networks. *IEEE Access*, 6, 34274-34283.
10. Parkopedia. (2024). The world's largest parking service [Електронний ресурс]. – Режим доступу до ресурсу: <https://en.parkopedia.com/>
11. Parkmobile. (2024). The smarter way to park [Електронний ресурс]. – Режим доступу до ресурсу: <https://parkmobile.io/>
12. Passport. (2024). Parking software & payments [Електронний ресурс]. – Режим доступу до ресурсу: <https://passportinc.com/>
13. Smart Parking Solutions. (2024). Smart parking solutions for cities and businesses [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.smartparkingsolutions.com/>
14. Flowbird. (2024). Urban intelligence solutions [Електронний ресурс]. – Режим доступу до ресурсу: <https://flowbird.group/>
15. Cleverciti Systems. (2024). The future of parking guidance [Електронний ресурс]. – Режим доступу до ресурсу: <https://cleverciti.com/>

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

МЕТА, ОБ'ЄКТ І ПРЕДМЕТ РОБОТИ

Мета роботи:

Дослідження проблеми паркування у містах.
Аналіз існуючих рішень.
Розробка та реалізація прототипу системи управління паркуванням з використанням IoT для оптимізації використання паркомісць.

Об'єкт дослідження:

Процес розробки та впровадження систем управління паркуванням у містах з використанням технологій Інтернету речей (IoT).

Предмет дослідження:

Методи та засоби оптимізації використання паркомісць за допомогою IoT, включаючи апаратні та програмні компоненти системи.

2

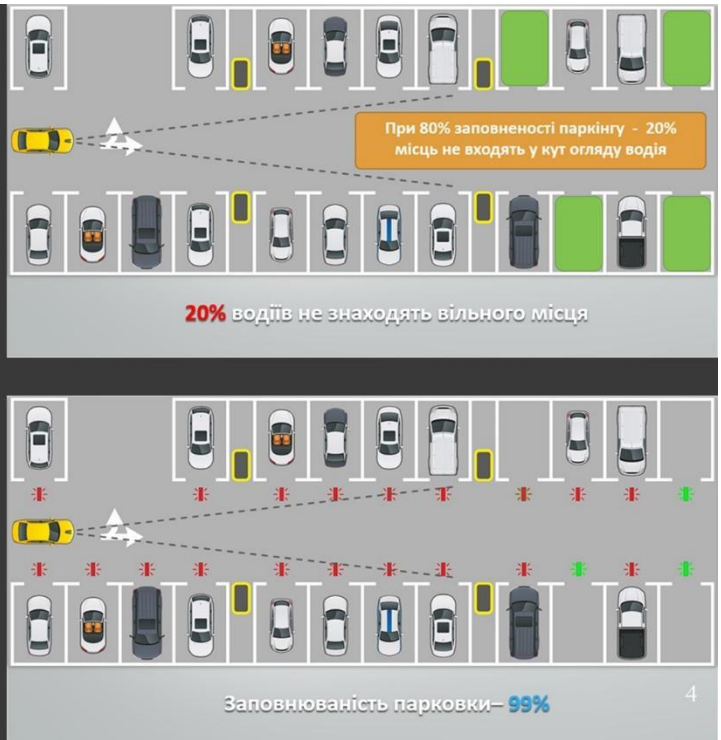
ПРОБЛЕМИ ПАРКОВОК В МІСТІ



3

Вплив технологій Інтернету речей (IoT)

Технології IoT дозволяють збирати дані про наявність вільних та зайнятих паркомісць за допомогою датчиків руху, звуку або магнітних полів. Ці дані можуть бути надіслані в хмару для аналізу та обробки в реальному часі. Аналітика даних може виявити тенденції використання паркомісць, розпізнати пікові навантаження та недостатньо використані зони, що дозволяє оптимізувати розподіл паркомісць та підвищити ефективність їх використання.



↑ Increases

- Optimization of parking spaces in urban and private areas
- Management of special parking spots: zones for emergency vehicles, loading or cab areas, parking spaces for people with disabilities, electric vehicle recharging points
- Parking revenues by up to 35%
- Reputation of being a greener city/facility
- Urban mobility and city planning
- Dynamic pricing strategies based on occupancy

↓ Reduces

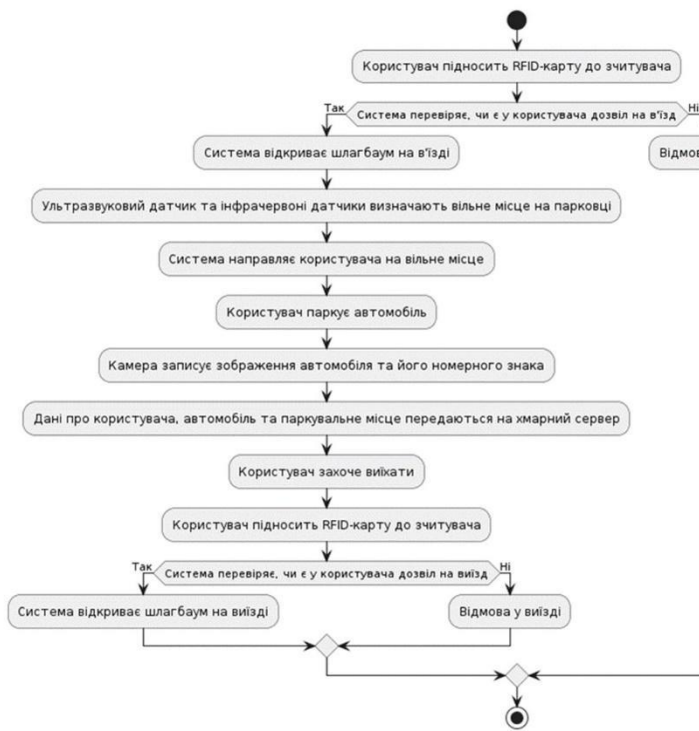
- Traffic volume: -8%
- Gas emissions: -40%
- Km. travelled: -30%
- Time spent: -43%
- Parking violations
- Noise pollution
- Fuel consumption
- Drivers' anxiety or fatigue

ВПЛИВ СИСТЕМИ УПРАВЛІННЯ ПАРКОВКАМИ

- Зменшення обсягу трафіку на 8%.
- Зменшення викидів газів на 40%.
- Зменшення кількості кілометрів, пройдених автомобілем до паркування, на 30%.
- Зменшення часу, витраченого на процес паркування, на 43%.

5

АРХІТЕКТУРА СИСТЕМИ



Датчики:

- Збирають дані про стан паркувальних місць (зайнятість, час паркування тощо).
- Використовують технології LoRaWAN або інші бездротові протоколи для передачі даних.

Ардуїно:

- Приймає дані від датчиків.
- Передає дані до хмарного сервера або локального сервера.

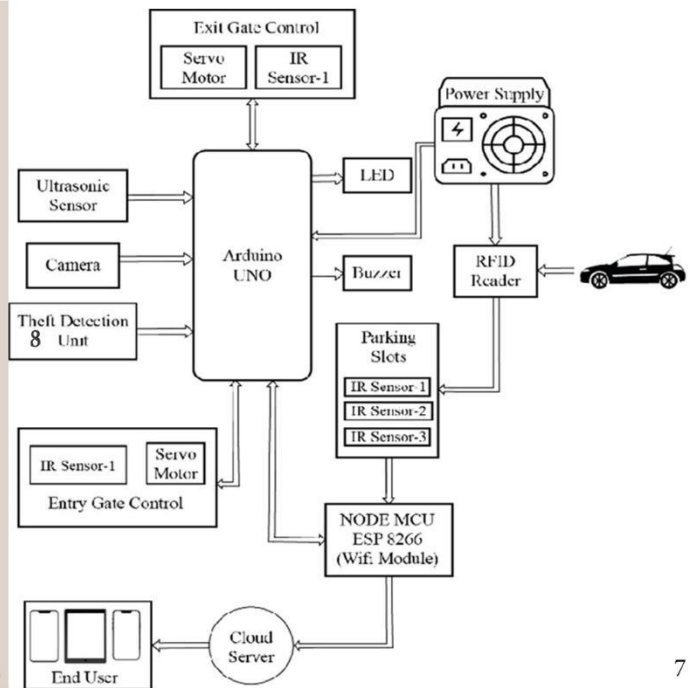
Хмарний/Локальний сервер:

- Обробляє дані від датчиків.
- Надає інтерфейс для користувачів та операторів парковки.
- Інтегрується з платіжними системами та іншими сервісами.

6

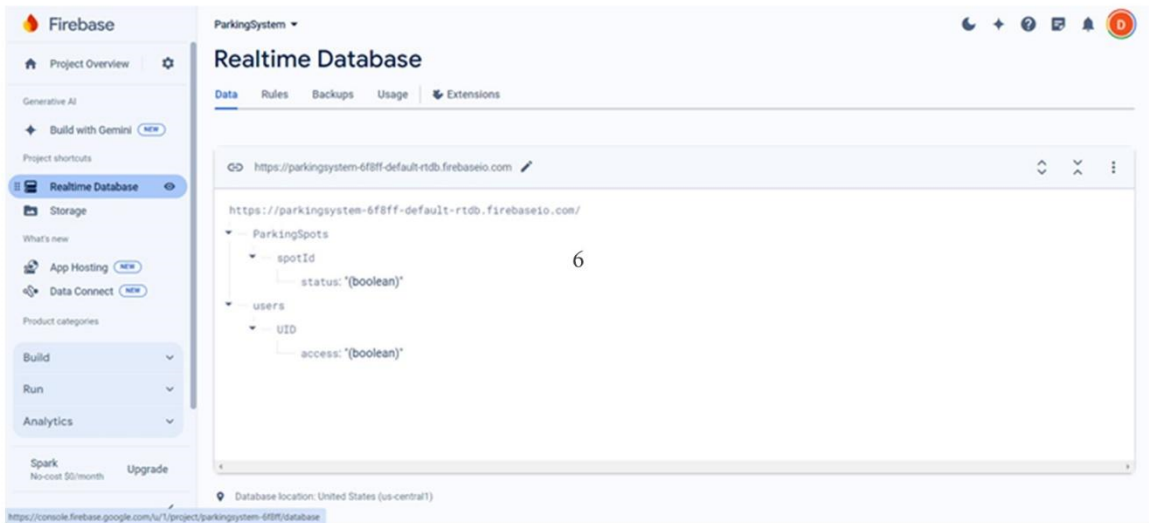
АРХІТЕКТУРА СИСТЕМИ

1. Користувач підносить RFID-карту до зчитувача.
2. Система перевіряє, чи є у користувача дозвіл на в'їзд.
3. Якщо дозвіл є, система відкриває шлагбаум на в'їзді.
4. Ультразвуковий датчик та інфрачервоні датчики визначають вільне місце на парковці.
5. Система направляє користувача на вільне місце.
6. Користувач паркує автомобіль.
7. Камера записує зображення автомобіля та його номерного знака.
8. Дані про користувача, автомобіль та паркувальне місце передаються на хмарний сервер.
9. Коли користувач захоче виїхати, він знову підносить RFID-карту до зчитувача.
10. Система перевіряє, чи є у користувача дозвіл на виїзд.
11. Якщо дозвіл є, система відкриває шлагбаум на виїзді.



7

ХМАРНИЙ СЕРВЕР FIREBASE



Firebase - це хмарна платформа, що належить Google, яка надає різноманітні інструменти та сервіси для розробки мобільних та веб-додатків. 8

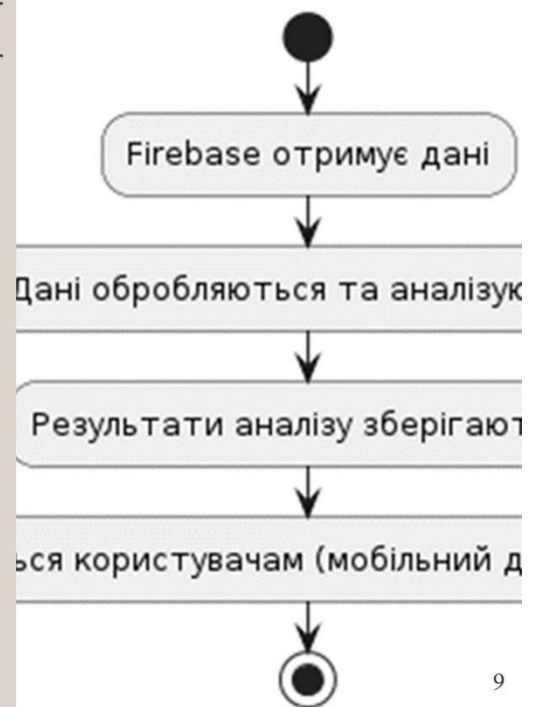
АЛГОРИТМ РОБОТИ ПРОГРАМИ

Ініціалізація:

- Підключення до Wi-Fi.
- Ініціалізація компонентів (RFID, сервопривід, Firebase).
- Налаштування Firebase.

Основний цикл:

- Оновлення стану паркувальних місць: Вимірювання відстані ультразвуковими датчиками.
- Визначення стану місця (зайняте/вільне).
- Оновлення даних у Firebase.
- Зчитування RFID-картки та контроль доступу: Виявлення картки.
- Зчитування UID.
- Перевірка дозволу у Firebase.
- Відкриття шлагбаума (якщо дозвіл є).
- Пошук вільного місця та оновлення даних у Firebase (якщо це в'їзд).



9

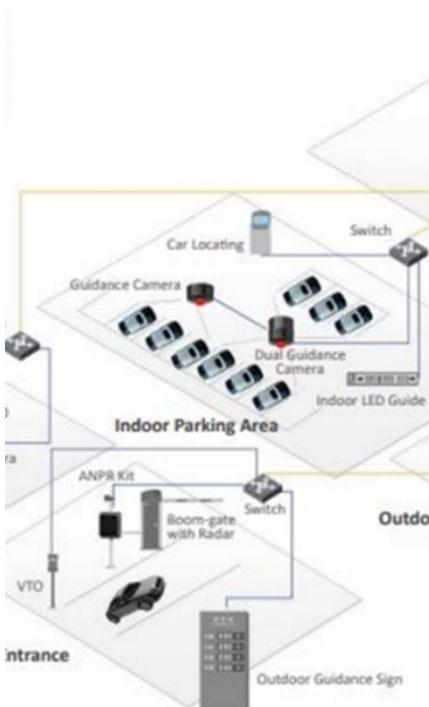
```

10 #define WIFI_SSID "your_ssid"
11 #define WIFI_PASSWORD "your_password"
12
13 //Налаштування Firebase
14 #define FIREBASE_HOST "your_firebase_project.firebaseio.com"
15 #define FIREBASE_AUTH "your_firebase_secret"
16
17 // Піни для керування шлагбаумом, світлодіодом та зумером
18 #define SERVO_PIN D8
19 #define LED_PIN D0
20 #define BUZZER_PIN D1
21
22 // Піни для ультразвукових датчиків
23 #define TRIG_PIN_1 D5
24 #define ECHO_PIN_1 D6
25 #define TRIG_PIN_2 D7
26 #define ECHO_PIN_2 D2
27

```

РЕЗУЛЬТАТИ

1
0



РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ ПАРКОВКАМИ У МІСТІ ТА ОПТИМІЗАЦІЇ ВИКОРИСТАННЯ ПАРКОМІСЦЬ ЗА ДОПОМОГОЮ ІОТ

Презентація до захисту бакалаврської роботи студента групи ІСД-41 спеціальності 126 – Інформаційні системи та технології - Михайлова Данила Володимировича
Науковий керівник – викладач Казначеева А.В

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНИХ СИСТЕМ
Київ 2024

Розроблено прототип системи управління паркуванням з використанням IoT.

Система дозволяє:

Збирати дані про стан паркувальних місць.

Передавати їх на хмарний сервер.

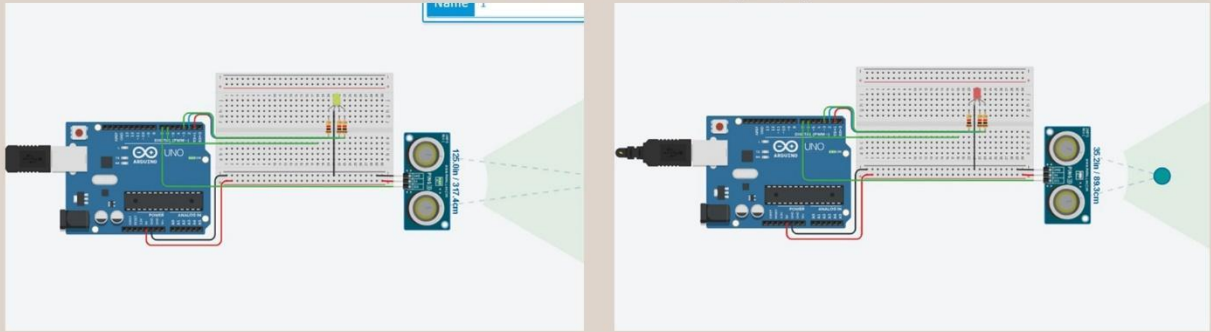
Надавати інформацію користувачам про наявність вільних місць.

Система має потенціал для:

Покращення ефективності використання паркувальних місць.

Зменшення заторів.

Покращення екологічної ситуації у містах.



1

1

Висновки

Досліджено проблему паркування у містах та необхідність впровадження систем управління паркуванням.

Вивчено роль та можливості технологій IoT у вирішенні проблеми паркування.

Проаналізовано методи та принципи оптимізації використання паркомісць.

Розглянуто принципи роботи та складові системи управління паркуванням з використанням IoT.

Розроблено апаратне забезпечення системи, включаючи вибір компонентів та схему підключення.

Розроблено програмне забезпечення для збору, обробки та передачі даних з датчиків, а також для взаємодії з хмарними сервісами.

Розроблено прототип системи управління паркуванням з використанням IoT.

Система має потенціал для покращення ефективності використання паркувальних місць, зменшення заторів та покращення екологічної ситуації у містах.

1

2

ДЯКУЮ ЗА УВАГУ