

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
Навчально-науковий інститут інформаційних технологій  
Кафедра інженерії програмного забезпечення автоматизованих систем

## **Пояснювальна записка**

до магістерської роботи  
на ступінь вищої освіти магістр  
на тему: **«РОЗРОБКА ПАРКУВАЛЬНОГО РАДАРУ ДЛЯ  
АВТОМОБІЛЬНОГО ТРАНСПОРТУ НА БАЗІ ARDUINO»**

Виконав: студент 6 курсу, групи ІСДМ–61  
спеціальності

126 Інформаційні системи та технології  
(шифр і назва спеціальності)

Кульчицький О.Є.

(прізвище та ініціали)

Керівник Тушич А.М.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

Київ – 2023

# НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти - «Магістр»

Спеціальність підготовки 126 Інформаційні системи та технології

Освітня програма «Інформаційні системи та технології»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ІСТ

К.П.Сторчак

“ \_\_\_\_\_ ” \_\_\_\_\_ 2023 року

## ЗАВДАННЯ НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Кульчицькому Олександрю Євгеновичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка паркувального радару для автомобільного транспорту на базі Arduino»

Керівник роботи: Тушич Аліна Миколаївна, д. ф., доц. каф. ПЗАС

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

затверджені наказом вищого навчального закладу від \_\_\_\_\_ року № \_\_\_\_\_

2. Строк подання студентом роботи \_\_\_\_\_

3. Вхідні дані до роботи :

1. Науково-технічна література

2. Аналіз предметної області

3. Дослідження щодо розробки проектів на базі Arduino

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

1. Аналіз предметної області

2. Вибір засобів для реалізації проекту

3. Опис процесу розробки проекту

5. Перелік графічного матеріалу

1. Титульний слайд

2. Мета і завдання

3. Актуальність дослідження

4. Вибір та обґрунтування технології

5. Порівняльна характеристика мов програмування

6. Створення проекту

7. Основні задачі системи

8. Демонстрація системи

6. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури		
2	Вивчення матеріалів для подальшої взаємодії з ними		
3	Вибір інструментів для розробки системи		
4	Дослідження методів розробки проєкту		
5	Визначення технічного завдання		
6	Проектування системи		
7	Вступ, висновки, реферат		
8	Розробка демонстраційних матеріалів		
9	Попередній захист роботи		

Студент \_\_\_\_\_ Кульчицький О.Є.  
(підпис) (прізвище та ініціали)

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





## РЕФЕРАТ

Текстова частина магістерської роботи 61 с., 1 табл., 16 рис., 4 дод., 34 джерела.

*Об'єкт дослідження* – процес розробки паркувального радару для автомобільного транспорту на базі Arduino.

*Предмет дослідження* – створення паркувального радару.

*Мета роботи* – дослідити методи створення автоматизованої системи паркування для автомобілів.

*Методи дослідження* – методи теорії інформації, методи наукового моделювання, методи дослідження інформаційних систем.

Проведено дослідження методів створення автоматизованої системи паркування для автомобілів, визначено їх цілі, типи та задачі, які вони вирішують.

На основі результатів виконаних досліджень було обрано оптимальні програмні інструменти та розроблено автоматизовану систему, яка надає можливість значно полегшити паркування автомобільного транспорту.

АВТОМАТИЗОВАНА СИСТЕМА, ІНФОРМАЦІЙНА СИСТЕМА,  
ARDUINO, МІКРОКОНТРОЛЕР, ІОТ, ПАРКУВАЛЬНА СИСТЕМА

## ЗМІСТ

<b>ВСТУП.....</b>	<b>8</b>
<b>1 МЕТОДОЛОГІЧНІ АСПЕКТИ РОБОТИ МЕРЕЖІ ІоТ.....</b>	<b>9</b>
1.1 Основні визначення в мережі ІоТ. Архітектура ІоТ.....	9
1.2 Принцип роботи ІоТ.....	12
1.3 Еталонна модель ІоТ.....	15
<b>2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....</b>	<b>19</b>
2.1 Паркувальний радар.....	19
2.2 Особливості використання паркувальних радарів.....	19
2.3 Аналіз існуючих рішень.....	22
<b>3 ВИБІР ТА ОБҐРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ .....</b>	<b>26</b>
3.1 Обґрунтування компонентів для пристрою.....	26
3.2 Вибір мікрокомп'ютера .....	27
3.3 Використання додаткових складових пристрою .....	30
<b>4 РОЗРОБКА СИСТЕМИ .....</b>	<b>38</b>
4.1 Структурна схема пристрою .....	38
4.2 Програмна реалізація парктроніка .....	41
4.3 Моделювання конструкції для пристрою.....	42
<b>ВИСНОВКИ .....</b>	<b>45</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ .....</b>	<b>46</b>
<b>Додаток А .....</b>	<b>50</b>
<b>Додаток Б .....</b>	<b>51</b>
<b>Додаток В .....</b>	<b>53</b>
<b>Додаток Г .....</b>	<b>55</b>
<b>ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація) .....</b>	<b>57</b>

## ВСТУП

У місцях, де відсутнє електропостачання, не обійтись без запрограмованої паркувальної системи, яка стане найкращим помічником для водіїв автомобілів, у яких не встановлено таку систему. «Сліпих зон» навколо автомобіля часто буває дуже багато та інколи людина не в змозі безпечно припаркувати автомобіль без підказки. Завдяки такому винаходу як паркувальний радар, ця проблема зникне раз і назавжди.

Це дослідження спрямоване на аналіз типів паркувальних систем, опис реалізації одного з таких типів системи та тестування на реальних даних. Дана система буде виявляти перешкоду на парковці, коли автомобіль заїжджає, щоб давати змогу паркувати водієві свій транспортний засіб.

Задачами дослідження є:

- провести аналіз відомих рішень;
- вибрати платформу та елементну базу;
- побудувати паркувальний радар на основі платформи та елементної бази;
- у тестовому режимі визначити роботоздатність даної розробки;
- розкрити суть побудови та програмування даного пристрою.

Паркувальна система створена з метою забезпечення безпечного паркування транспортного засобу, полегшення виконання рухів заднім ходом автомобіля.



# 1 МЕТОДОЛОГІЧНІ АСПЕКТИ РОБОТИ МЕРЕЖІ ІoT

## 1.1 Основні визначення в мережі ІoT. Архітектура ІoT

Мережа наступного покоління (NGN) – це мережа з пакетною комутацією, придатна до надання послуг електров'язку та для використання декількох широкосмугових технологій транспортування з увімкненою функцією якості обслуговування (QoS), в якій зв'язані з обслуговуванням функції не залежать від використаних технологій, що забезпечують транспортування.

Пристрій – обладнання, яке володіє обов'язковими можливостями зв'язку та додатковими можливостями такими, як: вимірювання, спрацювання, а також введення, зберігання та обробку даних.

Інтернет речей (ІoT) – це глобальна інфраструктура для інформаційного суспільства, котра забезпечує можливість надання більш важких послуг шляхом з'єднання один з одним (фізичних та віртуальних) речей на основі наявних функціонально сумісних інформаційно-комунікаційних технологій.

Архітектура ІoT пристроїв складається з чотирьох рівнів: рівень датчиків, мережевий рівень, рівень обробки даних та прикладний рівень (рис. 1.1):

Рівень датчиків. Головна мета рівня датчиків – ідентифікація різних явищ навколишнього середовища за допомогою периферійних пристроїв та отримання даних з реального світу. Цей рівень складається з декількох видів датчиків. Використання великої кількості датчиків є однією з найважливіших функцій ІoT пристроїв. Датчики в ІoT пристроях зазвичай інтегрується за допомогою концентратора датчиків. Концентратор датчиків є головною точкою з'єднання для кількох датчиків. Він акумулює та передає дані з датчиків до блоку обробки даних в пристрої. Концентратор датчиків використовує декілька транспортних механізмів таких, як (Inter-Integrated Circuit (I2C) чи Serial Peripheral Interface

(SPI)) для передачі даних між датчиками та додатками. Ці транспортні механізми залежать від IoT пристроїв. Вони створюють канал зв'язку між датчиками та додатками збору даних з датчиків.

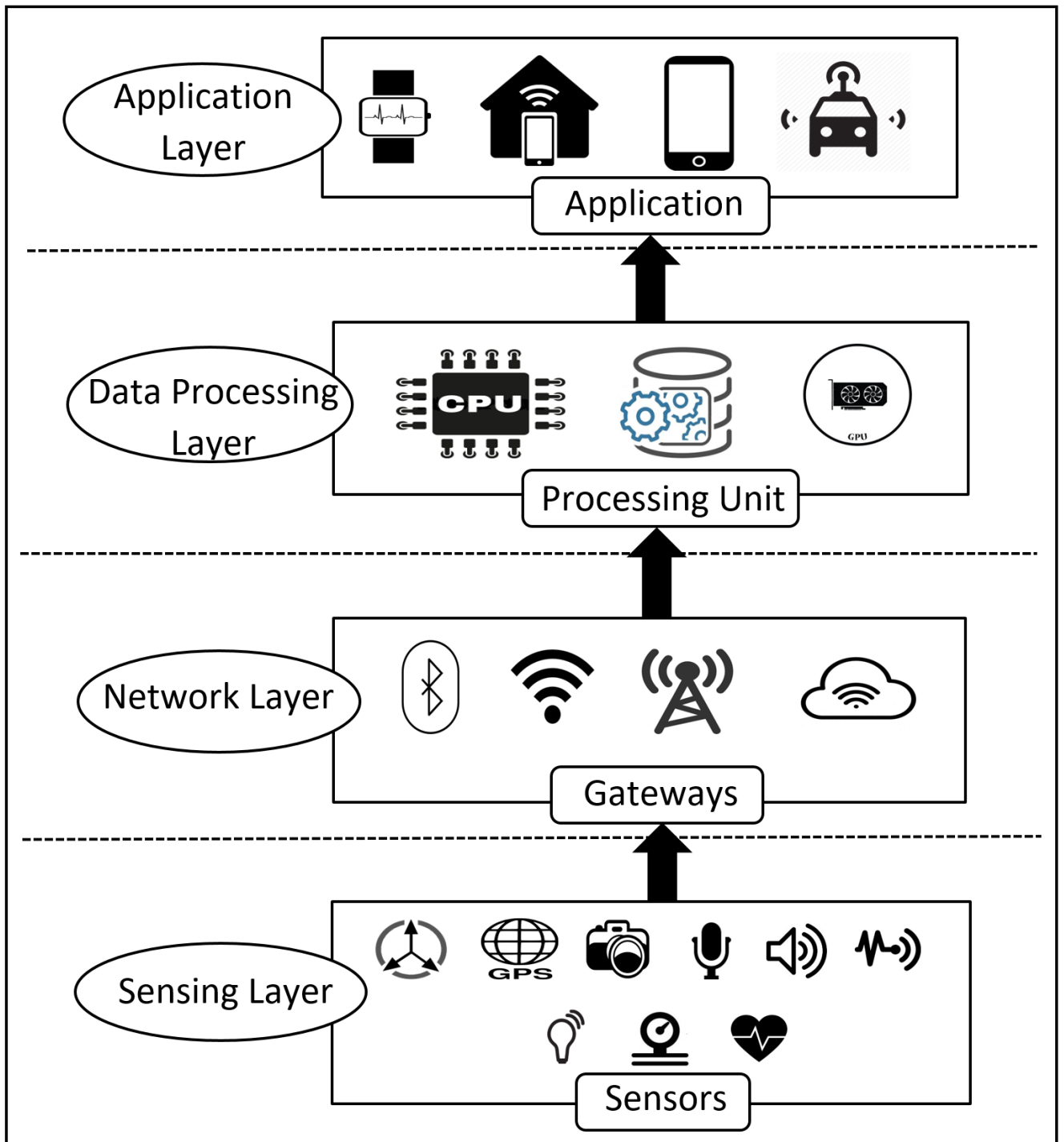


Рис. 1.1 - Архітектура мережі IoT

Датчики в IoT пристроях можуть бути класифіковані на 3 широкі категорії:

- Датчики руху, які вимірюють зміни в русі, а також орієнтацію пристроїв. Є два типи рухів, які можна спостерігати за допомогою датчику цього пристрою: лінійний та кутовий рух. Лінійний рух відноситься до лінійного переміщення пристрою IoT, в той час як кутовий рух відноситься до обертального переміщення пристрою.

- Датчики навколишнього середовища до них відносяться такі пристрої, як датчики світла, датчики тиску та інші, які вбудовані в IoT пристрої реагують на зміни в параметрах навколишнього середовища за допомогою периферійних пристроїв. Основною метою використання цих датчиків в IoT пристроях є допомога пристроям приймати автономні рішення відповідно до змін в периферійних пристроях. Наприклад, датчики навколишнього середовища використовуються в більшості додатків для спрощення життя користувачів (розумні замки, система домашньої автоматизації, розумне освітлення та ін.).

- Датчики місцеперебування пристроїв IoT взаємодіють з фізичним місцеперебуванням та розташуванням самого пристрою.

Найбільш поширеними датчиками місцеперебування, що використовуються в IoT є магнітні датчики та Global Position System (GPS) датчики. Магнітні датчики використовуються, як цифрові компаси та допомагають фіксувати орієнтацію дисплею пристрою. GPS датчики використовується для навігаційних цілей в IoT пристроях.

Мережевий рівень. Мережевий рівень використовується, як комунікаційний канал для передачі даних, зібраних на рівні датчиків, до інших підключених пристроїв.

В IoT пристроях, мережевий рівень реалізований для використання різноманітних комунікаційних технологій (наприклад, Z-Wave, LoRa, Wi-Fi, Bluetooth та ін.), що дозволяють передавати дані між різними пристроями всередині самої мережі.

Рівень обробки даних. Рівень обробки даних складається з головного блоку обробки даних IoT пристрою. Рівень обробки даних отримує зібрані дані на рівні датчиків та аналізує їх. Потім приймає рішення, що базуються на результаті аналізу. В деяких IoT пристроях (наприклад, смарт-годинник, розумний домашній концентратор та ін.), цей рівень також зберігає результати попередніх аналізованих даних для подальшої можливості використання цих даних.

Також цей рівень може передавати результати обробки даних з одного підключеного пристрою на іншій пристрій через мережевий рівень.

Рівень додатків. Рівень додатків презентує результати роботи рівню обробки даних за допомогою різного роду додатків IoT пристроїв. Рівень додатків є рівнем орієнтованим на користувачів, який виконує різні завдання для користувача.

Існують різні IoT додатки, які включають розумний будинок, розумний транспорт тощо.

## **1.2 Принцип роботи IoT**

На рис. 1.2 зображено принцип роботи найпростішої мережі IoT.

Принцип роботи IoT об'єднує чотири різні етапи:

- Зчитування інформації за допомогою датчиків. По-перше, датчики або пристрої збирають дані зі свого оточення. Це може бути як простим процесом, таким як зчитування температури, або складним, як запис відео на камеру відеоспостереження. Використовується на даному етапі також й пристрої, оскільки декілька датчиків можуть бути об'єднані разом або датчики можуть бути частиною пристрою, що робить більше, ніж просто аналіз даних. Наприклад, ваш телефон - це пристрій з кількома датчиками (камера, акселерометр, GPS тощо), але телефон не є лише датчиком. На цьому кроці дані збираються з нашого оточення.

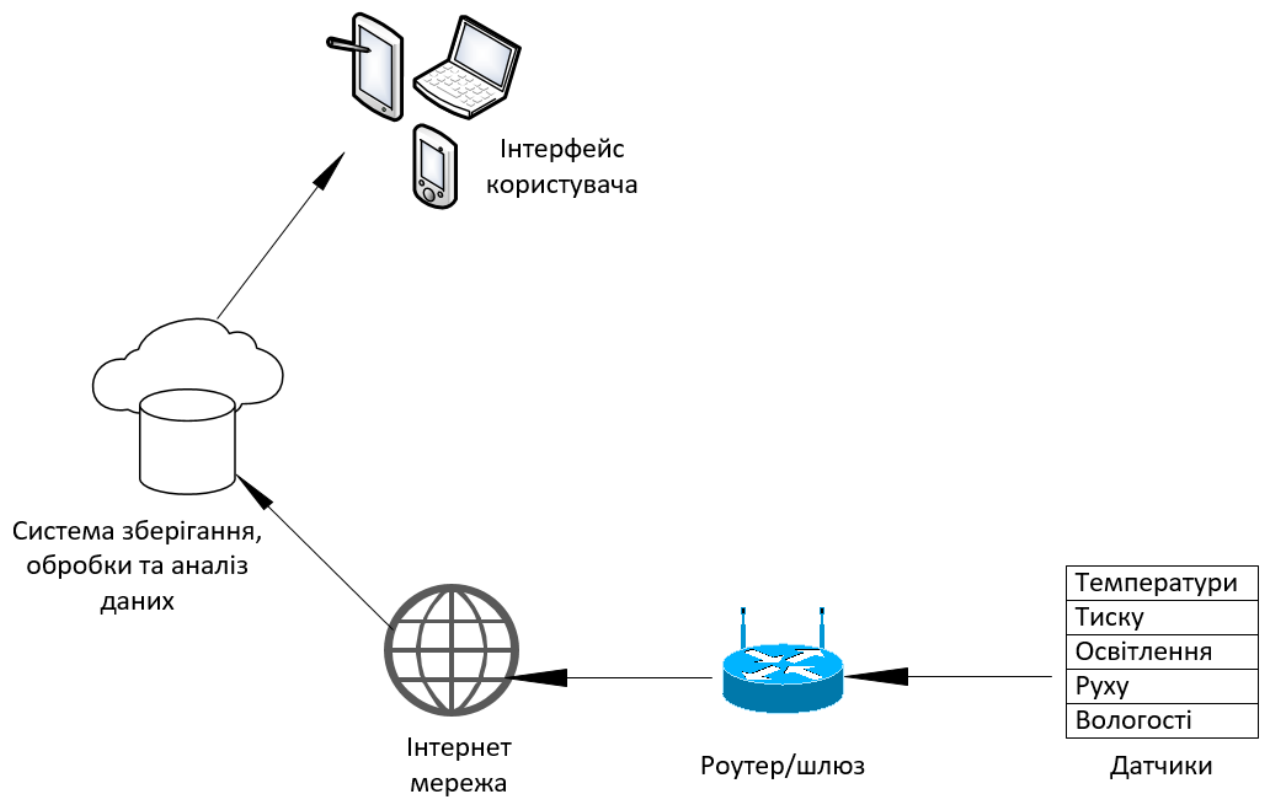


Рис. 1.2 Принцип роботи IoT

- Передача даних від датчиків до хмарних сховищ. Датчики можуть бути підключені до хмари за допомогою різних методів, включаючи: стільникову мережу, супутникову мережу, Wi-Fi, Bluetooth, малопотужні широкопasmові мережі (LPWAN), або з'єднатися безпосередньо до Інтернету через Ethernet технологію. Кожна опція має компроміс між споживанням енергії, діапазоном та пропускнуою здатністю. Вибір того чи іншого варіанту підключення зводиться до вибору конкретної області, де буде використовуватися Інтернет речей, але всі вони виконують одне і те ж завдання - передачу даних до хмари.

- Обробка даних отриманих за допомогою датчиків. Після того, як дані потраплять до хмари, програмне забезпечення виконує певну обробку даних. Це може бути, наприклад, перевірка того, що зчитувана температура знаходиться в

межах допустимого діапазону. Чи це може бути процесом обробки інформації, наприклад, використання комп'ютерного програмного забезпечення, яке здійснює ідентифікацію об'єктів, що знаходяться на відео (наприклад, зловмисників у вашому домі).

- Передача інформації на інтерфейс користувача. На даному етапі відбувається передача корисної інформації кінцевому користувачу певним чином. Це може бути через сповіщення користувача (електронна пошта, текст, сповіщення тощо). Наприклад, текстове сповіщення, коли температура занадто висока в холодному сховищі компанії. Крім того, користувач може мати інтерфейс, який дозволяє йому активно перевіряти систему. Наприклад, користувач може захотіти перевірити камери відеоспостереження в їхньому будинку через телефонну програму, або веб-сторінку. Однак це не єдиний шлях. Залежно від програми IoT, користувач може також виконувати дії на відстані та впливати на систему. Наприклад, користувач може дистанційно регулювати температуру в холодному сховищі за допомогою програми на своєму телефоні. А деякі дії виконуються автоматично. Замість того, щоб чекати, коли користувач налаштує температуру, система може зробити це автоматично за допомогою попередньо визначених правил. І замість того, щоб просто викликати вас, щоб попередити вас про зловмисника, система IoT також може автоматично повідомляти відповідні органи. Система IoT складається з датчиків, які розмовляють з хмарою через певний тип зв'язку. Після того, як дані потрапляють у хмару, програмне забезпечення обробляє його, а потім може вирішити виконати певну дію, наприклад, надіслати попередження, або автоматично регулювати роботу датчика без потреби користувача.

### 1.3 Еталонна модель IoT

На рис. 1.3 показана еталонна модель IoT, яка наведена в рекомендації Y.2060, вона складається з чотирьох основних рівнів, а також з двох додаткових рівнів: можливості управління та безпеки.

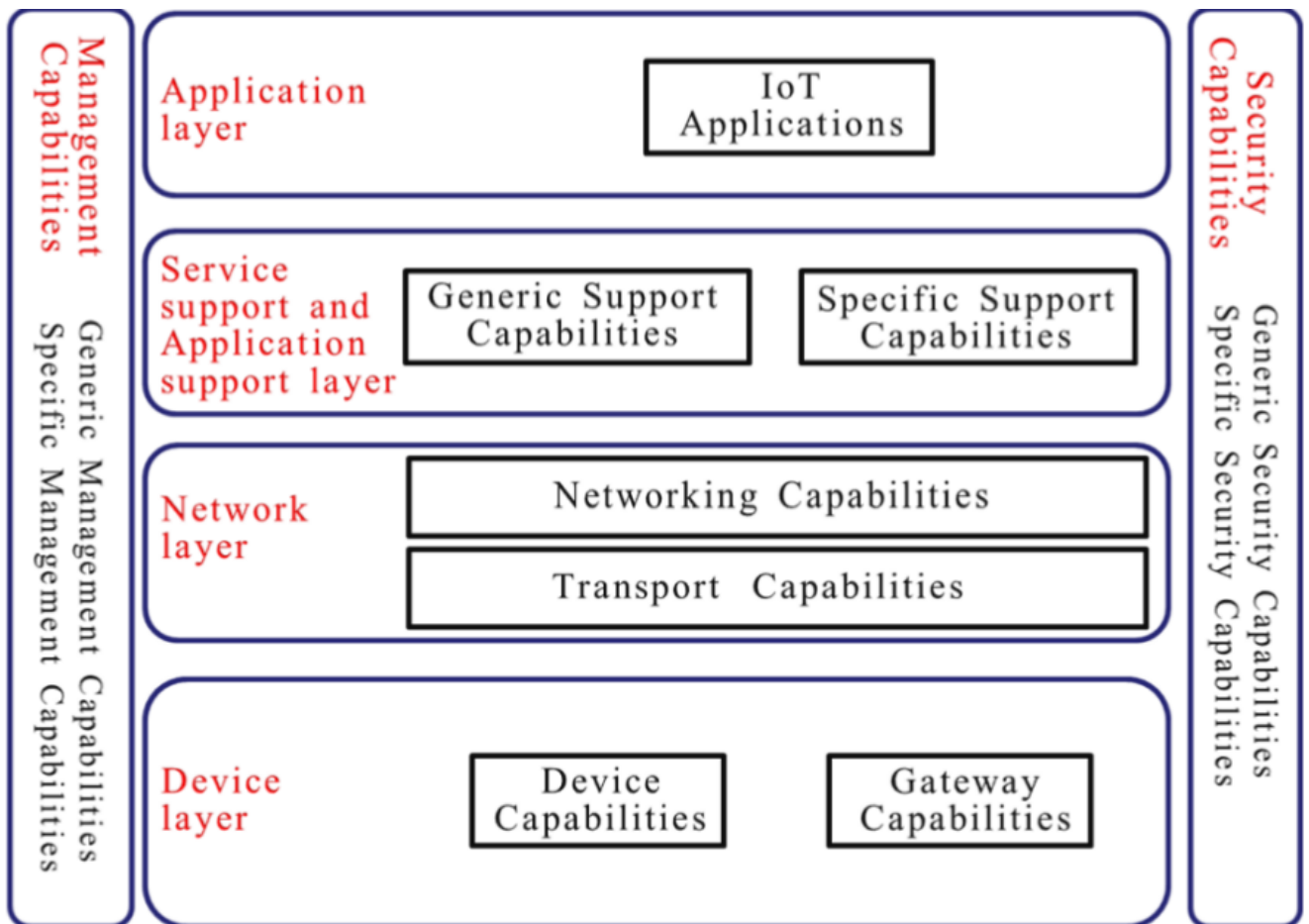


Рис. 1.3 Еталонна модель IoT

До основних рівнів належить:

- Прикладний рівень. Прикладний рівень в Рекомендації Y.2060 не розглядається.
- Рівень підтримки послуг та додатків. Рівень підтримки послуг та додатків складається з наступних двох груп можливостей, таких як загальні можливості

підтримки та спеціалізовані можливості підтримки:

- Загальні можливості підтримки - це можливості, які можуть використовуватися різними додатками такими як обробка та зберігання даних. Ці можливості можуть бути активовані за допомогою спеціалізованих можливостей підтримки, наприклад, для створення інших спеціалізованих можливостей підтримки;

- Спеціалізовані можливості підтримки - це конкретні можливості, які призначені для задоволення вимог різноманітних додатків. Насправді вони можуть складатися з ряду груп чітко визначених можливостей, для того щоб надавати різні функції підтримки різних програм IoT.

- Мережевий рівень. Цей рівень складається з наступних двох типів можливостей:

1) Можливості організації мереж: надає відповідні функції управління мережевим з'єднанням, такі як функції управління доступом та ресурсами транспортування, управління мобільністю або автентифікація, авторизація та облік (AAA).

2) Можливості транспортування: призначені для надання з'єднання для транспортування інформації у вигляді даних, що відносяться до послуг і додатків IoT, а також транспортування інформації контролю та управління, що відносяться до IoT.

- Рівень пристроїв. Можливості рівня пристроїв можна логічно розділити на два види можливостей:

1) Можливості пристрою. Можливості пристрою діляться в свою чергу на:

а) Пряму взаємодію з мережею зв'язку: пристрої здатні збирати та закачувати інформацію безпосередньо (тобто без використання можливостей шлюзу) в мережі зв'язку та можуть безпосередньо отримувати інформацію (наприклад, команди) із мережі зв'язку.

б) Непряму взаємодію мережі зв'язку: пристрої здатні отримувати та закачувати інформацію в мережі зв'язку непрямим чином, тобто за допомогою



можливості шлюзу. На іншій стороні пристрою можуть не прямим чином отримувати інформацію (наприклад, команди) з мережі зв'язку.

Організація спеціальних мереж в ряді сценаріїв вимагає підвищену масштабованість та швидкого розгортання, пристрої можуть мати можливість будувати мережі довільним чином

в) Сплячий режим та пробудження: можливості пристроїв можуть підтримувати механізми "сну" і "пробудження" для економії енергії.

2) Можливості шлюзу. Можливості шлюзу включають, в тому числі:

а) Підтримку декількох інтерфейсів: на рівні пристроїв можливості шлюзу підтримують пристрої, сполучені з використанням різних дротових і бездротових технологій, таких як шина локальних мереж контролерів (CAN), Bluetooth, або Wi-Fi. На рівні мережевих можливостей шлюзу можуть обмінюватися даними з використанням різних технологій, таких як комутована телефонна мережа загального користування, мережі другого або третього покоління (2G або 3G), мережі на базі технології довгострокового розвитку (LTE), Ethernet або цифрові абонентські лінії (DSL). б) Перетворення протоколу: існує дві ситуації в яких потрібні можливості шлюзу. Перша ситуація виникає тоді, коли для зв'язку на рівні пристроїв використовуються різні протоколи рівня пристроїв, наприклад, протоколу технологій Z-Wave, Bluetooth; друга ситуація виникає тоді, коли для зв'язку потребується і рівень пристроїв і мережевий рівень, використовуються різні технології, наприклад, протокол технологій Z-Wave на рівні пристроїв і протокол технології 3G на мережевому рівні.

- Можливості управління. За аналогією з традиційними засобами зв'язку, можливості управління IoT охоплює традиційні класи несправності, конфігурації обліку, показників роботи та безпеки (FCAPS), тобто управління несправностями, управління конфігурації, управління обліком, управління показниками роботи та

управління безпекою. Можливості управління IoT можуть бути поділені на категорії загальних можливостей управління і спеціалізованих можливостей управління.

Найважливіші загальні можливості управління в IoT включає:

а) управління пристроями, наприклад дистанційне ввімкнення або вимкнення пристроїв, діагностика, оновлення прошивки та/або програмного забезпечення, управління робочим станом пристроїв;

б) управління топологією локальної мережі;

в) управління трафіком і перевантаженнями, наприклад виявлення умов перевантаженості мережі та реалізації резервування ресурсів для термінових та/або життєво важливих потоків трафіку.

Спеціалізовані можливості управління тісно пов'язана з вимогою додатків, наприклад вимогами по контролю лінії передачі електроенергії в розумній електромережі.

- Можливості забезпечення безпеки. Існує два види можливостей забезпечення безпеки: загальні можливості забезпечення безпеки та спеціалізовані можливості забезпечення безпеки.

1) Загальні можливості забезпечення безпеки не залежить від додатків і включає:

а) на прикладному рівні: авторизацію, автентифікацію, захист конфіденційності та цілісності даних додатків, захист недоторканності приватного життя, аудит безпеки та антивірусну програму;

б) на мережевому рівні: авторизація, автентифікація, конфіденційність даних про використання та даних сигналізації, а також захист цілісності даних сигналізації;

в) на рівні пристроїв: автентифікація, авторизація, перевірку цілісності пристрою, управління доступом, захист конфіденційності і цілісності даних.

Спеціалізовані можливості забезпечення безпеки тісно пов'язані з вимогами додатків, наприклад вимогам безпеки мобільних платежів.

## **2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

### **2.1 Паркувальний радар**

Паркувальний радар, також відомий як акустична система паркування (APS), парктронік або ультразвуковий паркувальний датчик, є додатковою системою паркування, встановленою на деяких автомобілях.

Система використовує ультразвукові датчики, встановлені в передньому та задньому бамперах, для вимірювання відстані до найближчих об'єктів. Система видає переривчастий попереджувальний сигнал (а в деяких версіях відображає інформацію про відстань на РК-дисплеї дзеркала заднього виду, інтегрованого в панель приладів), щоб вказати, наскільки далеко автомобіль знаходиться від перешкоди.

Частота попереджувального сигналу збільшується в міру зменшення відстані від перешкоди. Перші звуки він видає при наближенні до перешкоди на відстань 1-2 метри, а при небезпечному наближенні до перешкоди (10-40 см в залежності від моделі) звуковий сигнал стає безперервним. На деяких моделях систему можна відключити, наприклад, під час руху по бездоріжжю. Як правило, система автоматично вмикає передачу заднього ходу (наприклад, живлення може подаватись від ланцюга ліхтаря заднього ходу).

### **2.2 Особливості використання паркувальних радарів**

Існує багато типів паркувальних систем, які відрізняються в першу чергу кількістю і розташуванням ультразвукових передавальних датчиків. Найпростіші

системи використовують два датчика, встановлені на задньому бампері автомобіля. Система активується, коли водій включає задню передачу. Найпопулярніші подібні системи використовують 4 датчика, розташовані на задньому бампері на відстані 30-40 см один від одного. Таке розташування датчиків дозволяє виключити виникнення «мертвих зон». У більш складних системах на передньому бампері встановлюють 2 або 4 датчика. Система попереджає вас про наближення до перешкоди, коли ви натискаєте педаль гальмів. Ексклюзивні системи можуть використовувати більше датчиків, а також датчики, розташовані з боків автомобіля.

Як правило, дисплей і блок управління з'єднуються кабелем, прокладеним уздовж кузова автомобіля, але існують і бездротові системи, які відрізняються від інших простотою установки. Принцип роботи такої системи полягає в бездротовій передачі радіосигналу від пульта до дисплея.

Система включає:

- електронний замок;
- випромінювачі ультразвукового датчика;
- пристрої відображення (РК-дисплей);
- звукові сповіщення (зумер).

Система працює за принципом сонара. Датчик передавача генерує ультразвуковий імпульс (приблизно 40 кГц), а потім приймає сигнал, відбитий від навколишніх об'єктів. Електронний блок вимірює час між надсиланням і отриманням відбитого сигналу та використовує швидкість звуку в повітрі як константу для розрахунку відстані до об'єкта. Таким чином по черзі перевіряються кілька датчиків і на основі отриманої інформації виводиться інформація на дисплей і, при необхідності, подаються попереджувальні сигнали за допомогою пристрою звукового сповіщення.

Його можна встановити як додаткову опцію практично в будь-який автомобіль, який не має стандартного паркувального радара. Автолюбителі, які володіють деякими знаннями в ремонті та обслуговуванні автомобілів, також

можуть самостійно встановити подібну систему на свій автомобіль, придбавши в магазині монтажний комплект.

Хоча система розроблена для підтримки водія, на неї не можна повністю покладатися. Незалежно від наявності системи, водій повинен візуально перевірити наявність перешкод перед тим, як рухатися в будь-якому напрямку. Деякі об'єкти не можуть бути виявлені паркувальним радаром через фізичні принципи роботи, інші можуть призвести до неправильної активації системи.

Паркувальний радар може генерувати помилкові сигнали в таких випадках:

- наявність льоду, снігу або іншого сміття на датчику;
- перебування на нерівній поверхні, ґрунтовій, похилій дорозі;
- міжміський рух;
- наявність джерел з підвищеним шумом в зоні дії датчика;
- робота під сильним дощем або снігопадом;
- робота радіопередавачів в зоні дії датчика;
- буксирування причіпу;
- паркування в обмежених умовах (ефект луни).

Система може не реагувати на наступне:

- гострі або тонкі предмети, наприклад, ланцюги, троси, тонкі стовпи;
- предмети, що поглинають ультразвук (одяг, пористі матеріали, сніг);
- предмети висотою менше 1 метра;
- об'єкти, що відбивають звук від датчиків;
- система не може виявити ями в асфальті, відкриті люки, розкидані дрібні гострі предмети та інші небезпечні предмети, які знаходяться поза полем зору датчиків.

## 2.3 Аналіз існуючих рішень

Парктронік, тобто система паркування - створений для полегшення виконання паркування водієм, задля безпечного паркування транспортного засобу та уникнення аварій при русі заднім ходом.

На ринку багато відомих рішень. Розглянемо деякі з них.

Паркувальна система Falcon 2616-4 (рис. 2.1) чорного кольору найдешевша система паркування на ринку. Її особливості включають:

- активація датчиків від 0,3 м до 1,5 м в залежності від кута виявлення перешкод (по вертикалі -  $60^\circ$  і по горизонталі -  $120^\circ$ ),
- робоча напруга - 9-16 В,
- працює на струмі від 20 до 200 мА.

Система Falcon 858K Black (рис. 2.2) відноситься до середнього цінового діапазону на ринку паркувальних систем. Її особливості:

- активація датчика від 0,3 м до 1,5 м, вертикальний кут виявлення перешкод  $60^\circ$  і горизонтальний -  $120^\circ$ ,
- робоча напруга - 9-16 В,
- робочий струм від 20 до 200 мА.

Отже, однакові робоча напруга і струм, робочий кут і відстань. Датчики як в дешевшому варіанті.

Також доступна система паркування Tiger TG-P8 Black (рис. 2.3), що має середній ціновий діапазон на ринку паркувальних систем. В її особливості включають:

- датчики спрацьовують від 0,2 м до 2,5 м, кут визначення перешкод по вертикалі -  $60^\circ$  і по горизонталі -  $120^\circ$ ,
- робоча напруга - 12 В,
- робочий струм  $\sim 200$  мА.

Пристрій дозволяє краще активувати датчики, від 20 до 250 см, що є

ефективнішим для уникнення дорожньо-транспортних пригод. Робоча напруга відносно стабільна в порівнянні з двома попередніми.

Розглянемо порівняльну характеристику відомих рішень (табл. 2.1).

Таблиця 2.1 – Порівняльна характеристика популярних рішень

Характеристики	Falcon 2616-4 black	Falcon 858K black	Tiger TG-P8 black
Низька вартість	+	-	-
Низький робочий струм	-	-	-
Низька напруга роботи	-	-	-
Низька мінімальна дальність спрацювання (<20 см)	-	+	+
Велика максимальна дальність спрацювання (>200 см)	-	-	+

Аналізуючи існуючі рішення паркувальних систем, можна побачити, що не всі вони ідеальні, адже кожна має свої переваги та недоліки. Тому «розробка парктроніків на базі плати Arduino» є актуальним завданням.

Отже, для досягнення поставленої у роботі мети необхідно вирішити такі завдання:

- проаналізувати відомі рішення;
- вибрати платформу та основні елементи;
- побудувати систему паркування на основі платформи та елементарної бази;







Рис. 2.2 - Falcon 858K black



Рис. 2.3 - Tiger TG-P8 black

## **3 ВИБІР ТА ОБҐРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ**

### **3.1 Обґрунтування компонентів для пристрою**

Для того, щоб почати розробку системи, необхідно розглянути існуючі компоненти, можливості для реалізації та вибрати підходящі відповідно до вимог. Для даної системи необхідно обрати елементи та технології для реалізації пристрою.

У складі майбутнього пристрою необхідним є використання датчика HC-SR04, за допомогою якого визначається відстань до перешкоди. Таким чином, за допомогою датчика HC-SR04 і плати Arduino можна скласти найпримітивніший радар.

Для відображення статусних повідомлень пристрою використовується РК-дисплей 16x2. Для приводу на дію механізму використовується сервомотор. Кут повороту радара можна регулювати за допомогою кута повороту сервомотора.

Необхідні компоненти:

- Arduino Uno, Nano, Mega;
- ультразвуковий далекомір HC-SR04;
- дисплей LCD 1602 I2C;
- світлодіоди та резистори;
- RGB світлодіоди Arduino;
- п'єзодинамік;
- з'єднувальні дроти;
- макетна плата.

## 3.2 Вибір мікрокомп'ютера

На ринку представлено багато різних моделей мікрокомп'ютерів. До них відносяться BeagleBoard, Raspberry Pi, Orange Pi і Stm32 Discovery, плати з сімейства Arduino.

Плата Beagle — це одноплатний комп'ютер, розроблений Texas Instruments та Digi-Key. Плата Beagle спочатку була розроблена в тісній співпраці з спільнотою відкритого коду для демонстрації можливостей системи на мікропроцесорі OMAP3530 на Linux ОС. Але зараз підтримка цього пристрою не дуже велика, тому розробка займає більше часу.

Raspberry Pi — одноплатний комп'ютер, що розроблений британським фондом Raspberry Pi. Хоча його початкова мета полягала в тому, щоб стимулювати вивчення основ інформатики в школах, у міру зростання популярності та можливостей платформи її почали використовувати в реальних проектах. Ви можете встановити один з кількох дистрибутивів Linux на Raspberry. Зазвичай використовується RaspbianOS.

Orange pi — одноплатний комп'ютер, має відкритий вихідний код. Він може працювати з Android 4.4, Debian, Ubuntu, Raspbian Image. Хоча творці стверджують, що мають великі можливості та підтримку спільноти, виявляється, що багато простих речей їм доводиться робити самим. Зокрема, є проблеми з поганою роботою драйверів виробника. Багато розробників також скаржаться на відсутність повної документації, що змушує їх використовувати методи зворотного інжинірингу, щоб дізнатися, як працює та чи інша функціональність.

Arduino - це апаратна обчислювальна платформа для аматорського проектування, основними елементами якої є плата мікроконтролера з елементами вводу-виводу і середовище розробки, написане на мові програмування, що є спрощеною підмножиною C/C++. Arduino можна використовувати як для створення незалежних інтерактивних об'єктів, так і для співпраці з програмним

забезпеченням, запущеним на комп'ютері (наприклад, Processing, Adobe Flash, Max / MSP, Pure Data, SuperCollider). Інформація про масив (компонування масиву, специфікації компонентів, програмне забезпечення) є загальнодоступною і може бути використана тими, хто віддає перевагу створенню власного масиву.

Stm32 — плата з мікроконтролером на основі ядра ARM Cortex-M3. Це ядро має багато переваг, які перераховані нижче, але головною його перевагою сьогодні є його універсальність. Протягом двох років Cortex-M3 став галузевим стандартом. Це видно з того, скільки виробників приєдналися до цієї архітектури. Усі основні виробники мікроконтролерів мають або розробляють рішення на основі цієї архітектури: STMicroelectronics, Texas Instrument, NXP, ATMEL, Analog Devices, Renesas тощо. Проте ST була однією з перших, хто випустив на ринок свої мікроконтролери Cortex-M3 і швидко став провідним гравцем на ринку. Це свідчить про високу якість і привабливість рішень STMicroelectronics. Чітке і значне збільшення продажів самих ядер Cortex-M3, однак, є деякі обмеження цієї платформи: вона повільна в порівнянні з іншими, важко налаштовується.

Отже, виходячи з того, що підтримка, актуалізація та оновлення програмного забезпечення для плат з мікрокомп'ютерами Beagle, Orange pi та Stm32 наразі не є достатньою, про що свідчать відгуки розробників та дати останніх оновлень, тому вибір мікрокомп'ютера зводиться до вибору між Raspberry Pi та Arduino.

Переваги використання Arduino:

- архітектура має низьке енергоспоживанням (у порівнянні з Raspberry Pi);
- з ним легко працювати, має відмінну онлайн-підтримку, швидкий, простий;
- створення прототипів;
- просте сполучення з датчиками і збір даних;
- дешевше, ніж Raspberry Pi (для продуктів, які не мають можливості підключення до Інтернету);
- можна відправляти дані без проводів, використовуючи Bluetooth, Rf на сервер через комп'ютер;

- багато GPIO з можливостями PWM, що є дружнім до розробників;
- повністю відкритий вихідний код.

#### Недоліки Arduino:

- обмеження пам'яті (в порівнянні з Raspberry Pi);
- для підключення до Інтернету потрібні додаткові підключені пристрої;
- менш потужний у порівнянні з Raspberry Pi;
- не можна запустити багато важких алгоритмів, або програмувати сенсорний екран і використовувати додаткові плати розширення;
- широка онлайн-спільнота і нескінченні можливості того, що можна зробити з її допомогою.

#### Переваги Raspberry Pi:

- високий показник продуктивності, достатня кількість пам'яті;
- ОС, заснована на Linux, а тепер навіть і Windows 10, яка може запускатися поверх неї, щоб зробити обробку більш зручною для користувача;
- велика кількість доступних GPIO чим більше GPIO, тим більше датчиків можна з'єднати;
- дуже легко почати розробку користувачам, у яких є досвід з Linux системами;
- мови Java, Python, C, C ++, Ruby, Go можуть бути використані для програмування саме так, як на комп'ютері;
- широка онлайн-спільнота і нескінченні можливості того, що можна зробити з її допомогою.

#### Недоліки Raspberry Pi:

- потрібно добре знання Linux-систем;
- для більшості застосунків продуктивність буде надмірною, оскільки використовуватиметься в основному тільки передача даних;
- закритий вихідний код;
- за рахунок своєї потужності є більш енерговитратним.

Таким чином, можна зробити висновок, що обидві платформи підходять для вибору їх за апаратну основу. Але для автоматизації з вирішенням поставлених завдань ефективніше з точки зору собівартості пристрою буде використовувати Arduino.

### **3.3 Використання додаткових складових пристрою**

Парктронік для гаража на Arduino зробити досить просто, при цьому можна використовувати світлову та звукову індикацію для попередження про мінімальну відстань. Використовувати цей проект на ультразвукових датчиках в автомобілі не рекомендується, оскільки вони мають велику похибку і не захищені від вологи. Розглянемо кілька варіантів реалізації проекту на Arduino.

Для складання парктроніка (додаток Б і В) слід підключити до плати ультразвуковий далекомір, п'єзодинамік і LCD екран, RGB-світлодіоди.

Ультразвуковий датчик HC-SR04 - це є стабільним та точним ultrasonic sonar (сонар) датчиком відстані, що не містить "сліпих зон". Може вимірювати відстань до перешкод від 0 см до 1500мм, точність досягає 3 мм.

Характеристики:

- Робоча напруга: 3.8 - 5.5В
- Тип: HC-SR04
- Струм: 8 мА
- Частота: 40 кГц
- Максимальна дистанція: 1500 мм
- Мінімальна дистанція: 0 см
- Роздільна здатність: 3 мм
- Ширина імпульсів: 10 мкс
- Кут: 15 градусів

- Зовнішні габарити: 37x20x15 мм

#### Принцип роботи

1. На вихід trig (тригер) посилається високий рівень протягом мінімум 10 мкс.

2. Модуль посилає ультразвукові імпульси з частотою 40 кГц та приймає їх назад у випадку, якщо в зоні видимості є будь-які перешкоди.

3. Якщо сигнал повернувся, то модуль встановлює низький рівень на виході echo на 150 мс. За період часу, який минув з п.1 до низького рівня на виході echo розраховується відстань до перешкоди за наступною формулою:

$$\text{Відстань} = (\text{time} * \text{sound velocity}) / 2$$

де time - це вимірний час імпульсу, sound velocity - це швидкість звуку (340 м/с)



Рисунок 3.1 - Ультразвуковий датчик HC-SR04 для Arduino

П'єзоелектричний випромінювач (п'єзодинамік) – це датчик, що є електроакустичним приладом, здатним відтворювати звукові хвилі завдяки зворотному п'єзоелектричному ефекту.

П'єзодинамік складається з шару п'єзоелектрика (або декількох шарів), який нанесений на металеву пластину (мембрану) товщиною до 1.5 мм. П'єзоелектрик виготовляють з діелектричних матеріалів, що мають властивість п'єзоефекту. Один з таких матеріалів – цирконат-тітнату свинцю. Зовнішня сторона п'єзоелектрика покривається струмопровідним напиленням.

Металева пластина і напилення є контактними виводами п'єзодинаміка. До них підводять живлення з використанням проводів або контактних груп.

Для посилення звуковипромінюючих властивостей п'єзодинамік можуть поміщати в корпус, який служить свого роду рупором. Корпус для п'єзодинаміка може бути як пластиковим, так і керамічним або металевим. Конфігурація приладу в основному є однотипною: заглиблення в корпусі, що створює додатковий резонанс та наявний отвір в центрі призначений для виходу звукових хвиль.

Деякі моделі приладу доповнюють дифузором чи діафрагмою, що безпосередньо пов'язані з мембраною та повторюють її коливання. Можуть мати додаткові елементи для кріплення або вбудовані чіпи.

Прямий п'єзоелектричний ефект було відкрито в 1880 році братами П'єром і Жаком Кюрі. Полягає цей ефект в поляризації п'єзоелектрика за наявності механічного впливу на нього та деформації. Уже через рік був доведений також і зворотний ефект.

Зворотний п'єзоелектричний ефект полягає у його механічній деформації п'єзоелектрика під впливом електричного поля.

Під час подачі напруги на п'єзодинамік, шар п'єзоелектрика зазнає деформації, при цьому то зменшуючи, то збільшуючи відстань до мембрани. Відповідно створюється ефект конденсатора, при якому між двома обкладинками



накопичується електричний заряд. У момент зарядки та розрядки пристрій випромінює звукову хвилю.

У випадку підключенні п'єзодинаміка до чутливих мікроконтролерів, не варто забувати, що у випадку механічної деформації п'єзодинаміка (удар, падіння), викликається прямий ефект. Після удару струм проходить по ланцюгу до контролера, що може вивести його з ладу. У такому разі потрібно обмежувати струм опором по входу п'єзодинаміка.

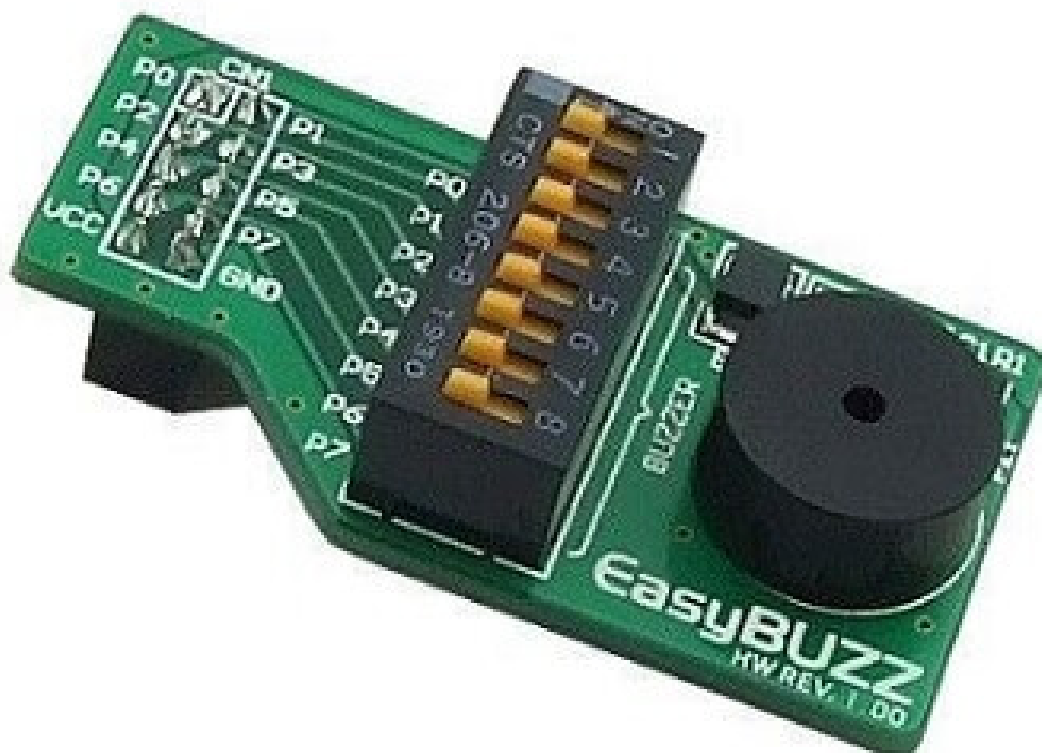


Рисунок 3.2 - П'єзодинамік для Arduino

LCD 1602 модуль для Arduino - використовується для перетворення сигналів від контролерів та датчиків у графічну інформацію (рис. 3.3).

Для використання LCD 1602 модуля Arduino потрібно на його основі зібрати макет (підключити до дисплея; контролера; живлення тощо). Потім потрібно створити програму для роботи з цим пристроєм. Наприкінці провести налаштування та розпочинати роботу.

Управління LCD 1602 модулем Arduino здійснюється або від Arduino контролера; або іншого мікропроцесорного керуючого пристрою через відповідні висновки.

LCD 1602 модуль Arduino з РК-дисплеєм має інтерфейс для введення виведення інформації.

4-бітний інтерфейс HD44780 має 16 позначених контактів:

- Vss - загальний GND;
- Vdd – харчування;
- VO – налаштування контрастності;
- RS – дані та команди;
- RW – читання та запис;
- E (Enable) – включення;
- DB0 - DB7 лінії даних;
- LEDA – анод підсвічування;
- LEDK – катод підсвічування.

Живлення модуля здійснюється або від Arduino контролера (іншого мікропроцесорного керуючого пристрою); або від зовнішнього джерела живлення (блок живлення; батареї). Напруга живлення модуля 5 вольт.

Характеристики:

- інтерфейс HD44780 сумісний;
- напруга живлення: 5В;
- розміри екрана: 64 x 15 мм;
- рідкокристалічний дисплей з синім підсвічуванням;
- дисплей 2 рядки по 16 символів;
- підтримка латиниці та кирилиці (із встановленням додаткових бібліотек);
- габарити: 79, 9 x 36, 0 x 11, 0 мм;
- вага: 31 р.



Рисунок 3.3 - LCD 1602 модуль для Arduino

Світлодіодний RGB є особливим типом світлодіодів, який складається із декількох простих світлодіодних масивів, що подібні до інших одноколірних світлодіодів. Тому, вони можуть випромінювати ці три основні кольори, при цьому створюючи таким чином всякі різні ефекти і кольори (навіть білий, що поєднує червоний, зелений і синій одночасно), просто керуючи одним із контактів цих компонентів.

Три упакованих світлодіода в такій же інкапсуляції здатні виробляти всю цю гаму кольорів. Він має трохи інший розмір, ніж звичайні світлодіоди, оскільки вони включають три виходи, по одному для кожного кольору (катоди або +) і ще один додатковий загальний для всіх, анод (-).

Завдяки типу напівпровідника можуть бути досягнуті різні кольори. Саме це і відрізняє червоні світлодіоди від зелених, жовтих, синіх і інших відтінків. Дослідники поєднують різні матеріали, щоб досягти всіх кольорів, які зараз існують на ринку. Наприклад:

Інфрачервоні світлодіоди використовують GaAs або AlGaAs як матеріали для випромінювання на цій довжині ІЧ-хвилі.

Червоний: AlGaAs, GaAsP, AlGaInP та GaP використовують у кольорових світлодіодах.

Помаранчевий: напівпровідникові матеріали, такі як GaAsP, AlGaInP, GaP, використовують з деякими варіаціями.

Амарілло: це може бути композиція, подібна до попередньої, така як GaAsP, AlGaInP та GaP, яка випромінює на довжині хвилі електромагнітний спектр, відповідний жовтому.

Зелений: для випромінювання на цій довжині хвилі потрібні спеціальні матеріали, такі як GaP, AlGaInP, AlGaP, InGaN / GaN.

Azul: у цьому випадку використовуються напівпровідники та легуючі речовини на основі таких матеріалів, як ZnSe, InGaN, SiC тощо.

Віолета: створено з InGaN.

Фіолетовий: щоб досягти цього кольору використовують подвійні сині і червоні світлодіоди. Пластик цього кольору навіть використовують з внутрішнім білим діодним світлом для отримання цього ефекту.

Роза: для отримання цього кольору немає матеріалу, щоб поєднати два світлодіоди різних кольорів для досягнення цього кольору, наприклад, червоного з жовтим тощо.

Вансо: саме цей колір породив сьогоденні світлодіодні лампи чисто білого або теплого білого кольорів. Для цього використовуються сині або УФ-світлодіоди з жовтим люмінофором для чистого білого або оранжевого світла.



Рисунок 3.4 - RGB-світлодіод

## 4 РОЗРОБКА СИСТЕМИ

### 4.1 Структурна схема пристрою

Парктронік для гаража на Arduino можна зробити в декількох варіантах. Використовувати цей проект на ультразвукових датчиках в автомобілі не рекомендується, оскільки вони мають велику похибку і не захищені від вологи. Розглянемо одразу кілька варіантів реалізації проекту на Arduino та оберемо найоптимальніший.

Насамперед необхідно вирішити проблему з хибними спрацьовуваннями датчика, зробивши в програмі функцію з декількома вимірами відстані та розрахунку середнього арифметичного значення (детальніше в п. 4.2).

Для складання парктроніка з екраном (рис. 4.1) слід підключити до плати ультразвуковий далекомір, п'єзодинамік і LCD екран, як на зображенні вище. У наведену вище програму залишається лише додати функцію виведення відстані на дисплей та умову для спрацьовування зумеру. Час між вимірюваннями та відстань при якому буде включений попереджувальний звуковий сигнал можна поміняти у скетчі (детальніше в п. 4.2).

Так як пристрій не рекомендується ставити на автомобіль через відсутність вологостійкості у датчиків HC-SR04, то і дисплей не має сенсу підключати до парктроніку Arduino Uno. Найкраще використовувати світлову індикацію. У наступному варіанті проекту використовуються червоні, жовті та зелені світлодіоди для індикації відстані від пристрою до автомобіля, що заїжджає до гаража (рис. 4.2).

У наступній програмі (додаток Г) реалізована функція вимкнення звукової сигналізації через 10 секунд (можна змінити програму), тобто. коли машина вже зупинилася. У схемі використані RGB світлодіоди 5 В, щоб для живлення

використовувати звичайну зарядку від телефону. Схема складання проекту на Arduino залишається незмінною (рис. 4.2).

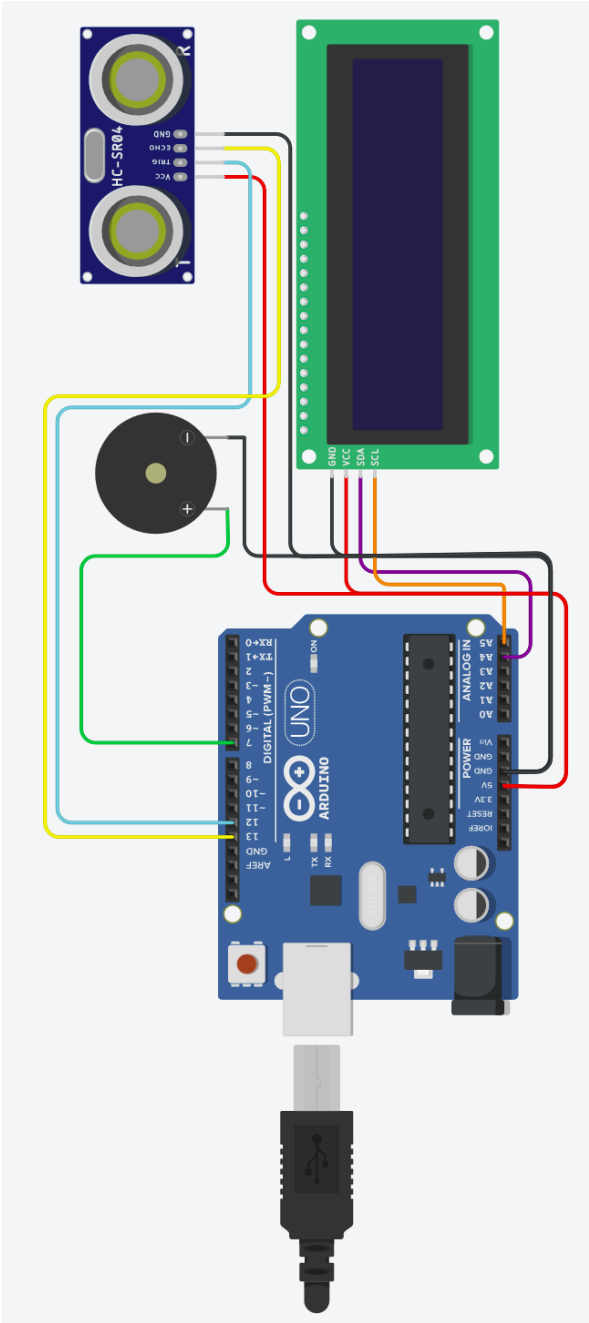


Рисунок 4.1 - Структурна схема пристрою (додаток Б)

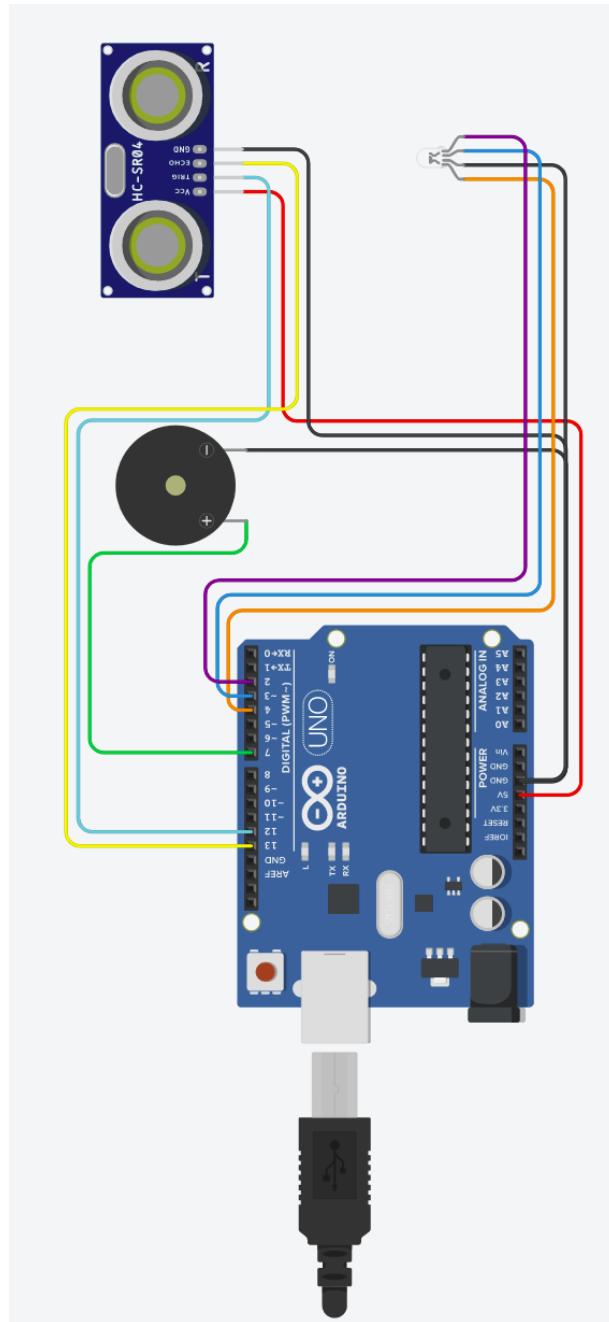


Рисунок 4.2 - Структурна схема пристрою (додаток В)

## 4.2 Програмна реалізація парктроніка



Насамперед необхідно вирішити проблему з хибними спрацьовуваннями датчика, зробивши в програмі функцію з декількома вимірами відстані та розрахунку середнього арифметичного значення. Для цього в циклі `for` виконуємо 10 вимірювань поспіль, складаючи отриманий результат у загальну суму, а після закінчення циклу отриману суму ділимо на кількість проведених вимірювань, тобто на 10 (додаток А). Перед виконанням циклу `for` необхідно обнулювати змінні `sum` і `total`, тому ми їх задаємо в циклі, при цьому вони стають рівними нулю. Алгоритм програми можна застосовувати до будь-якого датчика.

Для парктроніка з екраном (додаток Б) у наведену вище програму залишається лише додати функцію виведення відстані на дисплей та умову для спрацьовування зумеру. Час між вимірюваннями та відстань при якому буде включений попереджувальний звуковий сигнал можна поміняти у скетчі. Час затримки між вимірюваннями та відстань при спрацьовуванні п'єзодинаміка можна змінити на початку коду, змінивши значення `del` та `cm`. Кількість вимірів можна збільшувати, але це збільшить час виконання програми, не змінивши принципово якість виміру відстані.

У наступному варіанті проекту (додаток В) використовуються червоні, жовті та зелені світлодіоди для індикації відстані від пристрою до автомобіля, що заїжджає до гаража. На початку програми задаються змінні, які можна змінити, налаштувавши відповідний режим роботи автомобільного парктроніка на Arduino. Світлодіоди одного кольору можна підключати до одного піну.

У наступній програмі (додаток Д) реалізована функція вимкнення звукової сигналізації через 10 секунд (можна змінити програму), тобто коли машина вже зупинилася. У схемі використані RGB світлодіоди 5 В, щоб живлення використовувати звичайну зарядку від телефону. Зумер вимкнеться через заданий проміжок часу, завдяки зміні змінної  $t = 1$ ; при включенні п'єзодинаміки командою `tone()`. При відстані більше `cm4` змінна `t` знову стає рівною нулю і п'єзодинамік знову увімкнеться.

### 4.3 Моделювання конструкції для пристрою

Для захисту контактів майбутнього пристрою, датчиків та мікроконтролера необхідно змоделювати механічні частини для 3D-принтера. Матеріал, що використовується для виготовлення частин, називається PLA.

Усі частини конструкції, кудим буде поміщено пристрій, показані на рис. 4.3, 4.4, 4.5, 4.6.

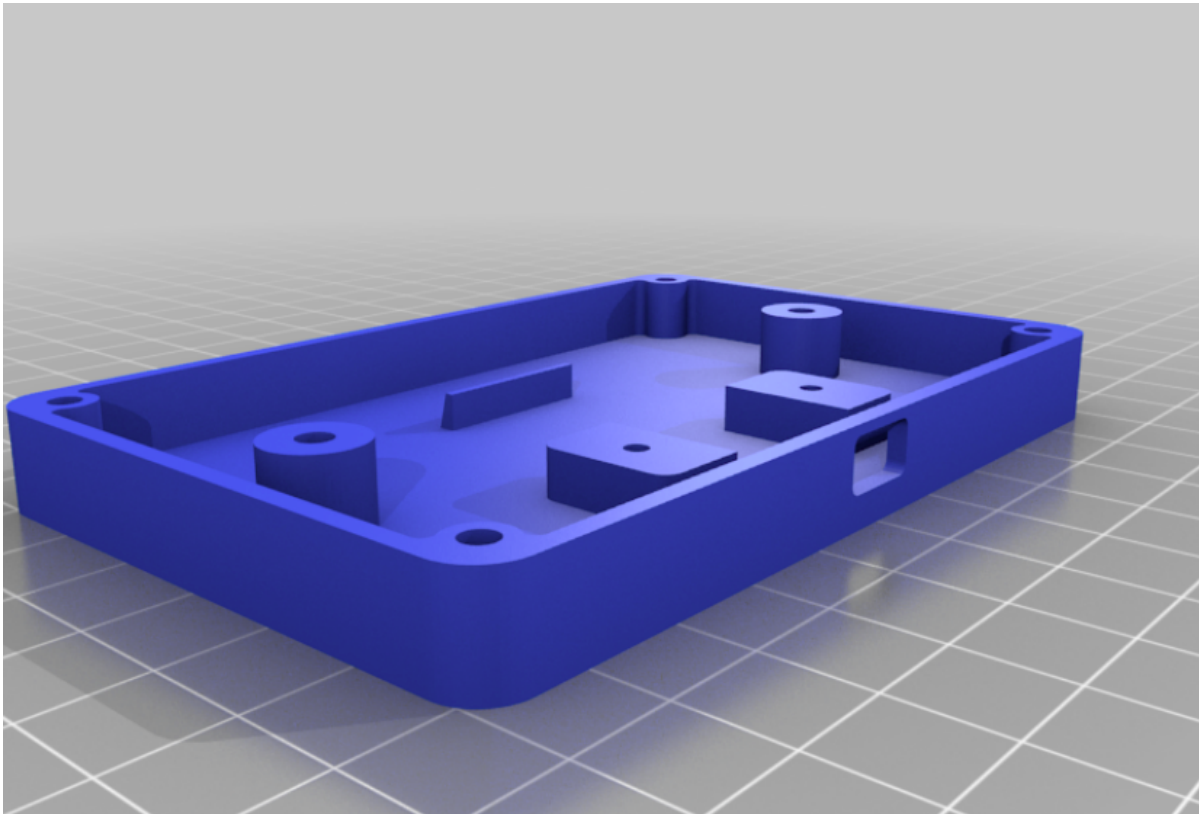


Рис. 4.3 - Нижня основа

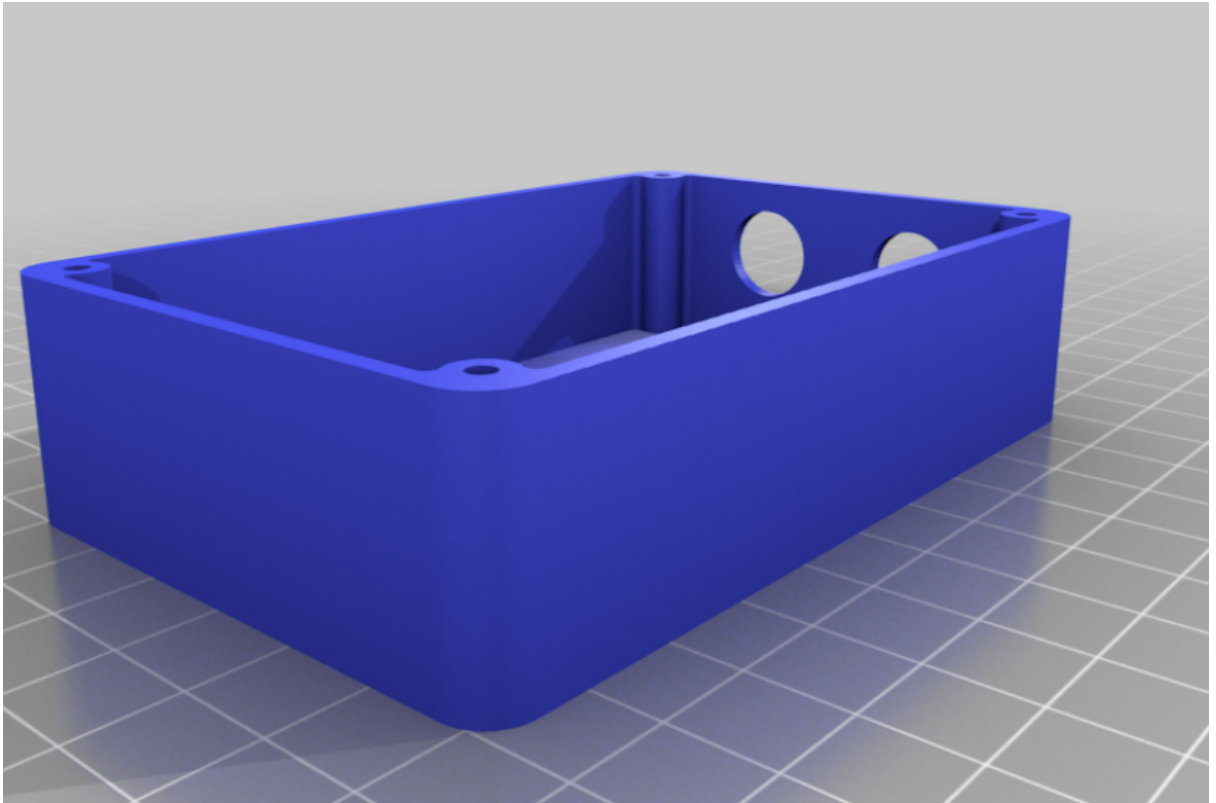


Рис. 4.4 - Бокова конструкція

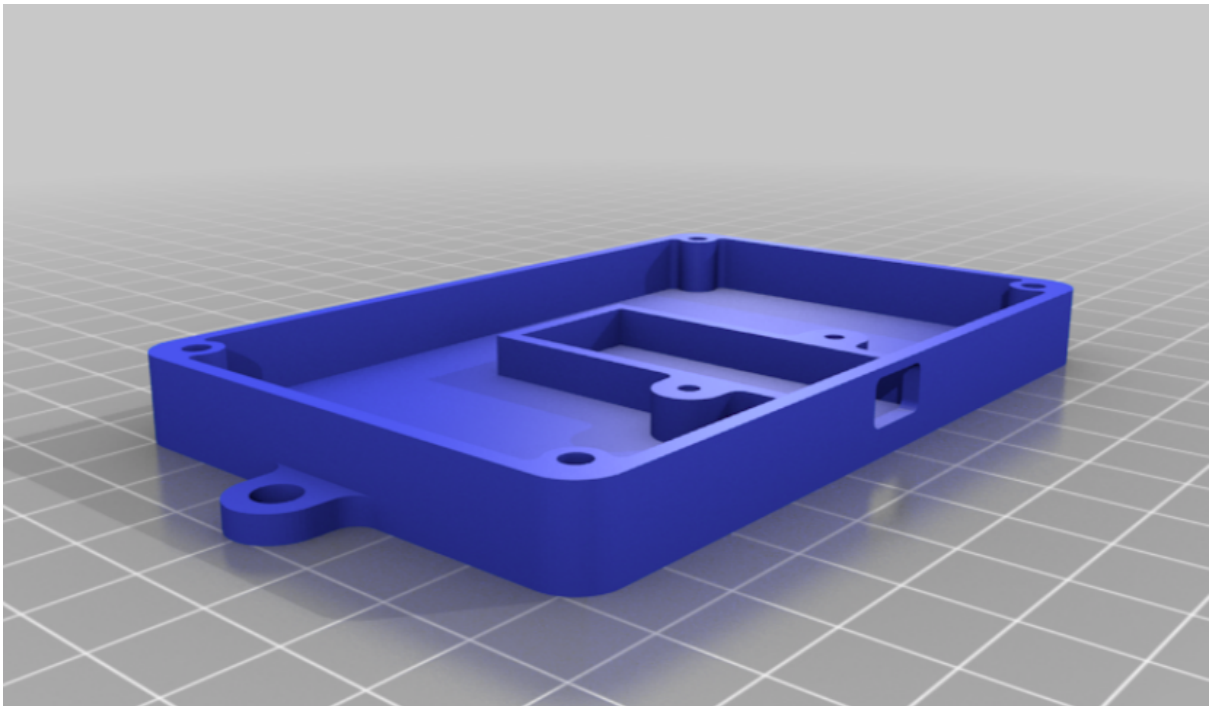


Рис. 4.5 - Верхня основа

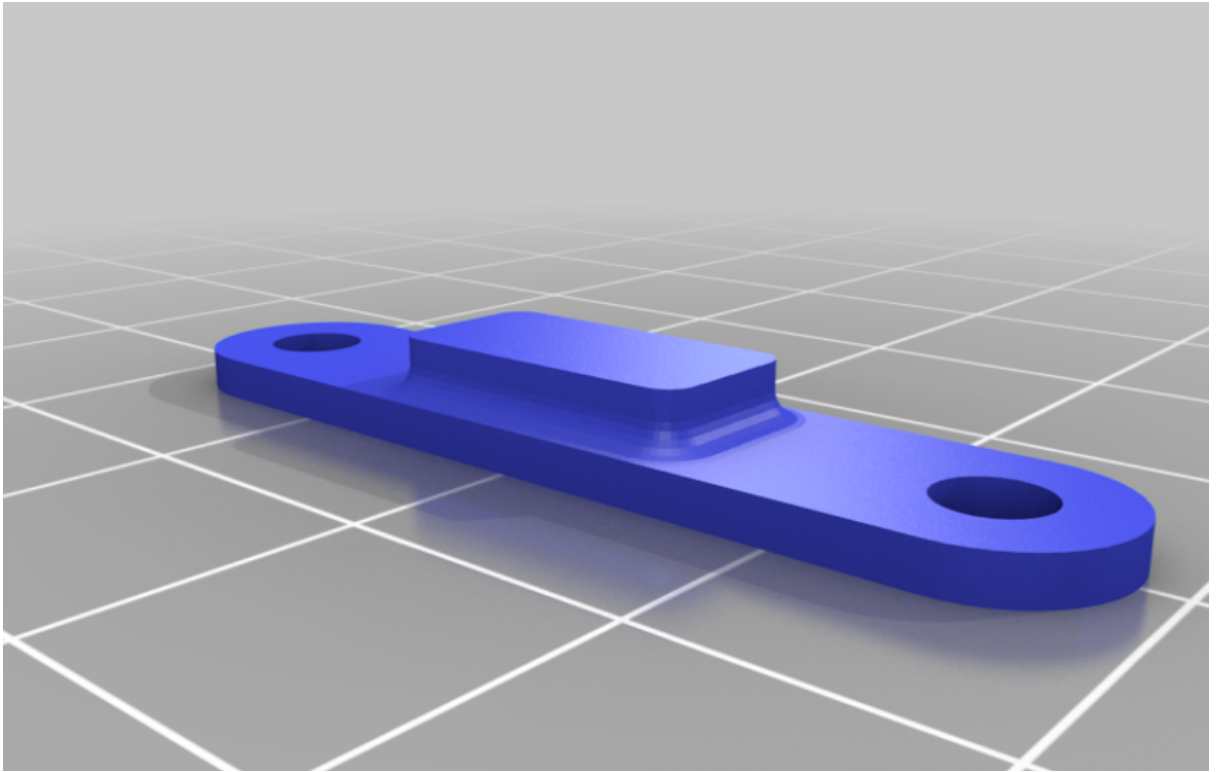


Рис. 4.6 - Бокове кріплення

## ВИСНОВКИ

У процесі розробки кваліфікаційної роботи була розроблена власна паркувальна система для безпечного паркування водієм та виконання рухів заднім ходом.

Перший розділ присвячено аналізу відомих рішень, їх порівняльній характеристиці, виявлено переваги та недоліки систем. Кожна технологія та пристрій має свої обмеження та індивідуальні можливості. Успішне застосування технології багато в чому залежить від правильного вибору пристрою. На вибір впливає багато факторів, такі як точність, дальність, напруга та сила струму на якій працює система, простота розробки, вартість та надійність. У роботі проаналізовано популярні паркувальні радары, виявлено їхні слабкі місця, що і зробила тему роботи актуальною. Окрім цього розглянуто різні типи та види парктроніків, загальні відомості про акустичну паркувальну систему (АПС) та її особливості використання. В другому розділі розглянуто середовище програмування Arduino IDE, платформа Arduino UNO та елементна база складових майбутнього пристрою, його програмна реалізація. У третьому розділі розроблено структурну схему проєкту та описано програмне забезпечення.

Для виявлення присутності перешкоди для автомобіля використано ультразвуковий датчик відстані та допоміжні компоненти. Для реалізації було використано мікроконтролер Arduino.

Дана розробка є чудовим варіантом для автомобілів без паркувального радару.

Проведені тести дозволяють переконатись у працездатності та ефективності системи, тому дану систему можна комерціалізувати. Для цього необхідно змоделювати основу, куди буде поміщено пристрій.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Arduino [Електронний ресурс] – Доступ: <https://uk.wikipedia.org/wiki/Arduino>
2. Amazon Echo [Електронний ресурс] – Доступ: [https://ru.wikipedia.org/wiki/Amazon\\_Echo](https://ru.wikipedia.org/wiki/Amazon_Echo)
3. Amazon Alexa [Електронний ресурс] – Доступ: [https://uk.wikipedia.org/wiki/Amazon\\_Alexa](https://uk.wikipedia.org/wiki/Amazon_Alexa)
4. Розробки під Arduino [Електронний ресурс] – Доступ: <https://habr.com/ru/hub/arduino/>
5. IBM Global Parking Survey: Drivers Share Worldwide Parking Woes [Електронний ресурс] — Режим доступу: <https://www-03.ibm.com/press/us/en/pressrelease/35515.wss>
6. Smart Parking Systems [Електронний ресурс] — Режим доступу: <https://smartparkingsystems.com/en/>
7. Офіційний сайт компанії Микком [Електронний ресурс] — Режим доступу: <http://miccom.ru/products/pm-about/pm1>
8. «Розумні парковки» - новий підхід до вирішення проблеми паркінгу в містах. Офіційний сайт компанії Акіома [Електронний ресурс] — Режим доступу: <http://www.aksioma-group.ru/proekty/sistema-monitoringa-parkovochnykh-mest-vpredelakh-bulvarnogo-koltsa-moskvy/>
9. Документація по технології React Native [Електронний ресурс] — Режим доступу: <https://reactnative.dev/> 6. Документація по хмарному сервісу Heroku [Електронний ресурс] — Режим доступу: <https://www.heroku.com/platform>
10. Документація по експлуатації інфрачервоного датчика IR Sharp GP2Y0D02YK0F [Електронний ресурс] — Режим доступу: [https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0d02yk\\_e.pdf](https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0d02yk_e.pdf)
11. Стандарт JSON Web Token (JWT) [Електронний ресурс] — Режим доступу:

<https://tools.ietf.org/html/rfc7519>

12. Паркувальний радар [Електронний ресурс] – Режим доступу до ресурсу:  
[https://uk.wikipedia.org/wiki/Паркувальний\\_радар](https://uk.wikipedia.org/wiki/Паркувальний_радар)
13. Підбираємо паркувальний радар – електромагнітний чи ультразвуковий?  
[Електронний ресурс] – Режим доступу до ресурсу:  
<https://130.com.ua/uk/ultrazvuk-elektromagnit-parktronik/>
14. Як правильно вибрати та купити парктронік? [Електронний ресурс] – Режим доступу до ресурсу: <https://130.com.ua/uk/parktronic/>
15. Парктронік для авто [Електронний ресурс] – Режим доступу до ресурсу:  
<https://hi-news.pp.ua/tehnka-tehnologyi/7042-parktronki-dlya-avto-vidiharakteristiki-ustanovka-priznachennya-vibr.html>
16. Parking Sensor with Arduino [Електронний ресурс] – Режим доступу до ресурсу:  
<https://create.arduino.cc/projecthub/vinicius-lobes/parking-sensor-with-arduino-53e598>
17. How to Make a Car Parking Sensor Using Arduino? [Електронний ресурс] – Режим доступу до ресурсу:  
<https://appuals.com/how-to-make-a-car-parking-sensor-using-arduino/>
18. Car Parking sensor with Arduino with distance Sound effects [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.circuitschools.com/car-parking-sensor-with-arduino-with-distance-sound-effects/>
19. Arduino Car Reverse Parking Sensor [Електронний ресурс] – Режим доступу до ресурсу: <https://www.electronicshub.org/arduino-car-reverse-parking-sensor/>
20. Arduino Based Car Parking Assistant [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.instructables.com/DIY-Arduino-Based-Car-Parking-Assistant/>
21. How To Make Parking Assist Car Sensors [Електронний ресурс] – Режим доступу до ресурсу:

- <https://www.deviceplus.com/arduino/how-to-make-parking-assist-car-sensors/>
22. Car Parking Sensor [Электронный ресурс] – Режим доступа до ресурсу:  
<https://www.hackster.io/omerfettahoglu/car-parking-sensor-5f6fb4>
23. How to Make an Arduino Ultra-Sound Car Parking Sensor [Электронный ресурс] – Режим доступа до ресурсу:  
<https://www.instructables.com/Arduino-Parking-Sensor/>
24. Parking sensor [Электронный ресурс] – Режим доступа до ресурсу:  
[https://en.wikipedia.org/wiki/Parking\\_sensor](https://en.wikipedia.org/wiki/Parking_sensor)
25. Ford Motor Company [Электронный ресурс] – Режим доступа до ресурсу:  
[https://en.wikipedia.org/wiki/Ford\\_Motor\\_Company](https://en.wikipedia.org/wiki/Ford_Motor_Company)
26. Parking Sensors [Электронный ресурс] – Режим доступа до ресурсу:  
<https://mycardoeswhat.org/safety-features/parking-sensors/>
27. Vehicle detection sensors [Электронный ресурс] – Режим доступа до ресурсу:  
<https://www.smartparking.com/smartpark-system/smart-sensors>
28. Parking Sensors and Cameras [Электронный ресурс] – Режим доступа до ресурсу:  
<https://www.sencor.com/parking-sensor-system/sca-park200>
29. Light Bulb Base Injection Mold [Электронный ресурс] – Режим доступа до ресурсу:  
<https://www.plasticmoldfactory.com/product/light-bulb-base-injection-mold>
30. Parking Sensors [Электронный ресурс] – Режим доступа до ресурсу:  
<https://luview.com/products/parking-sensors/>
31. Front or Rear Parking Sensor System Model: PSD100D [Электронный ресурс] – Режим доступа до ресурсу:  
[https://sc.voxxintl.com/docs/common/PSD100D/PSD100D\\_IM.pdf](https://sc.voxxintl.com/docs/common/PSD100D/PSD100D_IM.pdf)
32. Universal 4-Sensor Switchable Front or Rear Parking Sensor System 9002-3003 [Электронный ресурс] – Режим доступа до ресурсу:  
<https://brandmotion.com/product/universal-4-sensor-switchable-front-or-rear-parking-sensor-system-9002-3003/>
33. Car parking sensor system with 8x Sensors + 1 LCD Display Valeo beep&park



[Электронный ресурс] – Режим доступа до ресурсу:

<https://www.valeoservice.com/en-com/passenger-car/driving-assistance-and-parking-assistance-systems-cars/car-parking-sensor-system-8x>

34. How to Use the Parking Sensor System [Электронный ресурс] – Режим доступа до ресурсу:

<https://www.hondainfocenter.com/2021/Accord/How-To-Videos/EX-L/Notable-Features/How-to-Use-the-Parking-Sensor-System/>

## Додаток А

```
#include <Ultrasonic.h>      // імпортуємо бібліотеку
Ultrasonic ultrasonic (11, 12); // вказуємо піни для уз-датчика

void setup() {
  Serial.begin(9600);
}

void loop() {
  // виконуємо 10 вимірювань і додаємо результат
  int distance, sum, total;
  for (byte i = 0; i <= 10; i++) {
    distance = ultrasonic.Ranging(CM);
    sum = sum + distance;
  }
  total = sum / 10; // розраховуємо середнє значення
  Serial.println("Distance - " + String(total));
}
```

## Додаток Б

```
#include <Ultrasonic.h>          // імпортуємо бібліотеку
Ultrasonic ultrasonic (11, 12); // вказуємо піни для уз-датчика

#include <Wire.h>
#include <LiquidCrystal_I2C.h> // імпортуємо бібліотеку для дисплея
LiquidCrystal_I2C LCD(0x27,20,4);

int del = 5; // затримка між змінами в мкс
int cm = 10; // відстань в см при якій спрацьовує зумер

void setup() {
  Serial.begin(9600);
  LCD.init();
  LCD.backlight();
  pinMode(7, OUTPUT); // пін для підключення зумера
}

void loop() {
  // виконуємо 10 вимірювань і додаємо результат
  int distance, sum, total;
  for (byte i = 0; i <= 10; i++) {
    distance = ultrasonic.Ranging(CM);
    sum = sum + distance;
    delay(del);
  }
  total = sum / 10; // розраховуємо середнє значення
  Serial.println("Distance - " + String(total));
}
```

```
LCD.clear(); // очищуємо екран дисплею і виводимо відстань
LCD.setCursor(0,0);
LCD.print("Distance - ");
LCD.print(total);

if (total > cm) { noTone(7); }
if (total <= cm) { tone(7,100); }
}
```

## Додаток В

```
#include <Ultrasonic.h>          // імпортуємо бібліотеку
Ultrasonic ultrasonic (11, 12); // вказуємо піни для уз-датчика

int del = 5;    // затримка між вимірами в мкс
int cm1 = 30;  // відстань за якої увімкнеться зелений світлодіод
int cm2 = 20;  // відстань за якої увімкнеться синій світлодіод
int cm3 = 10;  // відстань за якої увімкнеться червоний світлодіод
int cm4 = 5;   // відстань за якої увімкнеться зумер

void setup() {
  Serial.begin(9600);
  pinMode(7, OUTPUT); // пін для підключення зумера
  pinMode(2, OUTPUT); // пін для підключення зеленого світлодіода
  pinMode(3, OUTPUT); // пін для підключення синього світлодіода
  pinMode(4, OUTPUT); // пін для підключення червоного світлодіода
}

void loop() {
  // виконуємо 10 вимірювань і додаємо результат
  int distance, sum, total;
  for (byte i = 0; i <= 10; i++) {
    distance = ultrasonic.Ranging(CM);
    sum = sum + distance;
    delay(del);
  }
  total = sum / 10; // розраховуємо середнє значення
  Serial.println("Distance - " + String(total));

  if (total >= cm1) {
    digitalWrite(4, LOW); digitalWrite(3, LOW); digitalWrite(2, LOW);
  }
  if (total < cm1 && total >= cm2) {
    digitalWrite(4, LOW); digitalWrite(3, LOW); digitalWrite(2, HIGH);
  }
  if (total < cm2 && total > cm3) {
```

```
    digitalWrite(4,LOW); digitalWrite(3,HIGH); digitalWrite(2,LOW);
}
if (total <= cm3) {
    digitalWrite(4,HIGH); digitalWrite(3,LOW); digitalWrite(2,LOW);
}

if (total > cm4) { noTone(7); }
if (total <= cm4) { tone(7,100); }
}
```

## Додаток Г

```
#include <Ultrasonic.h>          // імпортуємо бібліотеку
Ultrasonic ultrasonic (11, 12); // вказуємо пini для уз-датчика

int del = 5;    // затримка між вимірами в мкс
int cm1 = 30;  // відстань за якої увімкнеться зелений світлодіод
int cm2 = 20;  // відстань за якої увімкнеться синій світлодіод
int cm3 = 10;  // відстань за якої увімкнеться червоний світлодіод
int cm4 = 5;   // відстань за якої увімкнеться зумер

unsigned long currentTime; // змінна для відрахунку часу
int Time = 5000; // через скільки мкс потрібно відключити зумер
byte t = 0; // змінна для запам'ятовування стану зумера

void setup() {
    Serial.begin(9600);
    pinMode(7, OUTPUT); // пін для підключення зумера
    pinMode(2, OUTPUT); // пін для підключення зеленого світлодіода
    pinMode(3, OUTPUT); // пін для підключення синього світлодіода
    pinMode(4, OUTPUT); // пін для підключення червоного світлодіода
}

void loop() {
    // виконуємо 10 вимірювань і додаємо результат
    int distance, sum, total;
    for (byte i = 0; i <= 10; i++) {
        distance = ultrasonic.Ranging(CM);
        sum = sum + distance;
```

```
    delay(del);
}
total = sum / 10; // розраховуємо середнє значення
Serial.println("Distance - " + String(total));

if (total >= cm1) {
    digitalWrite(4, LOW); digitalWrite(3, LOW); digitalWrite(2, LOW);
}
if (total < cm1 && total >= cm2) {
    digitalWrite(4,LOW); digitalWrite(3,LOW); digitalWrite(2,HIGH);
}
if (total < cm2 && total > cm3) {
    digitalWrite(4,LOW); digitalWrite(3,HIGH); digitalWrite(2,LOW);
}
if (total <= cm3) {
    digitalWrite(4,HIGH); digitalWrite(3,LOW); digitalWrite(2,LOW);
}
// зумер вимкнеться через заданий проміжок часу
if (total > cm4) { t = 0; noTone(7); }
if (t == 0 && total <= cm4) { t = 1; tone(7, 100); currentTime = millis(); }
if (t == 1 && total <= cm4 && millis() - currentTime > Time) { t = 1; noTone(7); }
}
```



# ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35

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

Магістерська робота на тему:

«РОЗРОБКА ПАРКУВАЛЬНОГО РАДАРУ ДЛЯ АВТОМОБІЛЬНОГО ТРАНСПОРТУ НА БАЗІ ARDUINO»

Студент ІСДМ-61 групи  
Кульчицький О.Є.  
Науковий керівник  
д.ф., доц. каф. ІПЗАС  
Тушич А.М.

Київ - 2023 р.



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35

*Об'єкт дослідження – процес розробки паркувального радару для автомобільного транспорту на базі Arduino.*

*Предмет дослідження – створення паркувального радару.*

*Мета роботи – дослідити методи створення автоматизованої системи паркування для автомобілів.*

*Завдання:*

- Аналіз існуючих рішень
- Аналіз актуальних вимог до системи
- Вибір та обґрунтування елементів та технологій
- Розробка структурної схеми системи
- Моделювання механічних частин системи
- Написання програмного забезпечення для обслуговування системи



## Аналіз існуючих рішень



Характеристики	Falcon 2616-4 black	Falcon 858K black	Tiger TG-P8 black
Низька вартість	+	-	-
Низький робочий струм	-	-	-
Низька напруга роботи	-	-	-
Низька мінімальна дальність спрацювання (<20 см)	-	+	+
Велика максимальна дальність спрацювання (>200 см)	-	-	+



## Аналіз актуальних вимог до системи

### Система може включати:

- електронний замок;
- випромінювачі ультразвукового датчика;
- пристрої відображення (РК-дисплей);
- звукові сповіщення (зумер).

### Паркувальний радар може генерувати помилкові сигнали в таких випадках:

- наявність льоду, снігу або іншого сміття на датчику;
- перебування на нерівній поверхні, ґрунтовій, похилій дорозі;
- міжміський рух;
- наявність джерел з підвищеним шумом в зоні дії датчика;
- робота під сильним дощем або снігопадом;
- робота радіопередавачів в зоні дії датчика;
- буксирування причіпу;
- паркування в обмежених умовах (ефект луни).

### Система може не реагувати на наступне:

- гострі або тонкі предмети, наприклад, ланцюги, троси, тонкі стовпи;
- предмети, що поглинають ультразвукове випромінювання (одяг, пористі матеріали, сніг);
- предмети висотою менше 1 метра;
- об'єкти, що відбивають звук від датчиків;
- система не може виявити ями в асфальті, відкриті люки, розкидані дрібні гострі предмети та інші небезпечні предмети, які знаходяться поза полем зору датчиків.

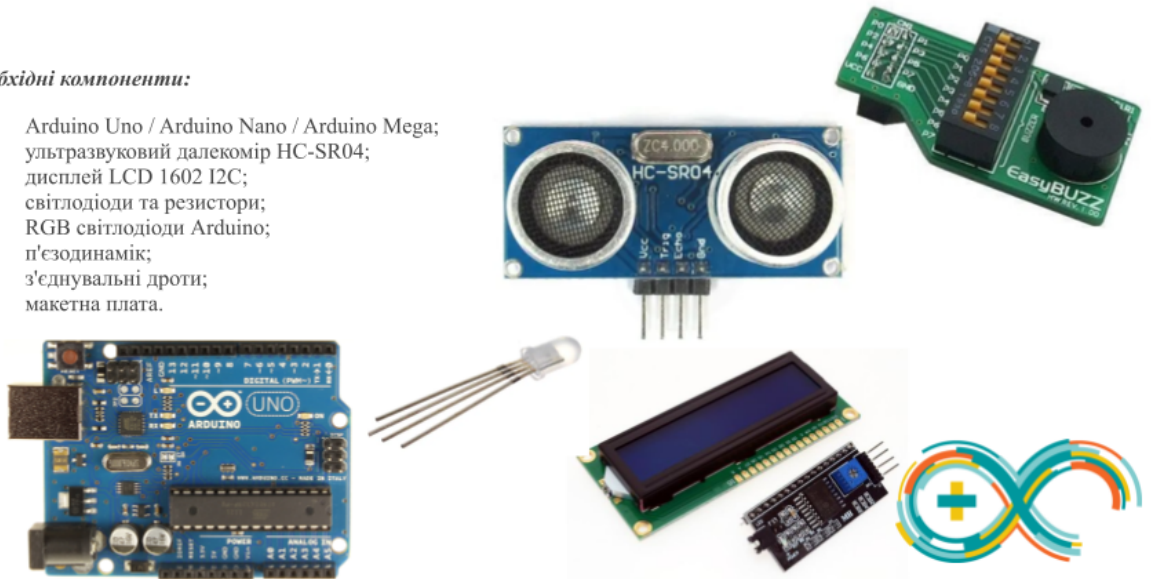


1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35

## Вибір та обґрунтування елементів та технологій

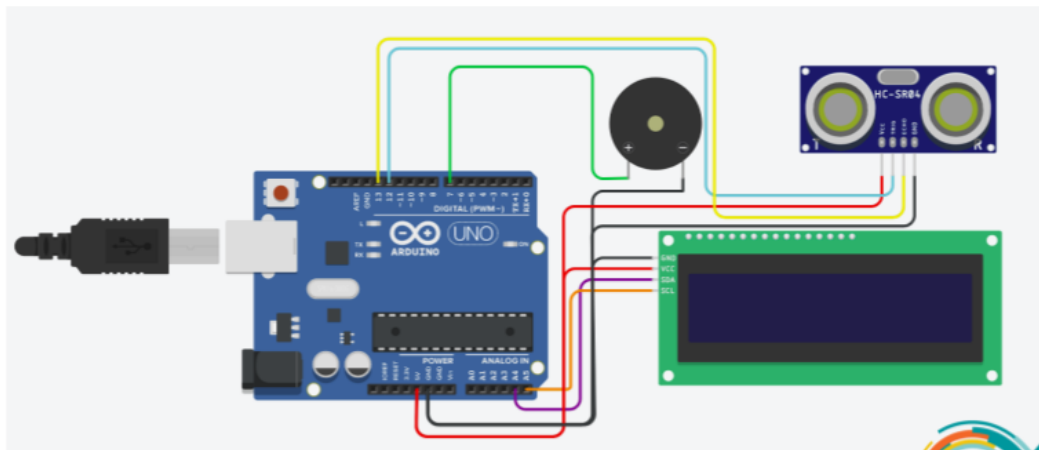
### Необхідні компоненти:

- Arduino Uno / Arduino Nano / Arduino Mega;
- ультразвуковий далекомір HC-SR04;
- дисплей LCD 1602 I2C;
- світлодіоди та резистори;
- RGB світлодіоди Arduino;
- п'єзодинамік;
- з'єднувальні дроти;
- макетна плата.



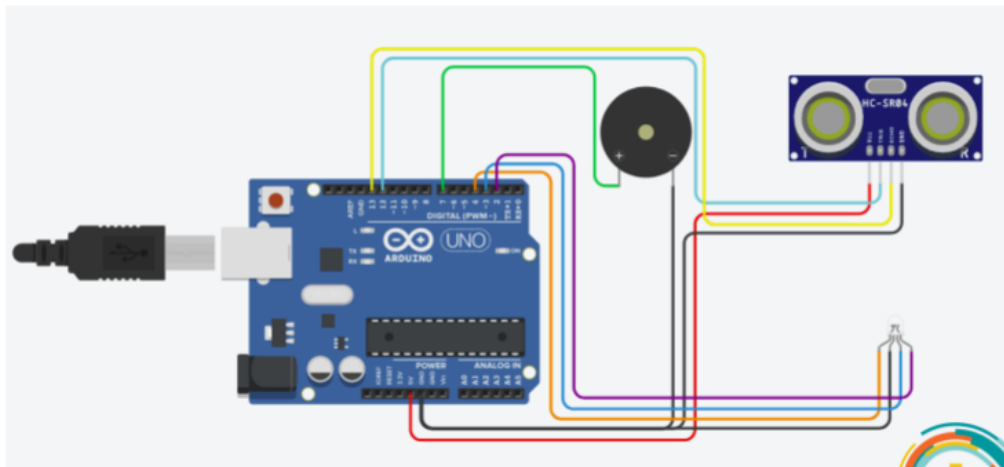
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35

## Початкова структурна схема пристрою



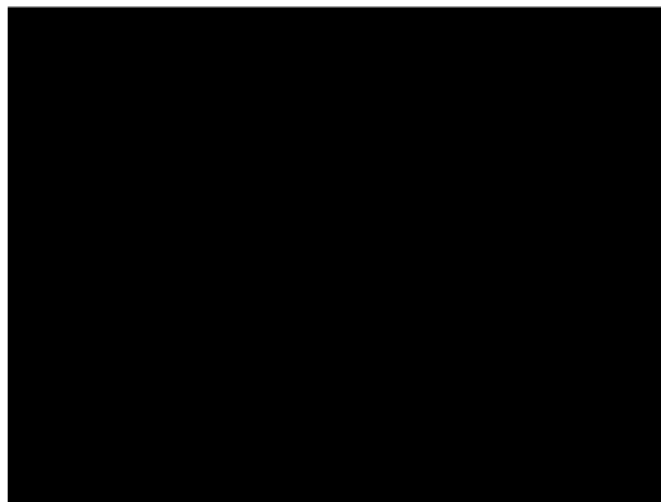
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35

## Фінальна структурна схема пристрою



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35

## Моделювання механічних частин системи



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35

## Висновки

- Визначено об'єкт, предмет, мету та завдання, які необхідно вирішити в роботі.
- Проведено аналіз предметної області та популярних паркувальних радарів, виявлено їхні слабкі місця, що і зробило тему роботи актуальною.
- Розроблено структурні схеми проєкту на описано програмне забезпечення.
- Розглянуто середовище програмування Arduino IDE, платформу Arduino UNO та елементну базу складових майбутнього пристрою, його програмна реалізація.
- Для виявлення присутності перешкоди для автомобіля використано ультразвуковий датчик відстані та допоміжні компоненти.
- Як результат проєктування, створено власну паркувальну систему для безпечного паркування водієм та виконання рухів заднім ходом.
- Проведені тести дозволяють переконатись у працездатності та ефективності системи, тому дану систему можна комерціалізувати. Для цього було змодельовано основу, куди буде поміщено пристрій.

Апробація:  
Всеукраїнська науково-технічна конференція "Сучасний стан та перспективи розвитку IoT" 10 травня 2022 року  
Тема: "РОЗРОБКА ПАРКУВАЛЬНОГО РАДАРУ ДЛЯ АВТОМОБІЛЬНОГО ТРАНСПОРТУ НА БАЗІ ARDUINO"



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35

Дякую за увагу!

