

Міністерство освіти і науки України
Чернівецький національний університету імені Юрія Федьковича»
Відокремлений структурний підрозділ «Фаховий коледж Чернівецького
національного університету імені Юрія Федьковича»

Природниче відділення
Циклова комісія комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»
зі спеціальності 123 Комп'ютерна інженерія
підготовки за освітньо-професійною програмою «Комп'ютерна інженерія»
на тему: **«Система паркування автомобілів IoT із
використанням esp8266.»**

Виконав (ла):

студент (ка) 4-го курсу Возняк Олег Миколайович
(прізвище, ім'я та по батькові)

(підпис)

Керівник:

к. ф.-м. н., Тащук О.Ю.
(науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент:

Букурос О.В.
(науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

До захисту допущено:

Протокол засідання циклової комісії № _

від „__” _____ 2024 р.

Голова циклової комісії _____ Олександр ТАЩУК

Чернівці, 2024

Завдання та календарний план роботи

Міністерство освіти і науки України
Чернівецький національний університету імені Юрія Федьковича»
Відокремлений структурний підрозділ «Фаховий коледж
Чернівецького національного університету імені Юрія Федьковича»

Затверджую
Голова циклової комісії,
Олександр ТАЩУК.

«___» _____ 2023 р.

Завдання

На кваліфікаційну роботу	Возняк Олег Миколайович
Тема кваліфікаційної роботи	Система паркування автомобілів IoT із використанням esp8266.
Постановка завдання в короткій формі	Моделювання, розробка та монтування Системи паркування автомобілів IoT із використанням esp8266
Вихідні дані (2-3 адреси сайтів, матеріали яких рекомендує керівник кваліфікаційної роботи)	https://www.instructables.com/IOT-Based-Smart-Parking/ https://www.instructables.com/Smart-Parking-System/

Календарний план роботи

Дата отримання версії звіту керівником	Етап виконання роботи	Виконання (зазначає керівник)
10.10.23	Отримання завдання до кваліфікаційної роботи.	
08.11.23	Огляд літератури, присвяченої розробці систем моніторингу стану здоров'я на основі IoT.	
21.12.23	Аналіз існуючих конструкторських рішень. Оформлення розділу 1.	
27.01.24	Вибір компонентної бази, узгодження модулів системи.	
29.02.24	Написання теоретичного матеріалу по використовуваних модулях.	
30.03.24	Проектування та розробка апаратної частини, підключення Оформлення розділу 2.	
28.04.24	Проектування та розробка програмної частини, підключення. Оформлення розділу 3.	

17.05.24	Оформлення та редагування кваліфікаційної роботи.	
10.06.24	Подання роботи на перевірку Інтернет-сервісом <i>Unichesk</i> .	
12.06.24	Подання роботи на рецензію	
13.06.24	Представлення роботи на засіданні Циклової комісії	
<i>за графіком</i>	Захист кваліфікаційної роботи	

Дата видачі завдання _____.

Студент

Олег ВОЗНЯК.

Керівник (П.І.Б)

Олександр ТАЩУК

АНОТАЦІЯ

Возняк Олег Миколайович Система паркування автомобілів IoT із використанням Esp8266. Кваліфікаційна робота.

В роботі виконано моделювання та розробку системи паркування автомобілів на основі технології IoT з використанням ESP8266. Розроблена система базується на концепції IoT, яка дозволяє дистанційно стежити за наявністю вільних місць для паркування автомобілів.

Розвиток міст та збільшення кількості автомобілів створюють значний тиск на інфраструктуру паркування. Відсутність достатньої кількості паркомісць та неефективне їх використання веде за собою підвищення рівня заторів, збільшення викидів CO₂ та втрат часу водіїв у пошуках місця для паркування. Інтелектуальні системи паркування, засновані на технологіях IoT, можуть значно покращити ефективність використання місць для паркування та знизити негативні екологічні наслідки.

Робота складається з 3 розділів, в яких проаналізовано роботу, змодельовано, описано та виконано розробку Системи паркування автомобілів на основі технології інтернету речей на основі ESP8266.

Кваліфікаційна робота містить 41 сторінку, 7 рисунків та 16 посилань на літературні джерела.

Ключові слова: Інтернет речей IoT, системи паркування автомобілів, , Arduino, IDE, ESP 8266, ІЧ датчик, хмарне сховище, сервопривід, RFID-технологій.

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
1. ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Існуючі системи паркування автомобілів	9
1.2 Технології IoT у системах паркування	11
2. ПРОЕКТУВАННЯ АПАРАТНОЇ СКЛАДОВОЇ СИСТЕМИ ПАРКУВАННЯ.....	17
2.1. Вибір компонентної бази	17
2.2. Схема з'єднання та комунікації.....	20
3. ПРОЕКТУВАННЯ ПРОГРАМНОЇ СКЛАДОВОЇ СИСТЕМИ МОНІТОРИНГУ ЗДОРОВ'Я.....	22
3.1 Структура та архітектура хмарного сховища для системи IoT.....	22
3.2. Налаштування та програмування ESP8266.....	25
3.5. Робота системи та інтеграція з веб-інтерфейсом.....	31
ВИСНОВКИ.....	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	39
ДОДАТКИ.....	22
Додаток А. Лістинг програми.....	41
Додаток Б. Схема системи моніторингу здоров'я на основі IoT.....	48
Додаток В. Структурна схема підключення датчиків до плати Arduino Uno.....	49
Додаток Г. Алгоритм роботи пристрою.....	50
Додаток Д. Принципова схема системи моніторингу здоров'я пацієнтів на основі IOT	51
Додаток Ж. Друкована плата системи моніторингу здоров'я пацієнтів на основі IOT.....	52

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

IoT – Інтернет речей.

APS – Автоматизовані системи паркування

ОС – операційна система.

САПР – система автоматизованого проектування.

СП – система приводів.

Wi-Fi – технологія бездротової локальної мережі.

ІЧ датчик – інфрачервоний датчик.

PCB – друкована плата

RFID – Radio Frequency Identification.

LoRaWAN – Long Range Wide Area Network

NB-IoT – Narrowband Internet of Things

ВСТУП

Інтернет речей (IoT) надає нові можливості для розробки інтелектуальних систем паркування. Використання сенсорів, мережевих протоколів та хмарних сервісів дозволяє створювати системи, які можуть в режимі реального часу збирати та аналізувати дані про паркування, надаючи користувачам точну інформацію про наявність вільних місць.

Сучасні системи паркування автомобілів зазнали значного розвитку з початку їх впровадження. Існують декілька типів таких систем, включаючи автоматизовані багаторівневі паркінги, системи з використанням RFID-технологій, а також системи з ультразвуковими датчиками для визначення наявності вільних місць.

Метою роботи є розробка та впровадженні системи паркування автомобілів на основі технології IoT з використанням ESP8266. Основними завданнями є:

Аналіз існуючих рішень та технологій у сфері паркування.

Проектування архітектури системи паркування з використанням ESP8266.

Розробка програмного забезпечення для системи паркування.

Впровадження та тестування системи у реальних умовах.

Об'єктом дослідження є інтелектуальні системи паркування, які використовують технології Інтернету речей. Предметом дослідження є методи та технології розробки системи паркування з використанням мікроконтролера ESP8266 та відповідного програмного забезпечення.

Предметом дослідження є розробка апаратного та програмного забезпечення для збору та відображення даних, а також забезпечення надійності та ефективності розробленої системи.

Для досягнення мети дослідження було використано комплекс методів, включаючи:

Аналіз літератури для вивчення сучасних рішень та технологій.

Проектування та моделювання для розробки архітектури системи.

Програмування та тестування для створення та перевірки працездатності системи.

Емпіричні методи для збору та аналізу даних під час тестування системи у реальних умовах.

Практична цінність полягає в тому що, отримані в роботі результати можна застосувати для подальшої розробки інтелектуальних систем паркування на основі IoT технологій.

Актуальність теми полягає у тому що розвиток міст та збільшення кількості автомобілів створюють значний тиск на інфраструктуру паркування. Відсутність достатньої кількості місць для паркування та неефективне їх використання призводять до підвищення рівня заторів, збільшення викидів CO₂ та втрат часу водіїв у пошуках місця для паркування. Інтелектуальні системи паркування, засновані на технологіях IoT, можуть значно покращити ефективність використання місць паркування та знизити негативні екологічні наслідки.

1. ПОСТАНОВКА ЗАДАЧІ

1.1 Існуючі системи паркування автомобілів

Паркування автомобілів є однією з найбільш важливих проблем у сучасних містах через зростання кількості транспортних засобів та обмеженого простору. Було створено та розроблено багато різних підходів та технологій для оптимізації процесу паркування та підвищення його ефективності. На даний час існують такі системи паркування автомобілів

Традиційні системи паркування включають ручне керування, де водії самостійно шукають вільне місце для паркування. Це може призвести до значних заторів, збільшення витрат палива та підвищення рівня викидів CO₂. Крім того, такий підхід є неефективним з точки зору використання простору та часу [1].

Механічні бар'єри використовуються для контролю доступу до паркувальних місць. Вони забезпечують певний рівень автоматизації, проте їх ефективність обмежена потребою у фізичній присутності обслуговуючого персоналу. Механічні системи паркування включають використання ліфтів, платформ та інших механічних пристроїв для автоматичного переміщення автомобілів до паркувальних місць. Ці системи дозволяють значно підвищити ефективність використання простору, особливо в міських умовах з обмеженою територією. [2].

Автоматизовані багаторівневі паркінги (APS) представляють собою складні механічні конструкції, які використовують механічні системи для переміщення автомобілів на вільні місця. Ці системи дозволяють значно підвищити ефективність використання простору та зменшити необхідність у широких проїздах. Вони можуть включати різні типи ліфтів, конвеєрів та роботизованих платформ [3, 4].

Системи з використанням RFID-технологій RFID (Radio Frequency Identification) технології використовуються для автоматизації процесу ідентифікації транспортних засобів при в'їзді та виїзді з паркувальної зони. Це дозволяє знизити час, необхідний для доступу до паркувального місця, та

покращити загальну ефективність системи. RFID-системи складаються з трьох основних компонентів: RFID-міток, RFID-зчитувачів та центральної бази даних. RFID-системи дозволяють зменшити затори на в'їзді та виїзді з паркувальних зон, оскільки ідентифікація транспортного засобу відбувається автоматично і швидко. Це також знижує потребу в персоналі для контролю доступу та підвищує безпеку паркувальної зони. [5, 6].

Системи з ультразвуковими датчиками використовують ультразвукові хвилі для визначення наявності транспортного засобу на паркувальному місці. Ці датчики встановлюються на паркувальних місцях та забезпечують точну інформацію про їх зайнятість в режимі реального часу [7, 8].

Впровадження систем з ультразвуковими датчиками дозволяє знизити час, необхідний для пошуку вільного місця, зменшити затори та покращити загальну ефективність використання паркувального простору. Такі системи також можуть бути інтегровані з мобільними додатками для надання користувачам інформації про наявність місць у режимі реального часу [9].

Інтелектуальні системи паркування, що використовують технології IoT, забезпечують моніторинг та управління паркувальними місцями в режимі реального часу. Вони включають датчики, комунікаційні мережі та хмарні сервіси для збору та аналізу даних про паркування. Ці системи можуть автоматично інформувати водіїв про наявність вільних місць через мобільні додатки або інформаційні табло [10, 11].

Впровадження таких систем дозволяє значно покращити ефективність використання паркувальних місць та знизити навантаження на транспортну інфраструктуру. Вони також сприяють зниженню рівня викидів CO₂ та економії палива завдяки зменшенню часу, необхідного для пошуку паркувального місця [12].

Основними перевагами таких систем є те що автоматизовані паркувальні системи та системи з ультразвуковими датчиками дозволяють оптимально використовувати доступний простір, зменшуючи потребу у широких проїздах. Інтелектуальні системи IoT та RFID-системи значно

знижують час, необхідний для пошуку вільного місця. Зниження рівня заторів та скорочення часу на пошук місця сприяє зменшенню викидів CO₂ та економії палива.

1.2. Технології IoT у системах паркування

Розвиток технологій Інтернету речей (IoT) здійснив значний вплив на різні сфери, включаючи транспорт та управління паркуванням. IoT дозволяє з'єднувати фізичні пристрої через інтернет, що забезпечує можливість збору, аналізу та обміну даними в режимі реального часу. Це створює нові можливості для оптимізації паркувальних процесів, знижуючи затори, витрати часу та підвищуючи ефективність використання паркувальних місць.

Технології IoT та їх значення для систем паркування

Інтернет речей (IoT) включає в себе мережу фізичних пристроїв, які оснащені датчиками, програмним забезпеченням та іншими технологіями для збору та обміну даними через інтернет. Основні компоненти IoT систем включають:

Датчики та пристрої: Вимірюють фізичні параметри, такі як наявність автомобіля, температура, вологість тощо.

Зв'язок: Забезпечує передачу даних між пристроями та центральною системою обробки даних.

Обробка даних: Включає аналіз зібраних даних для прийняття рішень.

Інтерфейси користувача: Надають доступ до даних та дозволяють керувати системою [13, 14].

Використання IoT у системах паркування має кілька ключових переваг а саме IoT дозволяє отримувати та обробляти дані про наявність паркувальних місць у реальному часі, завдяки аналітиці даних можна оптимізувати розподіл паркувальних місць, знижуючи затори та час пошуку.

Зручність: Інформація про наявні місця може бути надана користувачам через мобільні додатки або інформаційні табло.

Архітектура IoT систем паркування Архітектура IoT систем паркування складається з різних компонентів, які працюють разом для створення ефективного та інтелектуального рішення для управління паркуванням. Основні елементи включають датчики для виявлення транспортних засобів, комунікаційні протоколи для передачі даних, центральні системи обробки даних та інтерфейси користувача для надання інформації про наявність паркувальних місць.

Датчики є ключовими елементами IoT систем паркування, оскільки вони забезпечують збір даних про наявність транспортних засобів на паркувальних місцях. Вони встановлюються на паркувальних місцях та виявляють присутність або відсутність транспортного засобу. Існує кілька типів датчиків, які можуть бути використані в таких системах:

Ультразвукові датчики: Вимірюють відстань до об'єкта, за допомогою ультразвукових хвиль. Вони встановлюються на місцях парковки і здатні точно визначати наявність автомобіля.

Інфрачервоні датчики виявляють об'єкти за допомогою інфрачервоного випромінювання. Вони використовуються для визначення наявності транспортного засобу на паркувальному місці.

Індукційні датчики: виявляють зміни у магнітному полі, спричинені металевими об'єктами, такими як автомобілі. Вони можуть бути встановлені під поверхнею паркувальних місць і забезпечувати високу точність визначення наявності транспортних засобів.

Комунікаційні протоколи забезпечують передачу даних від датчиків до центральної системи управління. Вони включають як бездротові, так і дротові технології:

Wi-Fi є поширеним протоколом для передачі даних на невеликих відстанях з високою швидкістю. Він забезпечує надійне з'єднання та може бути використаний для підключення датчиків до локальної мережі.

LoRaWAN (Long Range Wide Area Network) призначений для передачі даних на великі відстані з низьким енергоспоживанням. Це робить його

ідеальним для використання в міських умовах, де важливо забезпечити покриття великої території.

NB-IoT (Narrowband Internet of Things) забезпечує надійну передачу даних з низькою затримкою та високою пропускну здатністю. Це робить його підходящим для використання у міських середовищах з великою кількістю пристроїв.

Обробка та аналіз даних Зібрані дані з датчиків передаються на центральні сервери або у хмару для подальшого аналізу. Це дозволяє виявляти закономірності, прогнозувати попит на паркувальні місця та надавати рекомендації користувачам. Основні методи обробки даних включають:

Аналіз даних у реальному часі: Включає використання стрімінгових платформ для обробки потоків даних у реальному часі, що дозволяє оперативно реагувати на зміни та надавати актуальну інформацію користувачам.

Машинне навчання: Дозволяє прогнозувати наявність паркувальних місць на основі історичних даних та поведінкових моделей це забезпечує точніші прогнози та покращує загальну ефективність системи.

Інтерфейси користувача надають доступ до даних та дозволяють керувати системою. Вони можуть включати мобільні додатки, веб-інтерфейси та інформаційні табло, які надають користувачам інформацію про наявність паркувальних місць у режимі реального часу.

Мобільні додатки дозволяють користувачам переглядати наявність паркувальних місць, резервувати місця та отримувати навігаційні вказівки до найближчого вільного місця.

Веб-інтерфейси забезпечують доступ до системи управління паркуванням через інтернет. Вони можуть бути використані адміністраторами для моніторингу та управління паркувальними місцями.

Інформаційні табло встановлюються на вулицях або в паркувальних зонах і надають водіям інформацію про наявність вільних місць. Це допомагає знизити затори та покращити загальну ефективність паркування.

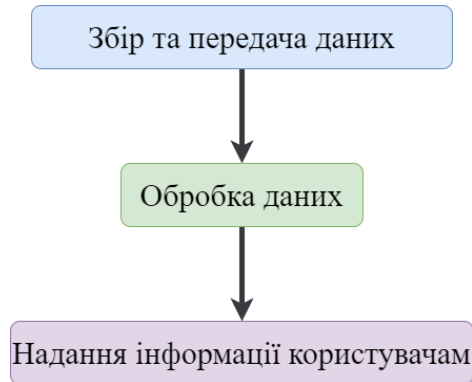


Рисунок 1.1. Взаємодія компонент системи Smart паркінгу

Принцип роботи системи паркування автомобілів на основі IoT технологій передбачає наступні етапи виконання збір та передача даних датчики встановлюються на паркувальних місцях та зчитують інформацію про наявність автомобілів. Зібрані дані передаються через комунікаційні протоколи (Wi-Fi, LoRaWAN, NB-IoT) на центральні сервери або хмарні платформи далі здійснюється обробка даних де центральні сервери або хмарні платформи обробляють зібрані дані, аналізують їх та зберігають у базах даних. Аналіз даних може включати виявлення закономірностей, прогнозування попиту на місця паркування та генерацію звітів. Після чого здійснюється надання інформації користувачам тобто

Інформація про наявність місця паркування надається користувачам через мобільні додатки, веб-інтерфейси або інформаційні табло. Це дозволяє водіям швидко знаходити вільні місця, резервувати їх та отримувати навігаційні вказівки.

Багато міст по всьому світу вже впровадили інтелектуальні системи паркування, що базуються на IoT. Наприклад, Сан-Франциско впровадив систему SFPark, яка використовує датчики для моніторингу наявності місць для паркування та надання цієї інформації користувачам через мобільний додаток. [15]

Кілька платформ та технологій було розроблено для забезпечення інтеграції IoT у системи паркування:

ParkMe: Надає дані про доступність паркувальних місць у реальному часі через мобільний додаток.

Streetline: Використовує датчики та аналітику даних для моніторингу та управління паркувальними місцями.

Використання IoT дозволяє оптимізувати використання паркувальних місць та знижувати час пошуку місця. Інтелектуальні системи можуть зменшити затори, спричинені пошуком паркувальних місць та зменшити затори та часу на пошук місця знижує рівень викидів CO₂.

Однак і виникає ряд проблем у використанні Smart паркінгів це висока вартість впровадження адже встановлення датчиків та забезпечення надійного з'єднання вимагає значних витрат.

Безпека та приватність: Забезпечення безпеки даних та приватності користувачів є важливим аспектом.

Технічні обмеження: Деякі технології, такі як бездротові сенсори, можуть мати обмежений діапазон або страждати від інтерференції.

Майбутнє систем паркування з використанням IoT пов'язане з впровадженням нових технологій, таких як штучний інтелект, блокчейн та 5G. Штучний інтелект може забезпечити більш точне прогнозування попиту на паркувальні місця та оптимізацію їх використання. Блокчейн може забезпечити безпеку транзакцій та зберігання даних. 5G забезпечить швидкі та надійні з'єднання для передачі великого обсягу даних у реальному часі.

Збільшення кількості впроваджень IoT у системах паркування можна очікувати не тільки у великих містах, але й у менших населених пунктах. Це забезпечить рівномірний розподіл технологічних переваг та підвищить загальну ефективність транспортної системи.

Технології IoT значно впливають на розвиток систем паркування, забезпечуючи підвищену ефективність, зручність та екологічні переваги. Впровадження IoT у системи паркування стикається з певними викликами,

такими як висока вартість та проблеми безпеки, проте перспективи цієї технології обіцяють значні поліпшення у транспортній інфраструктурі. Майбутні дослідження та інновації у цій галузі сприятимуть подальшому розвитку та вдосконаленню інтелектуальних паркувальних систем.

2. ПРОЕКТУВАННЯ АПАРАТНОЇ СКЛАДОВОЇ СИСТЕМИ ПАРКУВАННЯ

2.1. Вибір компонентної бази

Створення системи моніторингу місця для паркування автомобілів на основі ІОТ здійснюється на основі Wi-Fi модуля Esp8266, п'яти ІЧ-датчиків і двох серводвигунів. Два ІЧ-датчики розміщуються на в'їзді та виїзді для виявлення автомобіля, а три ІЧ-датчики використовуються для визначення наявності місця для паркування. Серводвигуни використовуються для відкриття та закриття воріт відповідно до значення датчика. У якості хмарного сховища для відображення та контролю даних використовується платформа Adafruit IO.

Мікроконтролер ESP8266

NodeMCU ESP8266 — це мікропрограмне забезпечення та плата розробки з відкритим вихідним кодом, яка спеціально призначена для роботи з додатками на основі Інтернету речей Рис. 2.1..

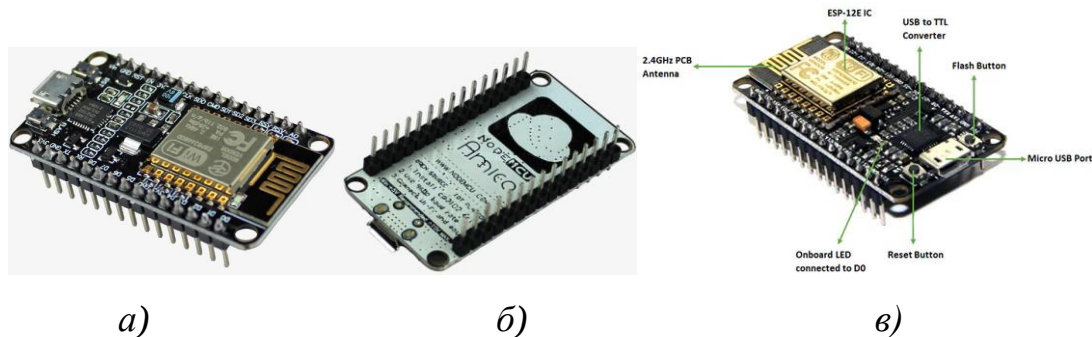


Рисунок. 2.1. Плата Модуля Wi-Fi ESP 8266. а) Вигляд зверху; б) Вигляд з низу; в) Розпіновка

Компоненти та технічні характеристики

Мікроконтролер ESP8266, володіє 32-бітним процесором Tensilica L106, працюючим на частоті 80 або 160 МГц., він має один аналоговий вхідний контакт (ADC) з діапазоном напруг живлення від 0 до 1 В, сама плата живиться від USB порту (5В) або через пін VCC (3.3В) також є регулятор напруги, який знижує напругу з 5 до 3.3В, Кількість цифрових

виходів (пінів) введення/виведення складає 17 GPIO (General Purpose Input/Output) із можливістю підключення до протоколів послідовного зв'язку UART (Universal Asynchronous Receiver-Transmitter), SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), PWM (Pulse Width Modulation).

Пам'ять ESP8266 містить 128 КБ оперативної пам'яті 64 КБ інструкційного кешу, 96 КБ даних SRAM та флеш-пам'ять від 512 КБ до 4 МБ, також він має Вбудований Wi-Fi модуль: 802.11 b/g/n.

Функціонування та застосування

ESP8266 включає мікропрограмне забезпечення, яке працює на SoC ESP8266 Wi-Fi від Espressif Systems що дозволяє ESP функціонувати як Wi-Fi клієнт або точка доступу (AP). Він підтримує протоколи TCP/IP, UDP, HTTP, MQTT, апаратне забезпечення, базується на модулі ESP-12, через що ESP 8266 програмується за допомогою Arduino IDE. Прошивка NodeMCU включає Lua інтерпретатор, що також дозволяє програмувати плату мовою скриптів Lua.

Висока обчислювальна потужність NodeMCU із вбудованим Wi-Fi/Bluetooth і функціями Deep Sleep Operating робить його ідеальним для IoT-проектів таких як: Розумні будинки де ESP8266 використовується для автоматизації систем освітлення, опалення, безпеки та інших систем у домі. Дистанційний контроль та моніторинг завдяки Wi-Fi модулю, плата може збирати дані з датчиків та відправляти їх на сервери або мобільні додатки для моніторингу в режимі реального часу. Вимірювання та збирання даних для створення метеостанцій або систем моніторингу навколишнього середовища.

Модуль інфрачервоного датчика/датчика наближення

Модуль ІЧ-датчика Рис.2.2. складається зі світлодіода та ІЧ-передавача, який пропускає ІЧ-світло. Потім це світло вловлюється світлодіодом ІЧ-приймача при відбиванні його від об'єкта. ІЧ-датчика зазвичай використовується в роботах, що стежать за лінією, датчиках наближення, виявленні об'єктів

Він має три штирі, з яких два використовуються для живлення датчика,

а третій штифт дає вихід як високий/низький на основі близькості об'єкта перед ним. Модуль датчика складається з операційного підсилювача LM358, інфрачервоного світлодіода, фотодіода, потенціометра 10K для регулювання чутливості датчика, світлодіодного індикатора, 2 резисторів по 330 Ом і 10K Ом.

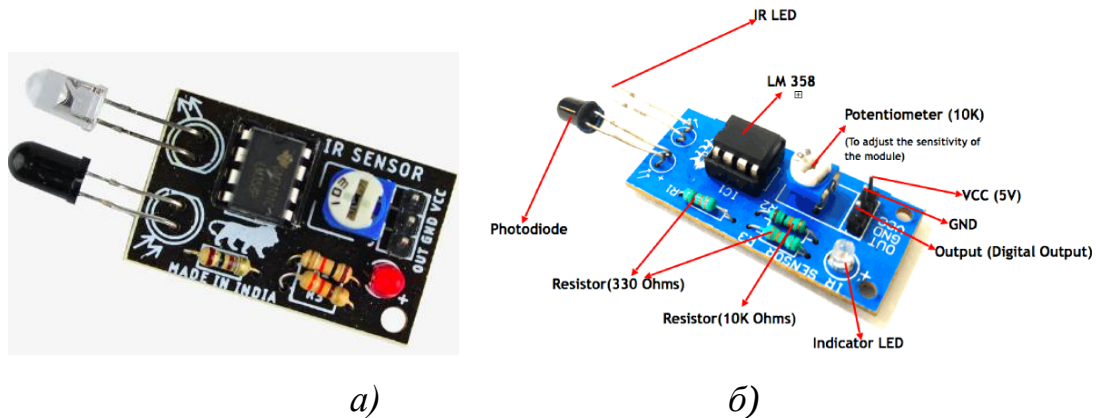


Рисунок. 2.2. Плата Модуля ІЧ-датчика а) Вигляд зверху; б) Розпіновка

Серводвигун Tower Pro SG90 9g

Серводвигуни використовуються для програм керування, які вимагають точного керування, наприклад позиціонування руки робота, положення інструменту в обробному обладнанні. Серводвигуни зазвичай забезпечують керування в діапазоні 180°. Це керування кутовим положенням виконується методом ШІМ, тому, змінюючи його робочий цикл, можна контролювати кутове положення двигуна. Робочий діапазон напруг SG90 9g від 3 В до 7,2 В. Серводвигун може піднімати максимум 1,6 кг, якщо підвішений на відстані 1 см від валу а його. Його також можна використовувати в роботах, верстатах з ЧПК, системах рульового керування на автомобілях з радіоуправлінням та інших роботах або системах автоматизації.



Рисунок. 2.3. Серводвигун Tower Pro SG90 9g

2.2. Схеми з'єднання та комунікацій.

У даній системі паркування, ми використовуємо чотири ІЧ-датчиків і два серводвигуни. ІЧ-датчики та сервомотори підключені до NodeMCU, який контролює весь процес і надсилає інформацію про наявність і час паркування в Adafruit IO, щоб можна було контролювати його з будь-якої точки світу за допомогою цієї платформи.

Два ІЧ-датчики використовуються на в'їзних і виїзних воротах, щоб вони могли виявляти автомобілі на в'їзді та виїзді та автоматично відкривати і закривати ворота.

Принцип роботи полягає у тому що ІЧ-хвилі, які випромінюються ІЧ-світлодіодом, потрапляють на об'єкт (розміщений на шляху передачі ІЧ-хвиль) і відбиваються назад. Фотодіод розміщується таким чином, що він сприймає відбиті ІЧ-хвилі при цьому його опір зменшується і діє як коротке замикання. Струм починає надходити в R2, і на неінвертуючому терміналі з'являється напруга. Якщо напруга на неінвертуючому терміналі більша, ніж напруга на інвертуючому терміналі, операційний підсилювач дає високий вихід напруг (5 В), а світлодіод починає світитися напругу на інвертуючому терміналі змінюється, за рахунок зміни опору потенціометра. Це робиться для зміни чутливості датчика. Якщо значення потенціометра дорівнює нулю, навіть до того, як фотодіод сприймає ІЧ-хвилі, вихід ОР -amp буде високим, це означає, що датчик має високу чутливість.

У випадку коли перед ІЧ-світлодіодом немає об'єкта ІЧ-хвилі, що випромінюють ІЧ-світлодіод, не досягнуть фотодіода, оскільки немає жодного об'єкта, який міг би відбити ІЧ-хвилі. Завдяки цьому фотодіод буде подавати максимальний опір напруги живлення Vcc. Оскільки струм, що протікає через резистор, мінімальний, напруга на неінвертуючому виводі буде мінімальною. Оскільки напруга на неінвертуючому терміналі менша за напругу на інвертуючому терміналі, вихідний сигнал операційного підсилювача буде низьким і складати 0 В.

Два серводвигуни використовуються як в'їзні та виїзні ворота, тому

щоразу, коли інфрачервоний датчик виявляє автомобіль, серводвигун автоматично повертається від 45° до 140°, і після затримки повертається у вихідне положення. Інші два інфрачервоних датчика використовуються для виявлення наявності або зайнятості місця для паркування та надсилання даних до NodeMCU.

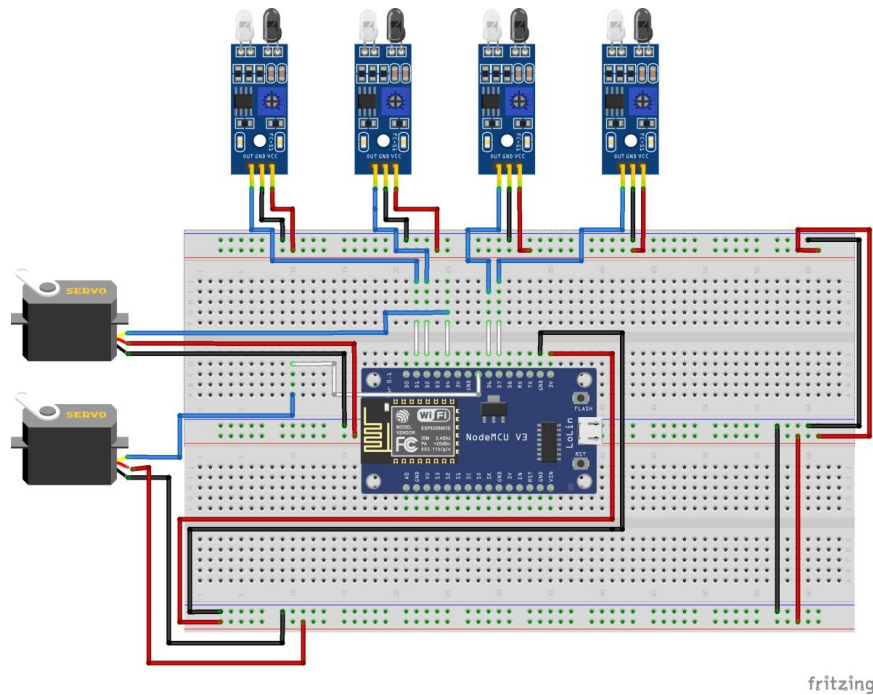


Рисунок. 2.4. Макетна схема Системи паркування

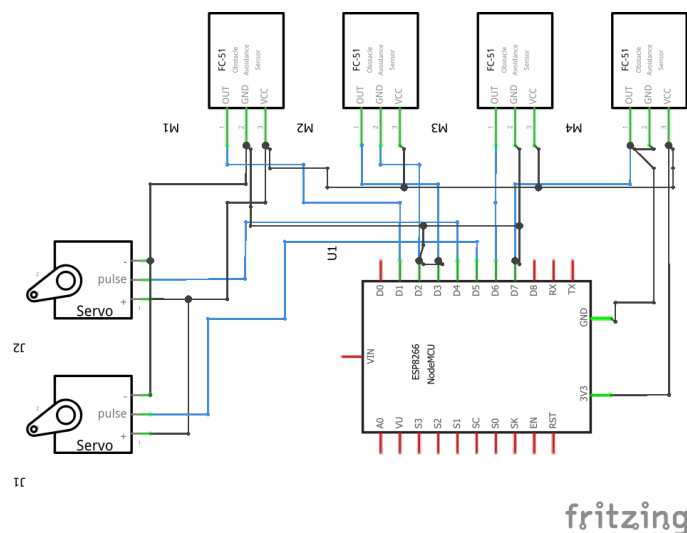


Рисунок. 2.4. Принципова схема системи паркування

ПРОЕКТУВАННЯ ПРОГРАМНОЇ СКЛАДОВОЇ СИСТЕМИ МОНІТОРИНГУ ЗДОРОВ'Я

3.1 Структура програмного забезпечення та хмарного сховища для системи IoT

Adafruit IO – це хмарна платформа для Інтернету речей (IoT), розроблена компанією Adafruit. Вона надає зручний спосіб збору, зберігання та візуалізації даних від різноманітних IoT пристроїв.

Adafruit IO дозволяє створювати налаштовані приладні панелі для візуалізації даних у режимі реального часу, також можна використовувати різні віджети, такі як графіки, діаграми, текстові блоки, кнопки та інші елементи для відображення та керування даними. Наявність каналів feeds у даній платформі дають змогу зберігати та здійснювати організацію даних де кожен канал містить дані, отримані від певного сенсора або пристрою.

Adafruit IO володіє такими протоколами зв'язку як MQTT, який є легковаговим і широко використовуваним у IoT. Крім MQTT, платформа також підтримує і HTTP протоколи для обміну даними також що забезпечує безпечний зв'язок через SSL/TLS, та гарантує захищеність даних при передачі. Для доступу до платформи та взаємодії з нею використовуються API ключі, що підвищує рівень безпеки.

Платформа підтримує інтеграцію з IFTTT (If This Then That), це дозволяє автоматизувати різні процеси на основі подій а інтеграція із Zapier надає додаткові можливості для автоматизації завдань та обміну даними з іншими сервісами.

Adafruit IO також дозволяє додавати та керувати різними пристроями через єдиний інтерфейс тобто надає можливість керувати пристроями віддалено через приладні панелі та мобільні додатки.

Платформа орієнтована на розробників та ентузіастів, що робить її простою у використанні навіть для новачків. Завдяки підтримці різних протоколів і інтеграцій, вона є дуже гнучкою платформою, яка володіє зручними інструментами для візуалізації даних це дозволяє швидко

отримувати потрібну інформацію. Також Adafruit IO пропонує безкоштовний тарифний план, що робить його доступним для невеликих проектів та тестування.

Архітектура Adafruit IO

Архітектура Adafruit IO Рис. 3.1., побудована таким чином, щоб забезпечити ефективний збір, зберігання, обробку та візуалізацію даних IoT пристроїв.

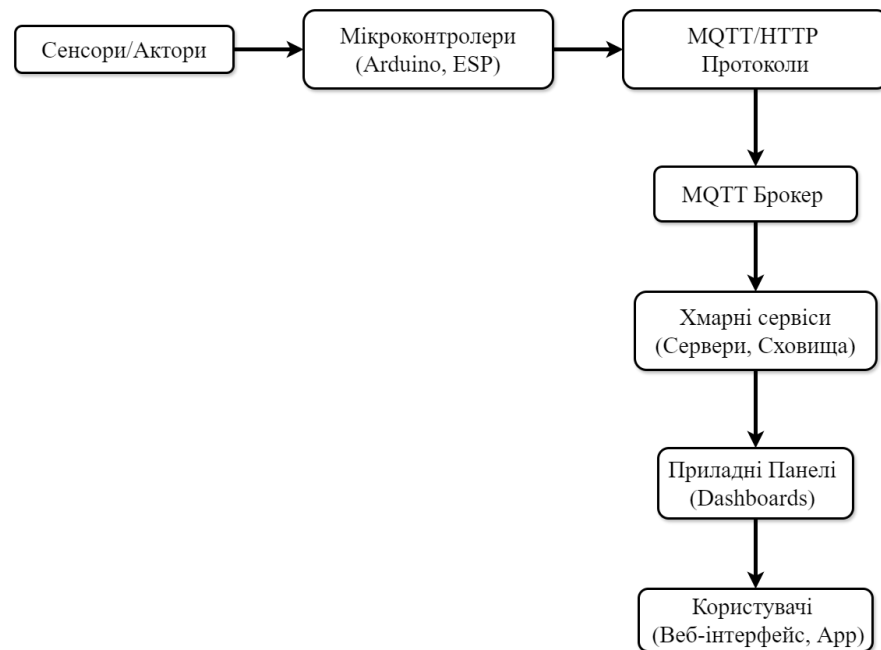


Рисунок. 3.1. Блок-схема архітектури Adafruit IO.

Основні компоненти архітектури включають такі елементи:

1. Клієнт (Пристрої) це елементи які використовуються для збору даних (сенсори) та виконавчі механізми.

Мікроконтролери та платформи це ряд пристроїв, Arduino, ESP8266, ESP32 та Raspberry Pi, які використовуються для збору та передачі даних на хмарну платформу такі як

2. Протоколи зв'язку Основний протокол для передачі даних між пристроями та платформою. MQTT брокер обробляє повідомлення від клієнтів та доставляє їх до відповідних каналів.

HTTP API: Використовується для передачі даних за допомогою стандартних веб-запитів, забезпечуючи альтернативу для пристроїв, які не підтримують MQTT.

3. Хмарні сервіси

Сервери та Сховища даних: Використовуються для зберігання даних, отриманих від пристроїв. Сховище даних забезпечує зберігання великих обсягів інформації з можливістю доступу до історичних даних.

Обробка та Аналіз даних: Використання серверів для обробки та аналізу даних у реальному часі, забезпечуючи їх актуальність та корисність.

4. Приладні панелі (Dashboards)

Інтерфейс користувача: Веб-інтерфейс для створення та налаштування приладних панелей, що дозволяє візуалізувати дані у вигляді графіків, таблиць, діаграм та інших віджетів.

Мобільний додаток: Додаток для iOS та Android, який надає можливість моніторингу та керування пристроями зі смартфона.

5. Канали (Feeds)

Організація даних: Канали використовуються для організації та структурування даних, що надходять від різних пристроїв.

Збереження та доступ: Забезпечення збереження даних у хмарі з можливістю доступу до них через веб-інтерфейс або API.

6. Безпека

SSL/TLS: Забезпечення захисту даних при передачі через Інтернет за допомогою шифрування SSL/TLS.

Аутентифікація та Авторизація: Використання API ключів для контролю доступу до ресурсів платформи.

7. Автоматизація та Інтеграції

IFTTT та Zapier: Інтеграція з платформами автоматизації, що дозволяє створювати складні робочі процеси на основі подій.

Webhooks: Використання webhooks для автоматичного виклику зовнішніх сервісів при настанні певних подій.

3.2 Налаштування та програмування ESP8266

Програмування системи передбачає збір, відображення та передачу даних про кількість вільних місць для паркування автомобілів на основі NodeMCU ESP8266 та ІЧ датчиків.

Програмування ESP8266 здійснюється на мові С чи С++, також запрограмувати NodeMCU можна за допомогою Arduino IDE. Для цього спочатку потрібно завантажити всі необхідні бібліотеки, для ESP8266 Wi-Fi і Servo.h вони вже встановлені в IDE

Основні компоненти коду:

1. Імпорт бібліотек та ініціалізація об'єктів:

Підключаємо бібліотеки для роботи з клієнт NTP і бібліотеки Adafruit MQTT.

```
#include <ESP8266WiFi.h>
#include <Servo.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <NTPClient.h>;
#include <WiFiUdp.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
```

2. Оголошення змінних

Потім додаємо облікові дані Wi-Fi та Adafruit IO, які можна скопіювати з сервера Adafruit IO. Вони включатимуть сервер MQTT, номер порту, ім'я користувача та ключ AIO

```
const char *ssid = "Ім'я WiFi"; // Введіть своє ім'я WiFi
const char *pass = "Пароль"; // Введіть пароль WiFi
Рядок time1= "89";
#define MQTT_SERV "io.adafruit.com"
#define MQTT_PORT 1883
#define MQTT_NAME "Ім'я користувача"
#define MQTT_PASS "Ключ AIO"
```

3. Налаштування канал:

Налаштовуємо канал, у якому буде здійснюватися публікація даних. У даному випадку Agriculture Data — це назва каналу.

```

Adafruit_MQTT_Subscribe          EntryGate          =
Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME "/f/EntryGate");
Adafruit_MQTT_Subscribe          ExitGate           =
Adafruit_MQTT_Subscribe(&mqtt, MQTT_NAME "/f/ExitGate");

```

4. Підключення серводвигунів

Підключаємо серводвигун входу та виходу до контактів D4, D5 NodeMCU та вибираємо вихідні контакти ІЧ-сенсора як INPUT.

```

myservo.attach(D4);
myservos.attach(D5);
pinMode(carExited, INPUT);
pinMode(carEnter, INPUT);
pinMode(слот1, BX1Д);
pinMode(слот2, BX1Д);
pinMode(slot3, INPUT);

```

Усередині циклу void функція timeClient.update() використовується для оновлення дати й часу кожного разу, коли надсилається запит на сервери NTP . Після отримання даних зберігаємо годину, хвилину та секунду в трьох різних цілих числах. ThingSpeak.

```

timeClient.update();
hh = timeClient.getHours();
mm = timeClient.getMinutes();
ss = timeClient.getSeconds();

```

Далі цифровим способом здійснюється зчитування контакту інфрачервоного датчика входу та виходу та перевіряється, чи вони високі.

Якщо штифти високі, потрібно перемістити серводвигун, щоб відкрити вхідні та виїзні ворота. Потім можна збільшувати кількість для входу та зменшувати кількість для виходу та здійснювати публікацію даних на інформаційній панелі Adafruit IO.

```

if (entrysensor == 1) { // якщо високий, тоді підрахувати та
надіслати дані кількість= кількість+1; //збільшити кількість
myservos.write(OPEN_ANGLE);
delay(3000);
myservos.write(CLOSE_ANGLE);
}

```

```

    if (exitsensor == 1) { //якщо високий, тоді підрахувати та
надіслати    кількість=    кількість-1;    //зменшення    кількості
myservo.write(OPEN_ANGLE);
    delay(3000);
    myservo.write(CLOSE_ANGLE);
}
if (! CarsParked.publish(count)) {}

```

Перевіряємо ІЧ-датчик слота 1. Якщо значення «1», а логічна функція — false, отримуємо час входу з сервера NTP і зберігаємо його в змінній EntryTimeSlot1. Публікуємо дані змінних у каналі Adafruit IO.

```

if (s1 == 1 && s1_occupied == false) {
    EntryTimeSlot1 = h + " : " + m;
    s1_occupied = true;
    if (! EntrySlot1.publish ((char*)
EntryTimeSlot1.c_str())){}
}

```

Якщо контакт інфрачервоного датчика змінюється на нуль і функція Boolean є істинною, Публікується час виходу в каналі введення-виведення Adafruit.

```

if(s1 == 0 && s1_occupied == true) {
    ExitTimeSlot1 = h + " : " + m;
    s1_occupied = false;
    if (! ExitSlot1.publish((char*)
ExitTimeSlot1.c_str())){}
}

```

Виконуємо такі ж кроки для датчиків slot2 і slot 3.

```

if (s2 == 1&& s2_occupied == false) {
    EntryTimeSlot2 = h + " : " + m;
    s2_occupied = true;
    if (! EntrySlot2.publish((char*)
EntryTimeSlot2.c_str())){}
}
if(s2 == 0 && s2_occupied == true) {
    ExitTimeSlot2 = h + " : " + m;
    s2_occupied = false;
    if (! ExitSlot2.publish((char*)
ExitTimeSlot2.c_str())){}
}
if (s3 == 1&& s3_occupied == false) {
    EntryTimeSlot3 = h + " : " + m;
    s3_occupied = true;
    if (! EntrySlot3.publish((char*)

```

```

EntryTimeSlot3.c_str())){}
}
if(s3 == 0 && s3_occupied == true) {
    ExitTimeSlot3 = h + " : " + m;
    s3_occupied = false;
    if (! ExitSlot3.publish((char*) ExitTimeSlot3.c_str())){
}
}
}

```

Перевіряємо конкретне слово в каналі, на який ми підписалися, і якщо слово збігається з вказаним словом, тобто «ON», воно обертатиме серводвигун, щоб відкрити ворота.

```

if (subscription == &EntryGate)
{
    Serial.println((char*) EntryGate.lastread);
    if (!strcmp((char*) EntryGate.lastread, "ON"))
    {
        myservos.write(OPEN_ANGLE);
        delay(3000);
        myservos.write(CLOSE_ANGLE);
    }
}

```

ВИСНОВКИ

Інтернет речей має великий потенціал для подальшого розвитку в медичній галузі. Зокрема, розвиток технологій 5G, штучного інтелекту (AI) та машинного навчання відкриває нові можливості для створення інтелектуальних систем моніторингу здоров'я. Однією з ключових тенденцій є розвиток периферійних обчислень (edge computing), які дозволяють обробляти дані безпосередньо на пристроях або у їхньому безпосередньому оточенні, зменшуючи затримки та підвищуючи ефективність систем.

У сучасному світі моніторинг стану здоров'я є важливою проблемою. Пацієнти страждають від серйозних проблем зі здоров'ям через відсутність належного контролю за здоров'ям. Зараз доступно багато пристроїв IoT для моніторингу здоров'я пацієнтів через Інтернет. Медичні працівники також використовують ці розумні пристрої, щоб стежити за своїми пацієнтами. IoT стрімко революціонізує галузь охорони здоров'я, завдяки сотням нових стартапів у галузі охорони здоров'я, які використовують систему моніторингу здоров'я пацієнтів

У роботі створено систему моніторингу стану здоров'я на основі IOT, яка реєструє частоту пульсу пацієнта та температуру навколишнього середовища. Ця система не тільки записує ці дані, але й оновлює їх на платформі IOT.

Платформою IOT вибрано Thing Speak. Thing Speak — це програма з відкритим кодом Інтернету речей (IoT) і API для зберігання та отримання даних із речей за допомогою протоколу HTTP через Інтернет або через локальну мережу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Shoup, D., "The High Cost of Free Parking," *Journal of Planning Education and Research*, vol. 17, no. 1, pp. 3-20, 1997.
2. McDonald, M., "The Evolution of Parking Management Systems: A Retrospective," *International Journal of Intelligent Transportation Systems Research*, vol. 12, no. 1, pp. 1-9, 2014.
3. Kato, T., "Automated Parking Systems: Trends and Technology," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 10, pp. 4561-4569, 2013.
4. Lim, S. H., & Kachroo, P., "Automated Parking System with Efficient Space Utilization and High Throughput," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1680-1687, 2018.
5. Samaras, K., & Bouras, C., "A Secure RFID-Based Parking Management System," *Journal of Network and Computer Applications*, vol. 36, no. 3, pp. 1181-1190, 2013.
6. Wahab, H. A., Talib, M. F., Kamarudin, L. M., & Ahmad, R. B., "RFID-based System for Parking Management Using WiMAX," *International Journal of Engineering and Technology*, vol. 6, no. 2, pp. 85-89, 2014.
7. Kim, E. K., & Lee, S. H., "Development of a Smart Parking System Based on Wireless Sensor Networks," *Sensors*, vol. 17, no. 9, p. 1999, 2017.
8. Abdullah, N. H., Anisi, M. H., & Razak, S. A., "A Survey on Smart Parking System Based on Internet of Things," *Journal of Network and Computer Applications*, vol. 177, p. 102857, 2021.
9. Idris, M. Y. I., Leng, Y. Y., Tamil, E. M., Noor, N. M., & Razak, Z., "Car Park System: A Review of Smart Parking System and Its Technology," *Information Technology Journal*, vol. 8, no. 2, pp. 101-113, 2009.
10. Geng, Y., & Cassandras, C. G., "A New 'Smart Parking' System

- Infrastructure and Implementation," *Procedia - Social and Behavioral Sciences*, vol. 54, pp. 1278-1287, 2012.
11. Zheng, Y., Rajasegarar, S., & Leckie, C., "Parking Availability Prediction for Sensor-enabled Car Parks in Smart Cities," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 4266-4275, 2019.
 12. Zeng, Y., Zhang, L., & Shen, H., "A Survey of Intelligent Transportation Systems," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5834610, 2018.
 13. Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M., "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22-32, 2014.
 14. Atzori, L., Iera, A., & Morabito, G., "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, 2010.
 15. Polak, J., & Axhausen, K. W., "Parking Search Behaviour and its Impacts on the Design of Parking Guidance Systems," *Transport Policy*, vol. 8, no. 1, pp. 13-29, 2001.

ДОДАТКИ

Додаток А

Лістинг програми

```
#include <ESP8266WiFi.h>
#include <Servo.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <NTPClient.h>;
#include <WiFiUdp.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
const char *ssid = "Galaxy-M20"; // Enter your WiFi Name
const char *pass = "ac312124"; // Enter your WiFi Password

#define MQTT_SERV "io.adafruit.com"
#define MQTT_PORT 1883
#define MQTT_NAME "aschoudhary"
#define MQTT_PASS "1ac95cb8580b4271bbb6d9f75d0668f1"
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 19800,60000);
Servo myservo; //servo as gate
Servo myservos; //servo as gate

int carEnter = D0; // entry sensor
int carExited = D2; //exi sensor
int slot3 = D7;
int slot2 = D6;
int slot1 = D3;
int count =0;
int CLOSE_ANGLE = 80; // The closing angle of the servo motor arm
int OPEN_ANGLE = 0; // The opening angle of the servo motor arm
int hh, mm, ss;
int pos;
int pos1;

String h, m,EntryTimeSlot1,ExitTimeSlot1,
EntryTimeSlot2,ExitTimeSlot2, EntryTimeSlot3,ExitTimeSlot3;
boolean entrysensor, exitsensor,s1,s2,s3;
```

					Система паркування автомобілів ІoT із використанням esp8266						Лім.		Маса		Масштаб	
Змн.	Арк.	№ докум.		Підпис							Дата					
Розроб.		Возняк О.М.														
Перевір.		Ташук О.Ю.														
Т. Контр.												Арк.		Аркушів		1
Реценз.												Циклова комісія Комп'ютерної інженерії				
Н. Контр.																
Затверд.		Ташук О.Ю.														


```

        if (exitsensor == 1) {                                     //if high
then count and send
        count= count-1;
//decrement count
        myservo.write(OPEN_ANGLE);
        delay(3000);
        myservo.write(CLOSE_ANGLE);
    }
    if (! CarsParked.publish(count)) {}

    if (s1 == 1 && s1_occupied == false) {
        Serial.println("Occupied1 ");
        EntryTimeSlot1 = h + " :" + m;
        //Serial.print("EntryTimeSlot1");
        //Serial.print(EntryTimeSlot1);

        s1_occupied = true;
        if (! EntrySlot1.publish((char*)
EntryTimeSlot1.c_str())){}
    }
    if(s1 == 0 && s1_occupied == true) {
        Serial.println("Available1 ");
        ExitTimeSlot1 = h + " :" + m;
        //Serial.print("ExitTimeSlot1");
        //Serial.print(ExitTimeSlot1);

        s1_occupied = false;
        if (! ExitSlot1.publish((char*)
ExitTimeSlot1.c_str())){}
    }
    if (s2 == 1&& s2_occupied == false) {
        Serial.println("Occupied2 ");
        EntryTimeSlot2 = h + " :" + m;
        //Serial.print("EntryTimeSlot2");
        //Serial.print(EntryTimeSlot2);

        s2_occupied = true;
        if (! EntrySlot2.publish((char*)
EntryTimeSlot2.c_str())){}
    }

```

					Система паркування автомобілів ІoT із використанням esp8266	Лім.			Маса		Масштаб	
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Возняк О.М.										
Перевір.		Ташук О.Ю.										
Т. Контр.						Арк.			Аркушів 1			
Реценз.						Циклова комісія Комп'ютерної інженерії						
Н. Контр.												
Затверд.		Ташук О.Ю.										

```

if(s2 == 0 && s2_occupied == true) {
    Serial.println("Available2 ");
    ExitTimeSlot2 = h + " :" + m;
    //Serial.print("ExitTimeSlot2");
    //Serial.print(ExitTimeSlot2);

    s2_occupied = false;
    if (! ExitSlot2.publish((char*)
ExitTimeSlot2.c_str())){}
}
if (s3 == 1&& s3_occupied == false) {
    Serial.println("Occupied3 ");
    EntryTimeSlot3 = h + " :" + m;
    //Serial.print("EntryTimeSlot3: ");
    //Serial.print(EntryTimeSlot3);
    s3_occupied = true;
    if (! EntrySlot3.publish((char*)
EntryTimeSlot3.c_str())){}
}
if(s3 == 0 && s3_occupied == true) {
    Serial.println("Available3 ");
    ExitTimeSlot3 = h + " :" + m;
    //Serial.print("ExitTimeSlot3: ");
    //Serial.print(ExitTimeSlot3);
    s3_occupied = false;
    if (! ExitSlot3.publish((char*) ExitTimeSlot3.c_str())){
}
}

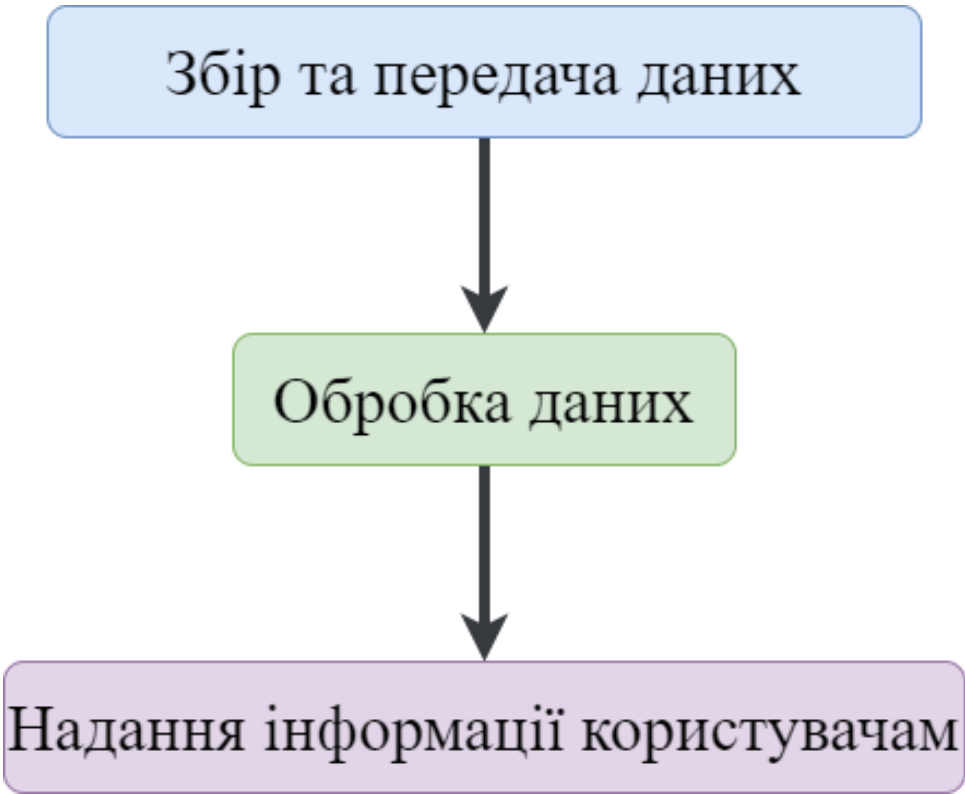
Adafruit_MQTT_Subscribe * subscription;
while ((subscription = mqtt.readSubscription(5000)))
{

if (subscription == &EntryGate)
{
    //Print the new value to the serial monitor
    Serial.println((char*) EntryGate.lastread);
}
}

```

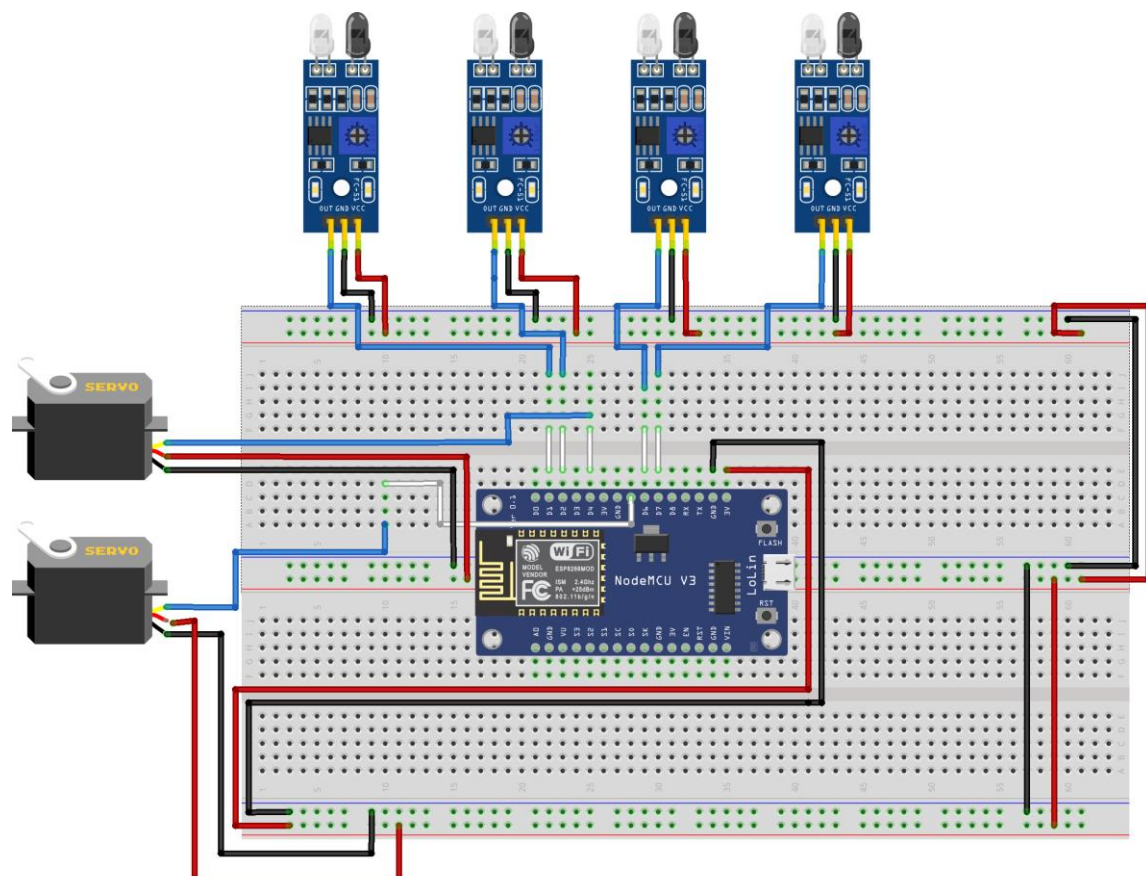
					Система паркування автомобілів IoT із використанням esp8266	Лім.			Маса		Масштаб	
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Возняк О.М.										
Перевір.		Тащук О.Ю.										
Т. Контр.						Арк.			Аркушів			1
Реценз.						Циклова комісія Комп'ютерної інженерії						
Н. Контр.												
Затверд.		Тащук О.Ю.										

Взаємодія компонент системи Smart паркінгу



					Система паркування автомобілів IoT із використанням esp8266	Літ.			Маса		Масштаб	
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Возняк О.М.										
Перевір.		Тащук О.Ю.										
Т. Контр.						Арк.			Аркушів			1
Реценз.						Циклова комісія Комп'ютерної інженерії						
Н. Контр.												
Затверд.		Тащук О.Ю.										

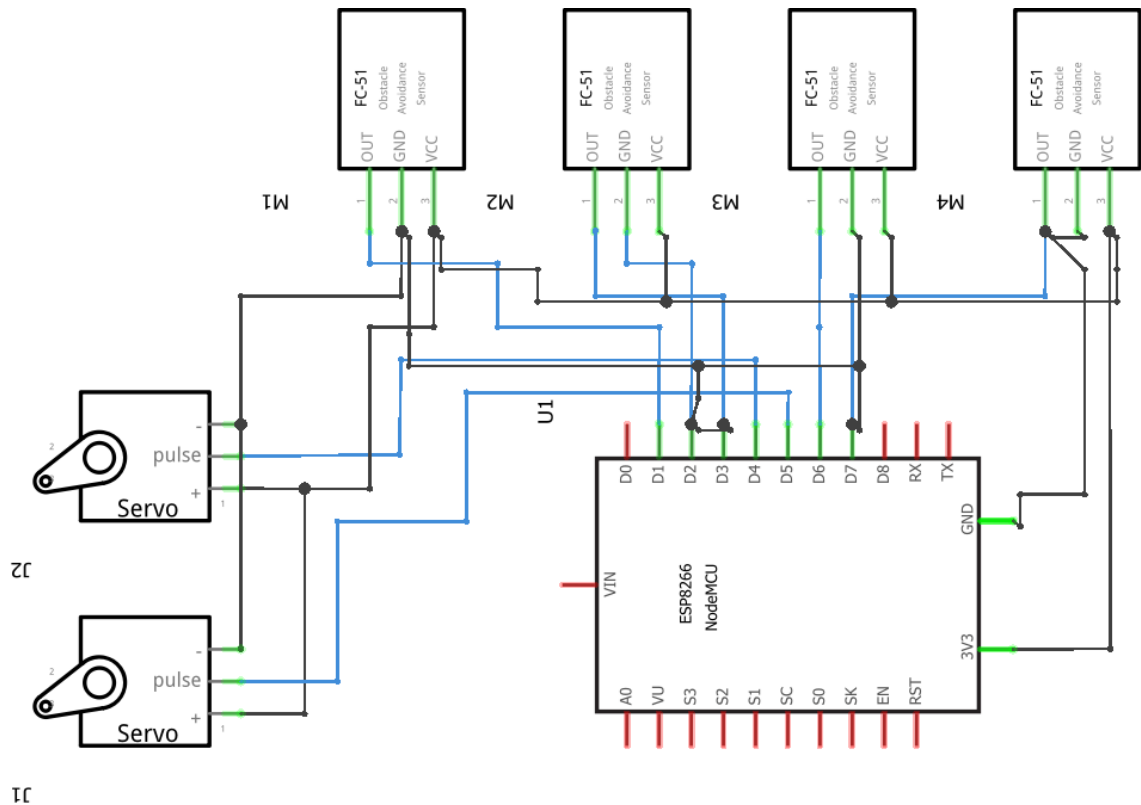
Макетна схема системи паркування



fritzing

					Система паркування автомобілів IoT із використанням esp8266	Лім.			Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.	Возняк О.М.									
Перевір.	Тащук О.Ю.									
Т. Контр.						Арк.	Аркушів 1			
Реценз.						Циклова комісія Комп'ютерної інженерії				
Н. Контр.										
Затверд.	Тащук О.Ю.									

Принципова схема системи паркування



fritzing

					Система паркування автомобілів IoT із використанням esp8266			
Змн.	Арк.	№ докум.	Підпис	Дата		Лім.	Маса	Масштаб
Розроб.		Возняк О.М.						
Перевір.		Тащук О.Ю.						
Т. Контр.						Арк.	Аркушів	1
Реценз.						Циклова комісія Комп'ютерної інженерії		
Н. Контр.								
Затверд.		Тащук О.Ю.						

Додаток Д

Блок-схема архітектури хмарного сховища Adafruit IO.

```
graph TD; A[Сенсори/Актори] --> B[Мікроконтролери  
(Arduino, ESP)]; B --> C[MQTT/HTTP  
Протоколи]; C --> D[MQTT Брокер]; D --> E[Хмарні сервіси  
(Сервери, Сховища)]; E --> F[Приладні Панелі  
(Dashboards)]; F --> G[Користувачі  
(Веб-інтерфейс, App)];
```

The diagram illustrates the architecture of the Adafruit IO cloud storage system. It shows a sequential flow of data from sensors to end-users. The process begins with sensors/actuators sending data to microcontrollers (Arduino, ESP), which then communicate via MQTT/HTTP protocols to an MQTT broker. The broker routes the data to cloud services (servers, storage), which are then accessed by dashboards, and finally by users (web interface, app).

						Система паркування автомобілів IoT із використанням esp8266	Лім.			Маса	Масштаб	
Змн.	Арк.	№ докум.	Підпис	Дата		Арк.			Аркушів 1			
Розроб.	Возняк О.М.											
Перевір.	Тащук О.Ю.											
Т. Контр.												
Реценз.						Циклова комісія Комп'ютерної інженерії						
Н. Контр.												
Затверд.	Тащук О.Ю.											