

БАКАЛАВРСЬКА РОБОТА

БР.КІ-19.00.00.000 ПЗ

Група КІ-22-1

Плахетко Ярослав

2026

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу
Факультет інформаційних технологій
Кафедра комп'ютерних систем і мереж

Плахетко Ярослав Юрійович

(прізвище, ім'я, по батькові)

УДК 004.415
(індекс)

БАКАЛАВРСЬКА РОБОТА

Розроблення мікроконтролерної системи детектування розташування людей в приміщенні на основі ESP32

(назва роботи)

Комп'ютерна інженерія

(назва освітньої програми)

123 – комп'ютерна інженерія

(шифр і назва спеціальності)

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

Здобувач освітнього ступеня _____

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник _____

Заячук Ярослав Іванович, к.т.н., доцент

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

Допущено до захисту

Завідувач кафедри КСМ

(посада)

С. І. Мельничук

(підпис)

(дата)

(ініціали та прізвище)

Івано-Франківський національний технічний університет нафти і газу

Інститут Інформаційних технологій

Кафедра Комп'ютерних систем та мереж

Спеціальність 123 – Комп'ютерна інженерія

Освітній рівень бакалавр

ЗАТВЕРДЖУЮ

Зав. кафедрою КСМ

С. І. Мельничук

“ ” 2026 р.

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Плахетку Ярославу Юрійовичу

(прізвище, ім'я, по-батькові)

1. Тема дипломної роботи “ Розроблення мікроконтролерної системи детектування розташування людей в приміщенні на основі ESP32 ”

Керівник роботи Заячук Ярослав Іванович

затверджена наказом по університету від “21.04.2026” № 180/7

2. Термін здачі студентом закінченої роботи 11.06.2026

3. Вихідні дані до проекту (роботи) Матеріали та результати, отримані під час проходження переддипломної практики, методичні вказівки, технічна література

4. Зміст розрахунково - пояснювальної записки (перелік питань, що їх належить розробити)

Аналіз технологій визначення розташування людей у приміщенні

Розроблення структури мікроконтролерної системи

Реалізація проектних рішень

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти по дипломній роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 02 лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів бакалаврської роботи	Термін виконання етапів роботи	Примітка
1	Визначення теми дипломної роботи	02.02.26 – 10.02.26	Виконано
2	Збір інформації. Дослідження предметної області	11.02.26 – 28.02.26	Виконано
3	Формування першого розділу пояснювальної записки	01.03.26 – 31.03.26	Виконано
4	Формування другого розділу пояснювальної записки	01.04.26 – 30.04.26	Виконано
5	Формування третього розділу пояснювальної записки	01.05.26 – 31.05.26	Виконано
6	Оформлення роботи	01.06.26 – 10.06.26	Виконано

Студент

(підпис)

Плахетко Я. Ю.
(прізвище та ініціали)

Керівник роботи

(підпис)

Заячук Я. І.
(прізвище та ініціали)

АНОТАЦІЯ

Дипломна робота зосереджений на створенні IoT-рішення, яке дозволяє відстежувати та зберігати дані про переміщення осіб у межах приміщення. У роботі розглянуто ключові підходи й сучасні інструменти для визначення місцезнаходження всередині приміщень, зокрема методи трилатерації, триангуляції та використання радіовідбитків. Проведено детальний аналіз характеристик і здійснено підбір оптимальних параметрів, необхідних для коректного позиціонування.

Окрему увагу приділено визначенню показників, що застосовуються під час обчислення координат користувачів у внутрішньому просторі. Також виконано огляд і порівняння наявних IoT-платформ, що дозволило обґрунтувати вибір відповідних технологічних рішень. На основі отриманих результатів сформовано загальну структуру системи та описано її архітектурні особливості.

IPS, CLOUD, РАДІОМІТКА, МІКРОКОНТРОЛЕР, MQTT

ABSTRACT

The thesis is devoted to the development of an IoT system that determines and records the history of people's positioning indoors.

The main methods and technologies for indoor localization (trilateration, triangulation, and radio fingerprinting) are studied. An analysis is carried out and parameters for localization are selected. Metrics required for calculating the coordinates of people indoors are determined. Existing IoT platforms are analyzed. The system architecture is designed.

Interaction diagrams of system components and the functioning of the IoT system are developed. Algorithms and their software implementation for Bluetooth signal scanning and coordinate calculation of radio tags are presented. The detection of human movement contacts and the storage of their history are implemented. A report with graphical visualization of the tag movement history and contact points with other tags is provided.

IPS, CLOUD, RADIO TRACKING, MICROCONTROLLER, MQTT

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	4
ВСТУП.....	5
1 АНАЛІЗ ТЕХНОЛОГІЙ ВИЗНАЧЕННЯ КООРДИНАТ У ЗАКРИТОМУ СЕРЕДОВИЩІ.....	7
1.1 Основні методи локалізації	7
1.2 Параметри локалізації в системах IPS	10
1.3 Технології зв'язку в системах IPS.....	13
1.4 Мікроконтролери для систем автоматизації «розумного будинку»	16
1.5 Постановка завдання.....	23
2 РОЗРОБЛЕННЯ СТРУКТУРИ МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ	25
2.1 Архітектура IoT системи	25
2.2 Протоколи передачі даних в IoT.....	29
2.3 Апаратне та програмне забезпечення системи	31
2.4 Проєктування структури системи	34
2.5 UML-діаграми системи.....	39
2.6 Висновки до розділу	42
3 РЕАЛІЗАЦІЯ СИСТЕМИ.....	43
3.1 Збір даних. MQTT	43
3.2 Обчислення координат об'єктів	49
3.3 Реалізація клієнтської частини	52
3.4 Висновки до розділу	60
ВИСНОВКИ.....	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	62

					БР.КІ-19.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розроб.		Плахетко Я. Ю.			Розроблення мікроконтролерної системи детектування розташування людей в приміщенні на основі ESP32	Літ.	Арк.	Аркушів
Перевір.		Заячук Я.І.					3	67
Реценз.						ІФНТУНГ, КІ-22-1		
Н. Контр.		Лазарів А.М.						
Затверд.		Мельничук С.І.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AOA – Angle Of Arrival

BLE – Bluetooth Low Energy

CSI – Channel State Information

GCP – Google Cloud Platform

GPS – Global Positioning System

IoT – Internet Of Things

IPS – Indoor Positioning System

MQTT – Message Queuing Telemetry Transport

OPS – Outdoor Positioning System

OFDM – Orthogonal Frequency-Division Multiplexing

POA – Phase Of Arrival

RFID – Radio Frequency Identification

RSSI – Received Signal Strength Indication

TDOA – Time Difference Of Arrival

TOA – Time Of Arrival

UWB – Ultra-Wide Band

Wi-Fi – Wireless Fidelity

БД – база даних

МК – мікроконтролер

ОС – операційна система

ПЗ – програмне забезпечення

					<i>БР.КІ-19.00.00.000 ПЗ</i>	Арк.
						4
Зм.	Арк.	№ докум.	Підп.	Дата		

ВСТУП

Протягом останніх років спостерігається інтенсивне вдосконалення технологій бездротового визначення координат, що особливо позначилося на розвитку концепції IoT [1,2]. Сучасні рішення цього класу застосовуються не лише для орієнтування на місцевості, а й у багатьох галузях – зокрема для ідентифікації місцезнаходження користувачів, пристроїв і матеріальних об'єктів, у фінансових сервісах, логістичних системах і транспортній інфраструктурі [3].

Сьогодні прийнято розрізняти дві основні категорії систем визначення координат: зовнішні OPS (Outdoor Positioning System) та внутрішні IPS (Indoor Positioning System). До першої групи належать, наприклад, Global Positioning System і BeiDou Navigation Satellite System. Такі рішення ефективно працюють на відкритому просторі, проте їх використання в будівлях є обмеженим через відсутність прямої видимості супутників, вплив перешкод, багатопроменевість сигналу та інші фактори [4].

Внутрішні системи позиціонування (IPS), навпаки, орієнтовані на роботу в закритих просторах, де супутникові сигнали недоступні або нестабільні [5]. Вони активно розвиваються і демонструють значний попит у різних сферах. За актуальними оцінками аналітичних компаній, глобальний ринок таких рішень уже наближається до десятків мільярдів доларів і продовжує зростати швидкими темпами у 2026 році [6]. Подібні технології застосовуються для навігації автономних роботів, відстеження переміщення людей та об'єктів у режимі реального часу, у медичних системах для контролю стану пацієнтів, у складській логістиці, а також у системах безпеки підприємств [7].

Актуальність теми дослідження. Контроль місцеперебування осіб у межах будівель відкриває можливість оперативного виявлення людей, які відповідають визначеним критеріям, і дозволяє впроваджувати вибіркові обмеження їхнього переміщення без необхідності повного припинення роботи об'єкта. Такий підхід є більш гнучким і економічно доцільним.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підп.	Дата		

Суттєву роль у розвитку подібних рішень відіграє доступність недорогих сенсорів і пристроїв, які можуть взаємодіяти між собою завдяки сучасним IoT-платформам і протоколам зв'язку [8]. Це дає змогу створювати малопомітні, енергоефективні та зручні для користувачів системи моніторингу, включно з рішеннями для виявлення контактів між людьми в приміщеннях. У зв'язку з цим розробка мікроконтролерної системи є актуальним завданням.

Метою дипломного проектування є розробка системи, що забезпечує відстеження переміщень осіб у межах приміщення з можливістю накопичення відповідних даних.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати технології визначення координат у закритому середовищі;
- обрати параметри, що використовуються для обчислення положення людини в приміщенні;
- організувати збереження історії переміщень користувачів із урахуванням вимог до безпеки даних;
- розробити та протестувати прототип системи.

Об'єктом дослідження є процес відстеження переміщення людей у закритих просторах.

Предметом дослідження є технології та програмно-апаратні засоби реалізації системи відстеження і збереження даних про переміщення осіб у приміщенні на основі IoT.

Методи дослідження. У роботі використані загальнонаукові емпіричні та теоретичні методи дослідження.

Практичне значення одержаних результатів полягає у можливості, наприклад, оперативного виявлення контактних осіб і подальшого адресного обмеження їхньої мобільності. Це, своєю чергою, сприяє ефективнішому контролю поширення інфекційних захворювань і підвищенню рівня безпеки в організаціях без повного зупинення їх діяльності.

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		6

1 АНАЛІЗ ТЕХНОЛОГІЙ ВИЗНАЧЕННЯ КООРДИНАТ У ЗАКРИТОМУ СЕРЕДОВИЩІ

1.1 Основні методи локалізації

Основою функціонування всіх сучасних систем внутрішнього позиціонування виступають алгоритми локалізації, які прийнято класифікувати на дві ключові групи:

- методи, що базуються на вимірюванні дальності сигналу (range-based) [9];
- підходи, які не використовують прямих вимірів відстані (range-free) [10].

Range-based підходи ґрунтуються на визначенні віддаленості між радіоміткою та опорними вузлами (базовими станціями), після чого ці значення застосовуються для обчислення координат об'єкта. Головною перевагою таких алгоритмів є підвищена точність локалізації у порівнянні з бездальномірними методами. До цієї категорії відносять трилатерація та триангуляція.

Натомість у range-free підходах положення об'єкта не визначається через фізичні вимірювання відстані. Замість цього використовується факт доступності сигналу: аналізується, чи перебуває вузол у зоні покриття конкретної базової станції [10]. Одним із найпоширеніших прикладів такого підходу є метод радіочастотних відбитків (fingerprinting). Як правило, реалізації таких систем є дешевшими у впровадженні та обслуговуванні, ніж рішення на основі range-based технологій, що особливо актуально для масштабних IoT-інфраструктур.

Детальніше розглянемо основні підметоди, що згадувалися вище.

Трилатерація є одним із найуживаніших способів просторового визначення координат у бездротових мережах. Її принцип полягає у використанні відстаней від невідомої точки до щонайменше трьох опорних вузлів. Схематично цей підхід ілюструється на рисунку 1.1, де координати $(x_1; y_1)$, $(x_2; y_2)$ та $(x_3; y_3)$ відповідають положенням трьох базових станцій, а d_1, d_2, d_3 позначають відстані від них до об'єкта. Шукане місцеположення маячка задається координатами $(x_M; y_M)$ [11].

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		7

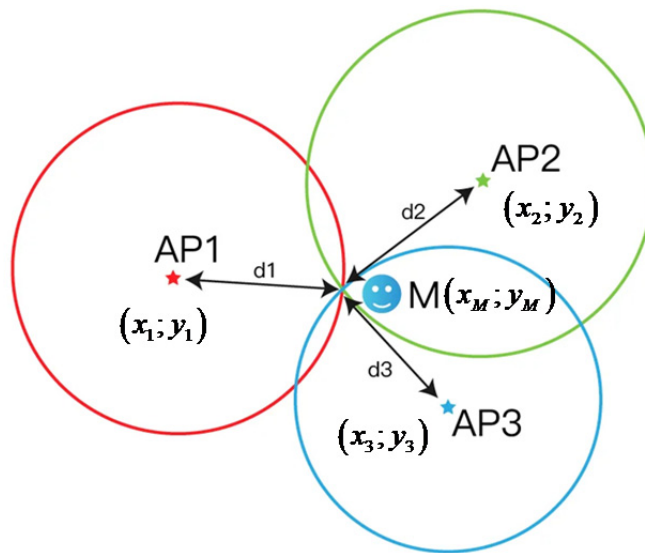


Рисунок 1.1 – Метод трилатерації

Розв’язання координат виконується шляхом математичного вирішення системи рівнянь (1.1), яка описує перетин кіл радіусів, що дорівнюють виміряним відстаням:

$$\begin{cases} (x_M - x_1)^2 + (y_M - y_1)^2 = d_1^2, \\ (x_M - x_2)^2 + (y_M - y_2)^2 = d_2^2, \\ (x_M - x_3)^2 + (y_M - y_3)^2 = d_3^2. \end{cases} \quad (1.1)$$

Триангуляція, на відміну від попереднього методу, базується не на відстані, а на вимірюванні кутів приходу сигналу. У цьому випадку визначається напрямок сигналу від кожного опорного вузла до цільового пристрою, а остаточне положення об’єкта знаходиться у точці перетину відповідних променів, що виходять із базових станцій у напрямку маячка. Такий підхід реалізується шляхом аналізу кутових характеристик сигналу, що надходить від опорних вузлів [12].

Для визначення координат цільового пристрою у практичних реалізаціях достатньо двох опорних вузлів (рис. 1.2).

Відомими є кути θ_1 та θ_2 , а також координати $(x_1; y_1), (x_2; y_2)$, на основі яких розраховуються координати невідомої точки $(x_3; y_3)$:

$$\begin{aligned} \tan(\theta_1) &= \frac{(y_3 - y_1)}{(x_3 - x_1)}, \\ \tan(\theta_2) &= \frac{(y_3 - y_2)}{(x_3 - x_2)}. \end{aligned} \quad (1.2)$$

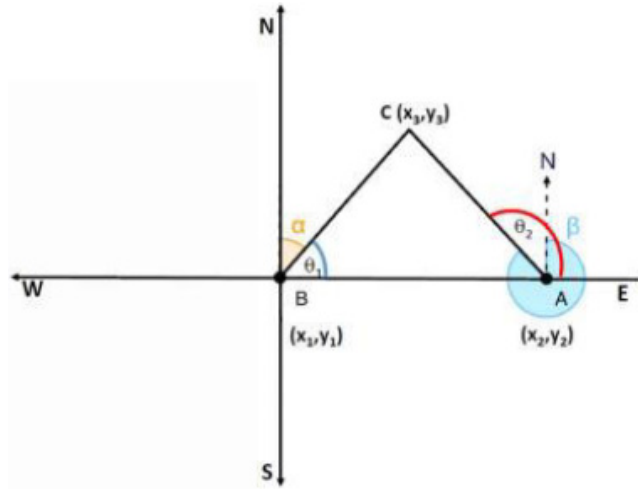


Рисунок 1.2 – Метод триангуляції

Далі, використовуючи співвідношення (1.2), можна отримати вирази для обчислення x_3 і y_3 у явному вигляді:

$$x_3 = \frac{y_1 + y_2 + x_2 \tan(\theta_2) - x_1 \tan(\theta_1)}{\tan(\theta_2) - \tan(\theta_1)}, \quad (1.3)$$

$$y_3 = \frac{y_1 \tan(\theta_2) - y_2 \tan(\theta_1) + (x_2 - x_1) \tan(\theta_2) \tan(\theta_1)}{\tan(\theta_2) - \tan(\theta_1)}. \quad (1.4)$$

Метод радіовідбитків (fingerprinting) базується на аналізі рівня прийнятого сигналу RSSI, який відображає потужність радіосигналу в конкретній точці простору. Його основна ідея полягає у формуванні детальної радіокарти приміщення, що містить унікальні характеристики сигналу для різних зон.

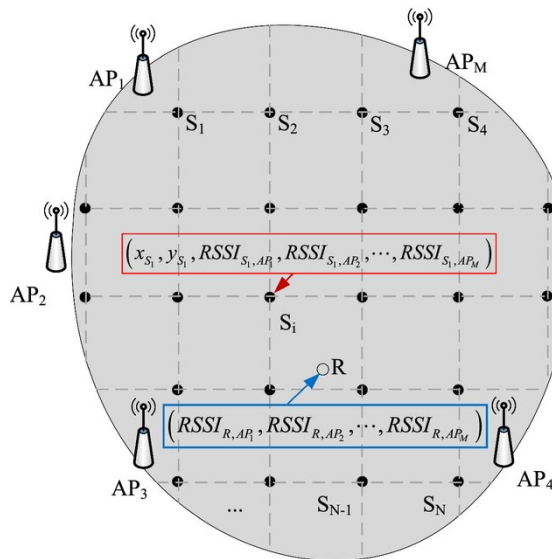


Рисунок 1.3 – RSSI fingerprinting localization

Зм.	Арк.	№ докум.	Підп.	Дата

На першому етапі простір розділяється на невеликі сегменти або координатні області, у кожній з яких фіксуються значення RSSI від різних радіомаяків. Отримані дані зберігаються у базі даних, формуючи так званий «радіопортрет» середовища. У сучасних реалізаціях 2026 року такі бази часто доповнюються методами машинного навчання для підвищення точності.

На другому етапі під час роботи системи поточні виміряні значення RSSI порівнюються з уже наявними записами у базі даних, після чого визначається найбільш ймовірне положення пристрою у межах приміщення (рис. 1.3).

1.2 Параметри локалізації в системах IPS

Необхідно проаналізувати ключові параметри, що застосовуються у сучасних системах внутрішнього позиціонування. До таких характеристик відносять RSSI, TOA, TDOA, AOA, POA, а також CSI. Узагальнена класифікація цих параметрів наведена на рисунку 1.4.

RSSI (Received Signal Strength Indicator) є відносною мірою потужності прийнятого радіосигналу. Цей показник широко використовується для приблизної оцінки відстані між об'єктом і базовою станцією, оскільки не потребує складних математичних обчислень або дорогого обладнання. Відстань між вузлами може бути визначена через втрати енергії сигналу під час його поширення. Для грубої оцінки достатньо взаємодії двох вузлів [13], однак для точного визначення координат необхідно застосовувати метод трилатерації, що передбачає використання щонайменше трьох базових станцій і одного цільового маяка.

Варто зазначити, що виробники по-різному визначають шкалу RSSI. Наприклад, у Wi-Fi модулях на базі чіпсетів Atheros значення можуть змінюватися в межах 0-60 dBm, тоді як у рішень Cisco використовується шкала приблизно 0–100 dBm [12]. Станом на 2026 рік RSSI все ще широко застосовується в IoT та мобільних системах, але дедалі частіше доповнюється більш точними метриками, такими як CSI.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підп.	Дата		

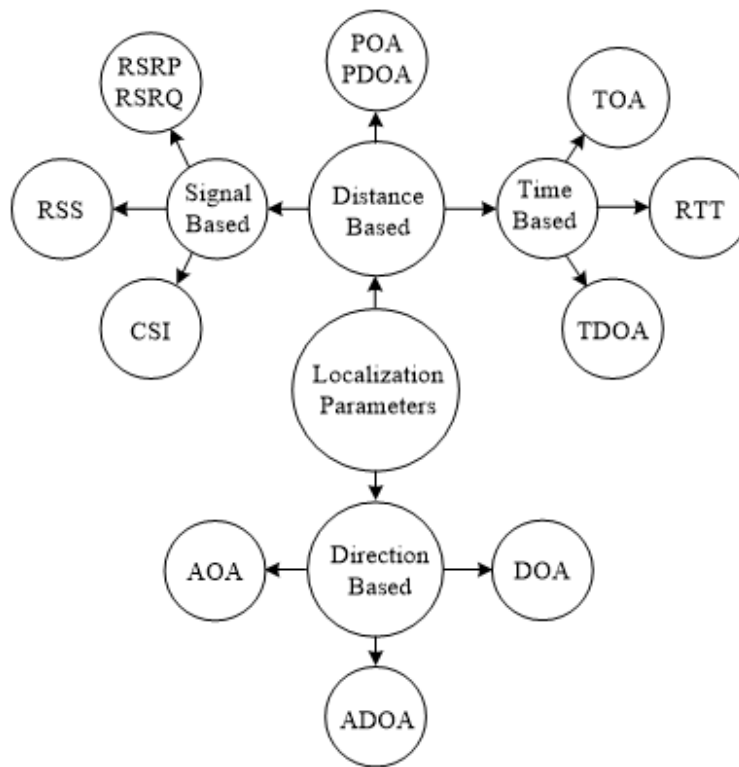


Рисунок 1.4 – Ключові параметри локалізації

TOA (Time of Arrival) визначає момент першого надходження сигналу до приймача. На основі цього параметра можна оцінити відстань між пристроями, обчислюючи час поширення радіосигналу [14,15]. Для коректного визначення координат об'єкта за TOA зазвичай потрібні щонайменше три базові станції, розташовані в зоні прямої видимості, що дозволяє підвищити точність обчислень [16]. У сучасних системах 2026 року TOA часто використовується разом із синхронізацією мережі через високоточні часові протоколи.

TDOA (Time Difference of Arrival) є модифікацією часових методів, де визначається не абсолютний час приходу сигналу, а різниця часу його надходження до кількох базових станцій. Якщо вузли мережі синхронізовані, ця різниця дозволяє обчислити відстані до кожної станції [17]. Базові станції приймають пакети від маячка, після чого координати визначаються через аналіз різниці часу прибуття сигналів у різні точки мережі [18]. У сучасних реалізаціях TDOA активно застосовується в системах високоточного трекінгу в реальному часі, зокрема в промисловому IoT.

AOA (Angle of Arrival) базується на визначенні напрямку, з якого надходить радіосигнал. Для цього використовуються спеціальні антени: обертові системи,

фазовані антенні решітки або багатоспрямовані антенні масиви. Після визначення напрямків від кількох базових станцій координати джерела сигналу знаходяться як точка перетину цих променів. Зі збільшенням кількості опорних вузлів точність позиціонування зростає [15]. Однак висока вартість обладнання та складність реалізації залишаються основними обмеженнями цього методу [19], хоча у 2026 році AOA поступово інтегрується в системи 5G/6G інфраструктури.

РОА (Phase of Arrival) пов'язаний із фазовими характеристиками сигналу та враховує затримку поширення через довжину хвилі та швидкість світла [20]. У таких підходах використовується фаза або різниця фаз несучої хвилі для оцінювання відстані між передавачем і приймачем. Існують різні методи фазового аналізу, які дозволяють підвищити точність вимірювання відстані навіть у складних умовах багатопробеневого поширення сигналу [21].

З розвитком технологій бездротового зв'язку, зокрема LTE та сучасних Wi-Fi систем, широкого поширення набув метод OFDM (Orthogonal Frequency Division Multiplexing). Ця технологія розподіляє дані між великою кількістю піднесучих у межах однієї частотної смуги. Відповідно до стандарту IEEE 802.11, приймальний пристрій повинен оцінювати CSI для коректного декодування сигналу.

CSI (Channel State Information) описує частотну характеристику каналу для кожної піднесучої в OFDM-системі. На відміну від RSSI, який є узагальненим показником, CSI надає значно детальнішу інформацію про канал зв'язку – обсяг даних може бути у десятки разів більшим [22]. Завдяки цьому CSI демонструє вищу стабільність та забезпечує значно точніше позиціонування порівняно з традиційними RSSI-методами [15,22]. У сучасних системах 2026 року CSI активно інтегрується з алгоритмами машинного навчання для підвищення точності внутрішньої локалізації.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підп.	Дата		

1.3 Технології зв'язку в системах IPS

Серед різноманітних технологій, які застосовуються для побудови систем внутрішнього позиціонування (рис. 1.5) розглядаються ключові. До них належать Wi-Fi, BLE, ZigBee, RFID, UWB, LoRa, Sigfox та NFC, кожна з яких має власні особливості, переваги й обмеження залежно від сценарію використання та вимог до точності.

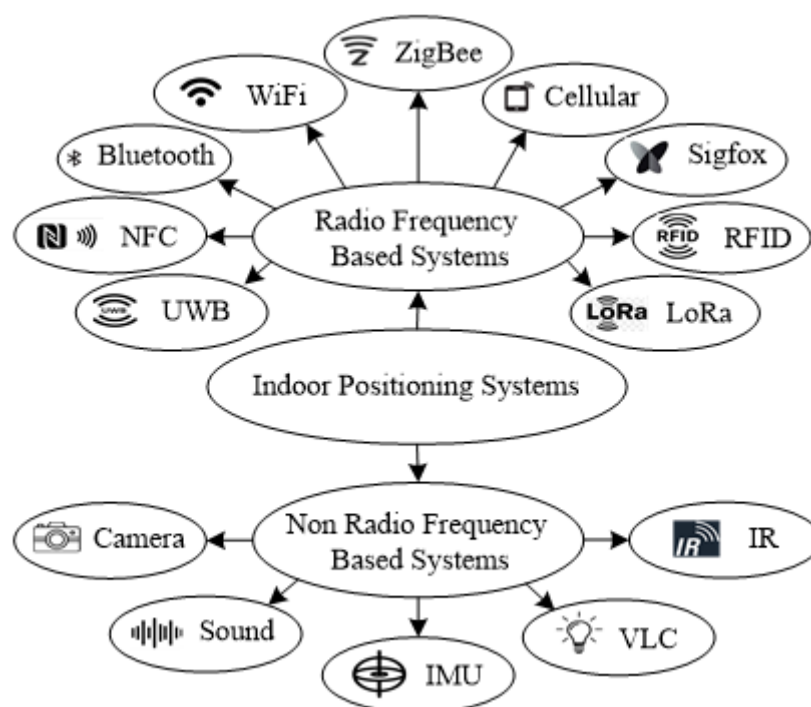


Рисунок 1.5 – Технології зв'язку для IPS

Wi-Fi є однією з найпоширеніших технологій бездротового широкосмугового зв'язку, яка активно використовується не лише для доступу до Інтернету, але й у системах міжмашинної взаємодії (M2M) [23]. Сучасні Wi-Fi пристрої працюють переважно в діапазонах 2,4 ГГц і 5 ГГц, причому останній забезпечує менший рівень завад і стабільніше з'єднання завдяки меншому перевантаженню каналу та вищій пропускну здатності [24].

У задачах локалізації Wi-Fi використовується як базова інфраструктура для різних методів визначення координат, зокрема AOA (триангуляційний підхід) [25], трилатерації [26], RSSI-фінгерпринтингу [27] та сучасніших методів на основі CSI [28]. Перевагами Wi-Fi є його повсюдне розповсюдження, відносно

низька вартість впровадження, відсутність потреби у спеціалізованому обладнанні та енергоефективність у порівнянні з деякими альтернативами [29,30]. Водночас середня точність позиціонування зазвичай обмежується рівнем 2–3 метрів, що є недостатнім для високоточних застосувань. Крім того, на якість визначення координат впливають умови середовища: планування приміщення, наявність перешкод, рух великої кількості людей та навіть зміни радіообстановки, які стали ще більш динамічними у 2026 році через щільність бездротових мереж [31].

BLE (Bluetooth Low Energy), що базується на стандарті IEEE 802.15, сьогодні підтримується практично всіма сучасними мобільними пристроями. Починаючи з версії Bluetooth 4.0 (2010 рік) [32], технологія працює в діапазоні 2,4–2,4835 ГГц. BLE спеціально оптимізований для короткодистанційної передачі даних з мінімальним енергоспоживанням, що робить його особливо придатним для IoT та внутрішнього позиціонування.

Завдяки обмеженому радіусу дії (приблизно до 10 метрів у стандартних сценаріях) BLE дозволяє досягати вищої точності порівняно з Wi-Fi [33]. У практичних реалізаціях похибка може становити близько 1 метра [34]. При цьому енергоспоживання залишається дуже низьким, що є критичним для автономних сенсорних пристроїв [35-37]. За рахунок налаштування потужності передавача робочий радіус може збільшуватися до приблизно 100 метрів, що розширює можливості побудови гнучких систем локалізації.

У BLE-системах найчастіше використовуються методи трилатерації та фінгерпринтингу. Значний вплив на розвиток технології мали стандарти iBeacon (Apple, 2013) та Eddystone (Google, 2015), які уніфікували підходи до передачі сигналів маячків і сприяли масовому впровадженню Bluetooth-базованих рішень у комерційних системах позиціонування [35,38]. У 2026 році BLE залишається однією з найпопулярніших технологій для indoor-навігації завдяки балансу між точністю, вартістю та енергоефективністю.

ZigBee базується на стандарті IEEE 802.15.4 і орієнтований на низькошвидкісні, але енергоефективні мережі з невеликим радіусом дії. Основна робоча частота – 2,4 ГГц. Архітектура ZigBee передбачає три типи вузлів:

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		14

координатор, маршрутизатор та кінцевий пристрій, причому останній є найдешевшим у виробництві.

Перевагою ZigBee є можливість розширення покриття завдяки mesh-топології, де повідомлення передаються через проміжні вузли, що збільшує загальний радіус мережі [39]. Однак це супроводжується вищим енергоспоживанням у порівнянні з BLE [39], а також складнішою архітектурою підтримки мережі. Додатковим обмеженням є необхідність постійної активності маршрутизуючих вузлів, тоді як кінцеві пристрої не виконують функцій ретрансляції [39]. У сучасних IoT-системах 2026 року ZigBee поступово витісняється більш універсальними протоколами, хоча все ще використовується в промислових рішеннях.

RFID (Radio Frequency Identification) є технологією безконтактної ідентифікації, яка працює шляхом передавання даних між міткою та зчитувачем за допомогою електромагнітного поля. Типова система RFID складається зі зчитувача, міток і обчислювального модуля [40].

Існують активні та пасивні RFID-мітки [41,42]. Активні мітки мають власне живлення, що забезпечує більшу дальність роботи, але призводить до підвищеної вартості та енергоспоживання. Пасивні мітки не містять джерела живлення і активуються сигналом зчитувача, тому вони дешевші, але обмежені за дальністю та придатні переважно для короткодистанційної ідентифікації.

Основним недоліком RFID у задачах позиціонування є низька ефективність у режимі реального часу при використанні пасивних міток, а також висока енергомісткість активних рішень. Крім того, поширення сигналу в приміщеннях ускладнюється багатопроменевістю, відбиттями та дифракційними ефектами, що знижує точність локалізації [43].

UWB (Ultra-Wideband) є перспективною технологією бездротового зв'язку, яка забезпечує надшироку смугу частот (приблизно 3,1-10,6 ГГц) і високу точність передачі даних на коротких дистанціях. Завдяки цьому UWB дозволяє досягати дуже високої точності позиціонування, що робить його одним із найкращих рішень для indoor-навігації [44].

					БР.КІ-19.00.00.000 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підп.	Дата		

Широка смуга пропускання забезпечує високу стійкість до перешкод і багатопроменевого поширення сигналу, що особливо важливо в складних приміщеннях. Додатковою перевагою є низька потужність випромінювання, що зменшує вплив на навколишнє середовище та підвищує безпечність для користувачів [45].

Основним недоліком UWB залишається обмежена поширеність інфраструктури та висока вартість обладнання. Крім того, широкий частотний діапазон може створювати потенційні конфлікти з іншими радіосистемами, що працюють у суміжних спектрах [46]. У 2026 році UWB активно інтегрується у смартфони та системи точного трекінгу, але все ще поступається Wi-Fi та BLE за масовістю впровадження.

1.4 Мікроконтролери для систем автоматизації «розумного будинку»

У системах Інтернету речей широко використовують мікроконтролери, а створення і тестування подібних проєктів значно спрощується завдяки спеціалізованим відлагоджувальним платам.

Мікроконтролери прямо або опосередковано формують сучасний побут. Вони інтегровані практично в усі типи вбудованої електроніки: від побутових приладів (пральні машини, кухонне обладнання) до інфраструктурних систем (паливні колонки, паркомати, ліфтові механізми тощо). Саме завдяки ним техніка отримує інтелектуальні функції та можливість автоматизації [47].

Станом на 2026 рік найбільш поширеними архітектурними напрямками у сфері мікроконтролерів залишаються сімейства PIC, AVR, ARM-сумісні рішення (зокрема STM32) та серія ESP від Espressif [48]. Водночас ринок активно розвивається у бік систем-на-кристалі (SoC) із вбудованими бездротовими інтерфейсами та підтримкою edge-обчислень.

Мікроконтролери PIC (рис. 1.6) класифікуються за кількома ключовими характеристиками: розрядністю, периферійними інтерфейсами та типом корпусу.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підп.	Дата		

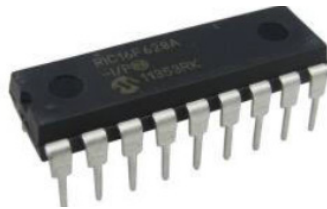


Рисунок 1.6 – Мікроконтролер PIC

Серед 8-бітних представників виділяють групи baseline, mid-range, enhanced mid-range та PIC18. Деталізовану класифікацію наведено на рисунку 1.7.

	Baseline Architecture	Mid-Range Architecture	Enhanced Mid-Range Architecture	PIC18 Architecture
Pin Count	6-40	8-64	8-64	18-100
Interrupts	No	Single interrupt capability	Single interrupt capability with hardware context save	Multiple interrupt capability with hardware context save
Performance	5 MIPS	5 MIPS	8 MIPS	Up to 16 MIPS
Instructions	33, 12-bit	35, 14-bit	49, 14-bit	83, 16-bit
Program Memory	Up to 3 KB	Up to 14 KB	Up to 28 KB	Up to 128 KB
Data Memory	Up to 138 Bytes	Up to 368 Bytes	Up to 1,5 KB	Up to 4 KB
Hardware Stack	2 level	8 level	16 level	32 level
Features	- Comparator 8-bit ADC - Data Memory - Internal Oscillator	In addition to Baseline: - SPI/PC™ - UART - PWMs - LCD 10-bit ADC - Op Amp	In addition to Mid-Range: - Multiple Communication Peripherals - Linear Programming Space - PWMs with Independent Time Base	In addition to Enhanced Mid-Range: 8x8 Hardware Multiplier - CAN - CTMU - USB - Ethernet 12-bit ADC
Highlights	Lowest cost in the smallest form factor	Optimal cost to performance ratio	Cost effective with more performance and memory	High performance, optimized for C programming, advanced peripherals

Рисунок 1.7 – Характеристики мікроконтролерів PIC

Для програмування таких пристроїв традиційно застосовують мови асемблера та C. Серед популярних середовищ розробки варто згадати MPLAB X, MicroC, а також сучасні версії MPLAB з інтеграцією хмарних сервісів. Зазвичай ці мікроконтролери використовують у нескладних пристроях – наприклад, генераторах сигналів, таймерах або системах індикації.

Мікроконтролери AVR (рис. 1.8), які раніше асоціювалися з компанією Atmel (нині входить до Microchip), залишаються актуальними завдяки широкому застосуванню в платформах Arduino.



Рисунок 1.8 – Мікроконтролер AVR

Вони поділяються на лінійки tinyAVR, megaAVR та AVR Dx (оновлення класичних серій). Основні характеристики представлені на рисунку 1.9.

Device Family	Key Features	Key Parameters	Technologies	Summary Benefit
32-bit AVR UC3	- Connectivity - Arithmetic performance - High data throughput - Low power consumption - Software Library	16 - 512KB Flash 48 - 144 pins - Up to 66 MHz 1.5 MIPS/MHz	- picoPower® - SleepWalking - FlashVault - DMA - Event System - Self Programming	The world's most efficient 32-bit microcontroller
8/16-bit AVR XMEGA Microcontroller	- High-precision analog - Real-time performance - Atmel picoPower® technology - High integration - AVR Software Library - Atmel QTouch® Sensing - USB connectivity - XMEGA Custom Logic (XCL)	16 - 384KB Flash 44 - 100 pins - Up to 32 MHz 1.0 MIPS/MHz	- picoPower - SleepWalking - DMA - Event System - EEPROM - Self Programming	Xtreme performance 8-bit
8-bit megaAVR Microcontroller	- Broad family - picoPower technology - High integration - Analog functions - Rapid development	4 - 256KB Flash 28 - 100-pins - Up to 20 MHz 1.0 MIPS/MHz	- QTouch - picoPower - SleepWalking - EEPROM - Self Programming	More peripherals and options
8-bit tinyAVR	- Small - Capacitive Touch - Fast and code efficient - High integration 0.7V operation	0.5 - 8KB Flash 6-32 pins - Up to 20MHz 1.0 MIPS/MHz	- QTouch - EEPROM 0.7V operation	Small and powerful

Рисунок 1.9 – Характеристики мікроконтролерів AVR

Завдяки популярності Arduino, початківці можуть швидко освоїти роботу з цими чіпами. Програмування можливе мовами C/C++, а також через спрощені візуальні середовища. Найбільш уживаними IDE у 2026 році є Arduino IDE 2.x, Microchip Studio та PlatformIO. Для прошивки використовують як зовнішні програматори, так і вбудовані засоби на самих платах.

Мікроконтролери STM поділяються на серії STM8 та STM32. Перша включає 8-бітні рішення, друга – 32-бітні. STM8 можна розглядати як

альтернативу AVR у бюджетних застосуваннях. Вони представлені серіями STM8L (енергоєфективні), STM8S (універсальні) та STM8A (для автомобільної електроніки). Їх перевагами є низьке енергоспоживання та доступна ціна.

STM32 є значно потужнішою платформою з широким спектром можливостей (рис. 1.10). Вони базуються на архітектурі ARM, яка ліцензується різними виробниками. Найбільшого поширення набули рішення компанії STMicroelectronics. У 2026 році STM32 активно застосовуються в IoT, робототехніці та промисловій автоматизації. Для них доступні різні операційні системи реального часу, зокрема ARM Mbed OS, FreeRTOS та Zephyr [48]. Програмування здійснюється мовами C/C++, а також із використанням сучасних фреймворків і навіть MicroPython.



Рисунок 1.10 – Мікроконтролер STM

Починаючи з середини 2010-х років, значну популярність здобули мікросхеми ESP8266 компанії Espressif. Вони стали одним із перших доступних рішень із вбудованим Wi-Fi, що значно спростило створення IoT-пристроїв. ESP8266 містить 32-бітний процесор Tensilica L106 із тактовою частотою близько 80 МГц, оперативну пам'ять та набір стандартних інтерфейсів (GPIO, SPI, I2C, UART, ADC). Головною перевагою є інтегрований Wi-Fi модуль, хоча підтримка Bluetooth відсутня [50]. Ці мікросхеми можна використовувати як автономні контролери або як мережеві модулі, що взаємодіють з іншими пристроями через UART за допомогою AT-команд. Найчастіше застосовують готові модулі (наприклад ESP-01), які містять усі необхідні компоненти для роботи (рис. 1.11).

ESP32 є подальшим розвитком цієї лінійки. У сучасних версіях (включаючи ESP32-S3, ESP32-C6) використовуються як Tensilica, так і RISC-V ядра з частотою до 240 МГц.

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		19

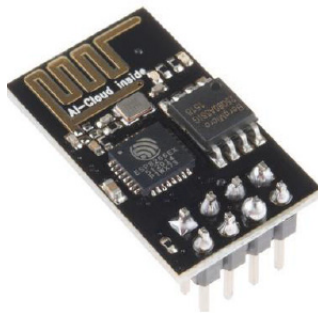


Рисунок 1.11 – Модуль ESP8266-ESP-01

Обсяг оперативної пам'яті може перевищувати 500 КБ, а також підтримуються численні інтерфейси: GPIO, SPI, I2C, I2S, UART, CAN, ADC, DAC. Крім Wi-Fi, ESP32 підтримує Bluetooth (включно з BLE 5.x у новіших моделях), що значно розширює можливості підключення [52].

Важливою перевагою ESP32 є апаратна підтримка безпеки: криптографічні модулі, захищене завантаження та шифрування flash-пам'яті. Це робить платформу придатною для сучасних IoT-рішень із підвищеними вимогами до кібербезпеки. Зовнішній вигляд модуля наведено на рисунку 1.12.



Рисунок 1.12 – ESP32-DEVKITC-S1

Загалом ESP32 перевершує ESP8266 за більшістю технічних характеристик, що підтверджується порівнянням (рис. 1.13).

Відлагоджувальні плати являють собою друковані плати, на яких уже встановлено мікроконтролер разом із необхідними допоміжними компонентами. Крім того, до цієї категорії можна віднести й одноплатні комп'ютери.

Такі плати зазвичай включають інтерфейси для підключення до ПК, роз'єми для модулів розширення, зони для прототипування власних схем.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підп.	Дата		

Specifications	ESP8266	ESP32
MCU	Xtensa® Single-Core 32-bit L106	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Yes, HT20	Yes, HT40
Bluetooth	None	Bluetooth 4.2 and below
Typical Frequency	80 MHz	160 MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash , up to 16 MBytes	SPI Flash , up to 16 MBytes
GPIO	17	36
Hardware / Software PWM	None / 8 Channels	1 / 16 Channels
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	None	1
Ethernet MAC Interface	None	1
Touch Sensor	None	Yes
Temperature Sensor	None	Yes
Working Temperature	- 40°C ~ 125°C	- 40°C ~ 125°C

Рисунок 1.13 – Порівняння параметрів ESP32 та ESP8266

Вони значно прискорюють процес створення прототипів, тестування функціональних вузлів та перевірки інженерних рішень [52].

На рисунку 1.14 показано приклад плати на базі AVR, а в табл. 1.1 наведено порівняння різних моделей відлагоджувальних платформ.

Одноплатний комп'ютер – це повноцінна обчислювальна система, реалізована на одній платі. Вона містить процесор, оперативну пам'ять, інтерфейси введення/виведення та інші компоненти, необхідні для роботи.

Існують різні форм-фактори таких пристроїв. Розробницькі варіанти забезпечують зручний доступ до контактів без необхідності пайки, що спрощує створення прототипів.

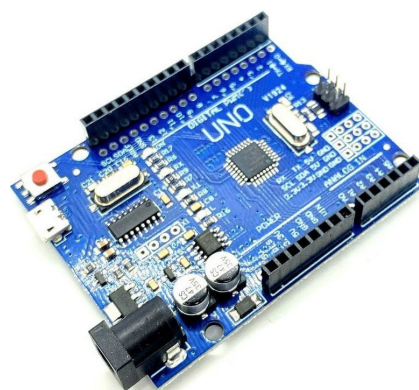


Рисунок 1.14 – Плата відлагодження на мікроконтролері AVR

Таблиця 1.1 – Порівняння плат на основі мікроконтролерів

Характеристика	Модель		
	Arduino UNO	Particle Electron	ESP8266-01
GPIO, Analog/Digital	6/14	12/30	1/2
Логічний рівень, В	5	3.3	3.3
Мікроконтролер	ATMega328P	STM32F205 ARM Cortex M3	32-bit Tensilica L106
Частота, MHz	16	120	80
Пам'ять	32 kB flash, 1kB EEPROM	1 MB flash, 128kB RAM	1 MB flash, 76 8kB RAM
Мережеві інтерфейси	-	2G/3G	Wi-Fi, Bluetooth
Ціна, \$	20	39	10

Промислові версії орієнтовані на інтеграцію у кінцеві продукти та зазвичай мають компактні розміри (наприклад, Raspberry Pi 4, рис. 1.15).

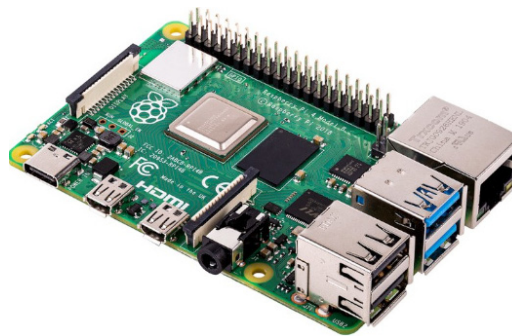


Рисунок 1.15 – Raspberry Pi 4

У табл. 1.2 наведено порівняння параметрів кількох одноплатних комп'ютерів.

Таблиця 1.2 – Порівняння параметрів одноплатних комп'ютерів

Характеристика	Модель		
	Arduino UNO	Particle Electron	ESP8266-01
GPIO, Analog/Digital	11/40	7/65	-/12
Логічний рівень, В	3.3	5	1.8
Мікропроцесор	ARM Cortex A72	AM335X ARM Cortex A8	ARM Cortex A53
Частота, GHz	1.5	1	1.2
Пам'ять	1-4 GB flash, 1GB RAM	4 GB flash, 512MB RAM	8 GB flash, 1GB RAM
Мережеві інтерфейси	Wi-Fi, Ethernet, Bluetooth	Wi-Fi, Ethernet, Bluetooth	Wi-Fi, Bluetooth, GPS
Ціна, \$	35	55	75

До основних переваг таких рішень належать висока продуктивність, наявність численних GPIO та мережевих інтерфейсів (Ethernet, Wi-Fi, Bluetooth). Водночас їхніми недоліками є більше енергоспоживання та вища вартість порівняно з класичними мікроконтролерними платформами.

1.5 Постановка завдання

Функціональні вимоги та задачі, які має реалізовувати IoT-система моніторингу та аналізу даних щодо присутності людей у приміщенні, включають наступний набір можливостей:

- забезпечення зчитування унікальних ідентифікаційних кодів або міток;
- формування повідомлення або індикатора стану у випадку успішного або невдалого виявлення об'єкта;
- передача зібраної інформації з мікроконтролера до серверної частини, де розгорнуто базу даних;
- організація стабільного обміну даними між застосунком та базою даних;
- забезпечення коректного та точного формування результатів у відповідь на запити користувача;
- дотримання вимог до швидкодії та мінімізації затримок під час обробки інформації;
- реалізація механізмів захисту та контролю доступу до даних про перебування осіб у приміщенні;
- забезпечення інтуїтивно зрозумілого та зручного інтерфейсу користувача;
- відображення результатів у вигляді графіків, діаграм та аналітичних візуалізацій;
- можливість розгортання системи на серверних платформах, а також створення та відновлення резервних копій бази даних.

У загальному вигляді, у межах проєктованого рішення мають бути реалізовані такі властивості та характеристики:

					БР.КІ-19.00.00.000 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підп.	Дата		

- висока надійність функціонування апаратних компонентів системи;
- достатній рівень точності визначення та фіксації присутності користувача;
- ефективна продуктивність програмного забезпечення при обробці потокових даних;
- підтримка одночасного доступу декількох користувачів до бази даних без втрати цілісності інформації;
- реалізація розмежування прав доступу відповідно до ролей користувачів;
- оптимізація часу обробки запитів і раціональне використання обчислювальних ресурсів;
- надання зручного, зрозумілого та функціонального інтерфейсу для взаємодії з радіомітками та пов'язаною апаратно-програмною системою.

Таким чином, метою даної роботи є створення системи, що забезпечує відстеження переміщень осіб у межах приміщення з можливістю накопичення відповідних даних.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати технології визначення координат у закритому середовищі;
- обрати параметри, що використовуються для обчислення положення людини в приміщенні;
- організувати збереження історії переміщень користувачів із урахуванням вимог до безпеки даних;
- розробити та протестувати прототип системи.

2 РОЗРОБЛЕННЯ СТРУКТУРИ МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ

2.1 Архітектура IoT системи

Технології Інтернет речей відкривають широкі можливості для реалізації інноваційних задумів у різних галузях. Сьогодні значна кількість організацій інтегрує IoT-рішення у власні операційні процеси, однак у реальних умовах впровадження часто супроводжується труднощами: складно об'єднати велику кількість пристроїв у єдину систему та забезпечити її стабільне функціонування. Саме тому критично важливим є дотримання чітко визначеної архітектурної моделі під час розробки таких систем.

Як правило, структура IoT-систем включає чотири основні рівні (рис. 2.1) [53].

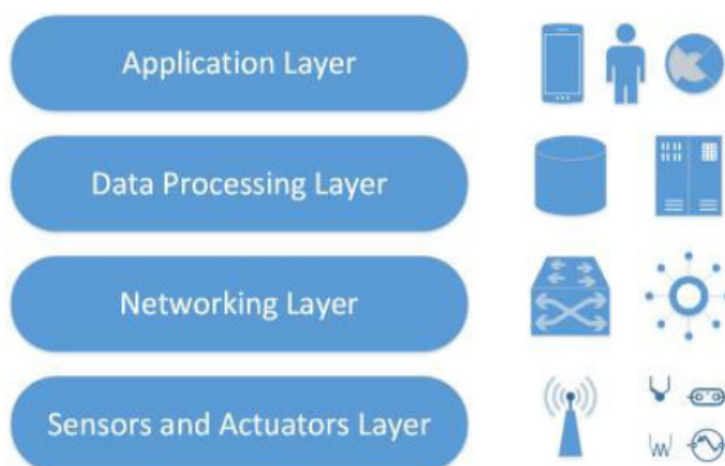


Рисунок 2.1 – Архітектура IoT систем

Нижній рівень – апаратний (hardware layer) – охоплює фізичні пристрої: сенсори, давачі, а також мікроконтролери, які здійснюють збір первинних даних.

Після отримання та базової обробки інформації виникає потреба її передачі до серверної частини. Для цього застосовуються різні засоби комунікації, зокрема LoRaWAN, NB-IoT, 4G/5G, Wi-Fi та інші сучасні стандарти бездротового зв'язку. Цей етап належить до мережевого рівня.

Наступний щабель – рівень обробки – відповідає за виконання серверної логіки. Тут відбувається аналіз даних, взаємодія з пристроями, а також обмін

					БР.КІ-19.00.00.000 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підп.	Дата		

інформацією з прикладними сервісами. Серверна інфраструктура може бути як локальною, так і реалізованою у вигляді хмарного рішення, що надається за моделлю SaaS.

Використання хмарних технологій дозволяє підприємствам ефективно управляти інформацією через спеціалізовані платформи, забезпечуючи гнучкість і масштабованість програмного забезпечення. Станом на 2026 рік, хмарні сервіси також активно інтегрують edge computing, що дає змогу обробляти частину даних ближче до джерела їх генерації, зменшуючи затримки та навантаження на центральні сервери. Передача інформації між пристроями, застосунками та хмарию здійснюється безперервно через інтернет або виділені канали.

Для IoT характерною є генерація значних обсягів даних у реальному часі. Хмарні платформи забезпечують їх накопичення, обробку та аналітику, що дозволяє бізнесу отримувати додаткову цінність від впроваджених технологій. IoT-рішення поєднують фізичні об'єкти, користувачів і бізнес-процеси в єдину екосистему, де хмарні обчислення відіграють ключову роль [54].

Водночас IoT не обмежується лише збором і аналізом інформації. Технологія сприяє цифровій трансформації виробництва, логістики та інших сфер шляхом інтеграції навіть застарілого обладнання в цифрове середовище. Це дозволяє зменшити розрив між IT та OT-підрозділами завдяки єдиному доступу до даних і систем.

Хмарні рішення значно спрощують впровадження IoT, оскільки усувають потребу в розгортанні складної локальної інфраструктури. Організації можуть масштабувати ресурси відповідно до поточних потреб без придбання додаткового обладнання. Такий підхід пришвидшує розробку та оптимізує витрати, адже оплата здійснюється лише за фактично використані ресурси.

Однією з провідних платформ у цій сфері є Google Cloud Platform. Вона пропонує десятки сервісів для керування інфраструктурою, обробки даних і розробки програмного забезпечення (рис. 2.2).

Платформа відповідає сучасним вимогам безпеки та стандартам захисту персональних даних.

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		26

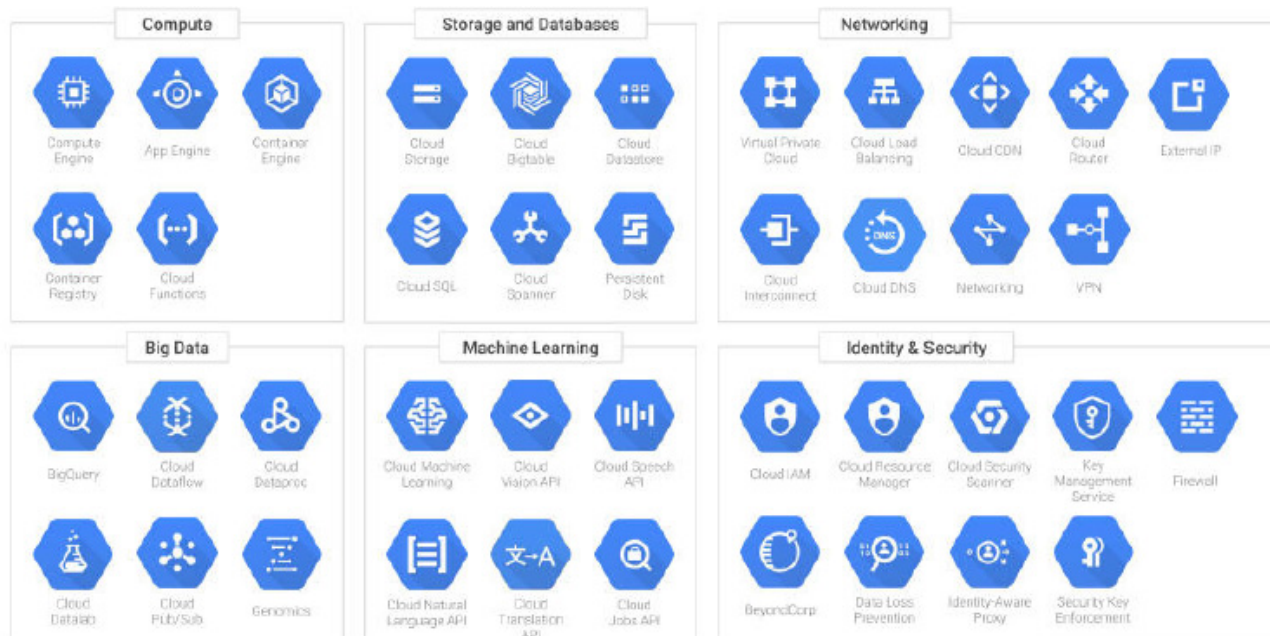


Рисунок 2.2 – Структура сервісів GCP

Інфраструктура сервісів і даних розміщується у глобальних дата-центрах Google. Керування ресурсами здійснюється через веб-консоль, де користувачі можуть запускати віртуальні машини, створювати бази даних й інтегрувати різні сервіси. Також доступні інструменти адміністрування через CLI або застосунки.

Користувачами платформи можуть бути як фізичні особи, так і організації. На 2026 рік нові акаунти зазвичай отримують стартовий кредит (зазвичай близько 300 USD протягом 90 днів), що дозволяє протестувати функціональність сервісу та покрити початкові витрати.

До складу платформи входять різні категорії сервісів:

- інфраструктурні рішення для мереж і масштабування ресурсів;
- інструменти для роботи з даними (обробка, зберігання, аналітика, візуалізація);
- бізнес-сервіси для автоматизації командної роботи;
- засоби контейнеризації та мікросервісної архітектури (зокрема Kubernetes);
- безсерверні обчислення (serverless);
- механізми кібербезпеки;
- сервіси штучного інтелекту та машинного навчання.

Платформа забезпечує доступ до уніфікованих обчислювальних ресурсів без прив'язки до окремих фізичних серверів. Основна частина сервісів функціонує на віртуальних машинах, що дозволяє легко масштабувати систему – від одного вузла до великих дата-центрів із тисячами серверів [55].

На рисунку 2.3 наведено загальну структуру Google Cloud Platform.

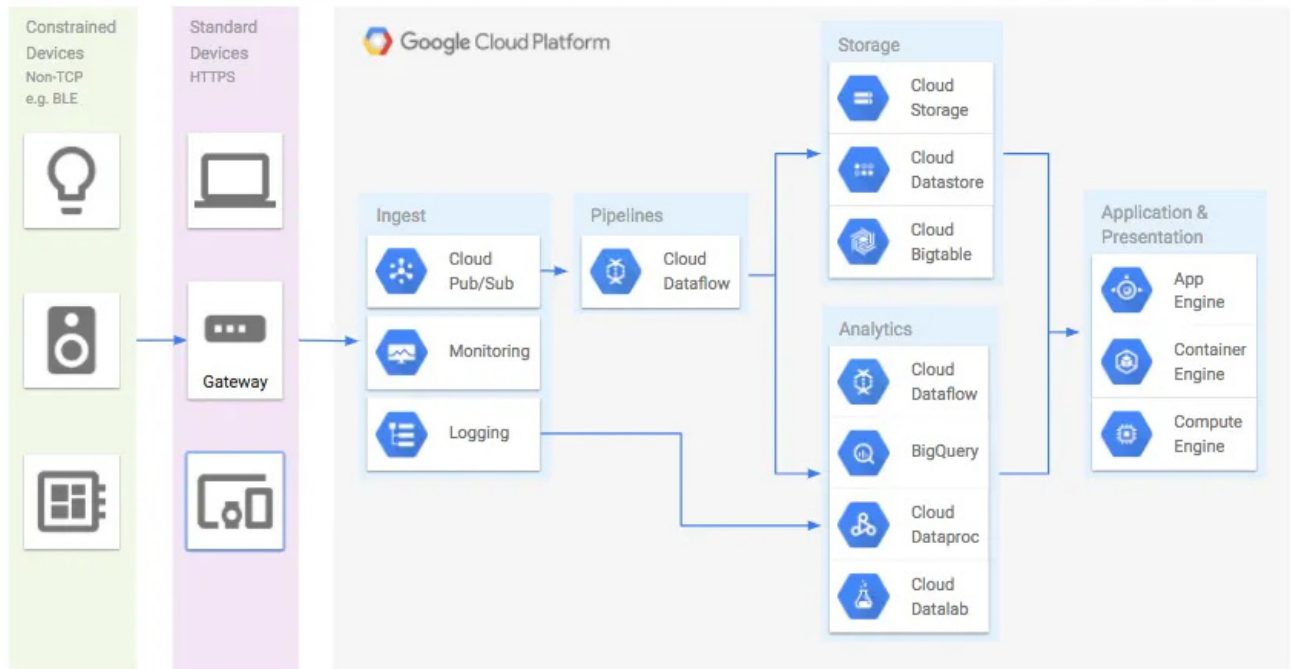


Рисунок 2.3 – Архітектура GCP

Рівень джерел даних (Devices). Зазвичай можна виділити такі два різновиди пристроїв:

- Constrained Devices: малопотужні пристрої (датчики), що працюють через енергоефективні протоколи (наприклад, BLE). Вони потребують Gateway (шлюзу) для виходу в інтернет;
- Standard Devices: розумні пристрої (ноутбуки, смартфони), що підтримують стеки TCP/IP та HTTPS для прямої передачі даних.

Рівень прийому даних (Ingest) :

- Cloud Pub/Sub: центральний хаб для вхідних повідомлень. Він працює як черга, що дозволяє системі витримувати величезні пікові навантаження, "відв'язуючи" пристрої від внутрішньої логіки обробки;
- Stackdriver (Monitoring/Logging): забезпечує спостереження за станом системи та збір системних журналів у реальному часі.

Рівень обробки (Pipelines) – Cloud Dataflow: "серце" обробки. Це сервіс для ETL-процесів (Extract, Transform, Load), який у реальному часі очищує дані, агрегує їх (наприклад, рахує середню температуру за хвилину) та направляє у сховища.

На рівні зберігання та аналітики (Storage & Analytics) відбувається розподіл даних залежно від їх призначення:

- Cloud Bigtable: для «гарячих» даних, де потрібна надзвичайна швидкість запису та зчитування (наприклад, графіки в реальному часі);
- Cloud Storage: «холодне» сховище для сирих даних або бекапів;
- BigQuery: хмарне сховище для аналітичних запитів (Data Warehouse), де можна аналізувати мільярди записів за допомогою SQL;
- Cloud Datalab/Dataproц: інструменти для Machine Learning та складної статистичної обробки.

Рівень представлення (Application & Presentation) – App Engine/GKE/Compute Engine: тут розміщується користувацький інтерфейс (Dashboard) або API. Це може бути веб-сайт або мобільний додаток, де користувач бачить результати роботи всієї системи.

2.2 Протоколи передачі даних в IoT

У середовищі інтернету речей застосовується широкий спектр комунікаційних протоколів, серед яких DDS, XMPP, CoAP, MQTT, KMS, AMQP та інші. Окрім спеціалізованих рішень, часто використовують і класичні мережеві протоколи, зокрема HTTP. Вибір конкретного варіанту визначається вимогами задачі, характеристиками обладнання та умовами експлуатації [56].

Для автономних пристроїв, здатних функціонувати протягом кількох років без заміни джерела живлення, зазвичай обирають протоколи з мінімальним енергоспоживанням. До таких належить MQTT, який оптимізований для передачі невеликих обсягів даних і ефективного використання батареї. Станом на 2026 рік

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		29

MQTT залишається одним із найпоширеніших стандартів у IoT, зокрема завдяки підтримці версії MQTT 5.0, що пропонує покращене управління повідомленнями та гнучкі механізми обробки помилок.

У випадках, коли необхідна двостороння взаємодія між пристроями з мінімальними затримками, застосовують протоколи реального часу. Одним із таких є DDS, який забезпечує прямий обмін повідомленнями між вузлами без використання проміжних серверів. Це робить його придатним для критично важливих систем, наприклад у промисловій автоматизації або транспортних мережах.

Інша група протоколів, до якої належать AMQP, JMS, XMPP та MQTT, функціонує за моделлю з використанням брокера. У цьому випадку передача даних відбувається через проміжний сервер, який відповідає за маршрутизацію, фільтрацію та доставку повідомлень. Такі брокери можуть розміщуватися як у локальних дата-центрах, так і у хмарній інфраструктурі. У 2026 році дедалі частіше застосовуються хмарні IoT-платформи з інтегрованими брокерами повідомлень, що забезпечують масштабованість і високу доступність сервісів.

Серед протоколів брокерного типу MQTT продовжує набирати популярність, оскільки він спеціально створений для IoT-середовища. Його перевагами є невеликі накладні витрати, здатність працювати в умовах нестабільного з'єднання та ефективне використання мережевих ресурсів. Оновлені реалізації MQTT також підтримують покращену безпеку, зокрема завдяки інтеграції з TLS 1.3 та сучасними механізмами автентифікації. Спрощена схема його роботи наведена на рисунку 2.4.

Протокол CoAP є альтернативою HTTP, адаптованою для використання в IoT. Його ключова особливість – компактні заголовки та зменшене навантаження на мережу, що робить його придатним для середовищ із обмеженими ресурсами. Станом на 2026 рік CoAP також активно використовується разом із UDP та підтримує розширення для безпечної передачі даних (DTLS).

					БР.КІ-19.00.00.000 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підп.	Дата		

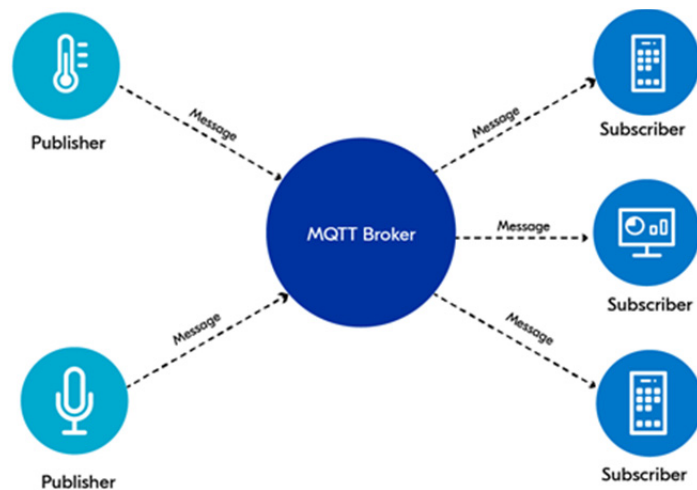


Рисунок 2.4 – Робота протоколу MQTT

Механізм роботи CoAP базується на моделі клієнт-сервер, де клієнт надсилає запити до сервера та отримує відповіді, подібно до HTTP. Водночас CoAP додатково підтримує асинхронну взаємодію та спостереження за ресурсами, що дозволяє ефективніше організовувати обмін даними у розподілених IoT-системах.

2.3 Апаратне та програмне забезпечення системи

Функціональні характеристики та перелік завдань, які має виконувати IoT-рішення для відстеження й обробки відомостей щодо присутності людей у приміщенні, можна сформулювати так:

- підтримка зчитування унікальних ідентифікаторів пристроїв або користувачів (зокрема з урахуванням сучасних стандартів безконтактної ідентифікації, включаючи BLE 5.x);
- генерування повідомлень про результат розпізнавання (успішний або помилковий) із можливістю миттєвого інформування користувача;
- передавання інформації від вбудованого контролера до серверної частини, де розміщено систему зберігання даних з використанням протоколу MQTT;
- організація стабільної взаємодії клієнтського застосунку з серверною інфраструктурою та базою даних;

					БР.КІ-19.00.00.000 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підп.	Дата		

- надання коректних, релевантних і своєчасних відповідей на запити користувачів у режимі реального часу;
- досягнення високої продуктивності та мінімальних затримок під час обробки інформації;
- впровадження механізмів контролю доступу до відомостей про перебування осіб (з урахуванням сучасних підходів до кібербезпеки, наприклад Zero Trust та багатофакторної автентифікації);
- забезпечення інтуїтивно зрозумілого інтерфейсу та зручності взаємодії з програмним продуктом;
- відображення результатів у вигляді інтерактивних графіків, дашбордів і аналітичних панелей;
- підтримка розгортання системи (у тому числі на основі хмарних сервісів).

Вимоги до апаратної частини:

- мікроконтролер із модулями Wi-Fi та підтримкою BLE;
- BLE-маячки стандарту iBeacon або їх сучасні аналоги з покращеною енергоефективністю;
- мережевий маршрутизатор із доступом до Інтернету та підтримкою актуальних стандартів бездротового зв'язку (Wi-Fi 5/6/6E).

Вимоги до клієнтських пристроїв:

- центральний процесор із частотою від 2,0 ГГц і щонайменше двома фізичними або чотирма логічними ядрами;
- оперативна пам'ять обсягом від 8 ГБ (з урахуванням сучасних вимог до багатозадачності);
- накопичувач SSD місткістю не менше 256 ГБ.

Вимоги до серверної інфраструктури:

- процесор із тактовою частотою від 2,5 ГГц і кількістю ядер не менше 8 (або використання хмарних обчислювальних ресурсів із можливістю масштабування);
- оперативна пам'ять від 16 ГБ із можливістю подальшого розширення;
- дисковий простір від 1 ТБ.

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		32

Вимоги до програмного забезпечення:

- системне ПЗ для розробки мікроконтролерів (Arduino Core або сучасні альтернативи, такі як PlatformIO);
- програмне забезпечення клієнтських станцій – актуальні операційні системи (Windows 11 або 10), сучасні середовища виконання (.NET 6/7/8 або новіші), а також допоміжні інструменти;
- серверне програмне забезпечення – сучасні серверні ОС (Windows Server 2022/2025 або Linux-дистрибутиви), а також системи керування базами даних (MS SQL Server 2022/2024, PostgreSQL чи інші актуальні рішення).

Виходячи із вимог до системи та проведеного аналізу в першому розділі, у якості апаратної основи для базових станцій у розроблюваній системі доцільно застосувати мікроконтролерну платформу ESP32 (рис. 1.12), яка залишається актуальним і широко використовуваним рішенням у 2026 році завдяки вбудованій підтримці Wi-Fi та Bluetooth Low Energy. Для цих станцій передбачається створення прошивки, що забезпечує безперервне сканування радіоефіру з метою виявлення Bluetooth-маячків (рис. 2.5). У процесі роботи система визначає рівень сигналу RSSI кожного знайденого пристрою, що дозволяє оцінювати відстань або факт присутності об’єкта в зоні покриття.



Рисунок 2.5 – iBeacon Bluetooth

Отримані показники передаються до серверної інфраструктури за допомогою протоколу MQTT, який у сучасних реалізаціях використовується разом із захищеним з’єднанням (TLS/SSL), що гарантує конфіденційність і цілісність інформації під час передачі. Як хмарне середовище для обробки та

зберігання даних буде використовуватися Google Cloud, що забезпечує масштабованість і високу доступність сервісів.

З'єднання з мережею Інтернет реалізується через бездротовий інтерфейс Wi-Fi, який підтримує сучасні стандарти зв'язку та дозволяє інтегрувати пристрої в існуючу мережеву інфраструктуру без додаткового обладнання.

2.4 Проектування структури системи

На основі сформульованих вимог та вибраного апаратного й програмного забезпечення пропонується архітектура системи, що забезпечує відстеження переміщень осіб у межах приміщення з можливістю накопичення відповідних даних, зображена на рисунку 2.6.

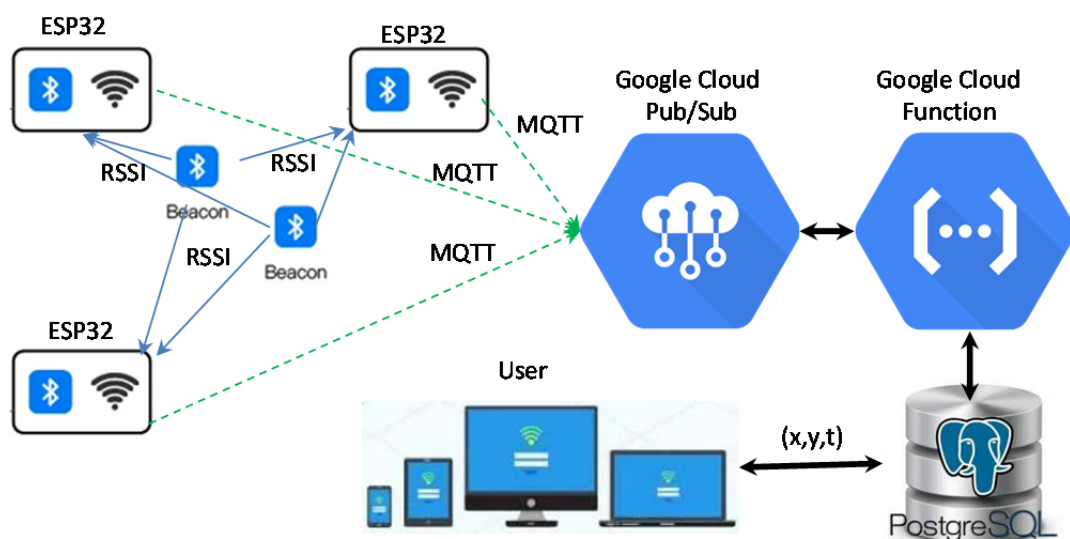


Рисунок 2.6 – Архітектура системи детектування розташування людей у приміщенні

Архітектура системи моніторингу об'єктів у закритих приміщеннях на базі BLE та GCP базується на поєднанні периферійних обчислень (Edge Computing) та хмарної інфраструктури Google Cloud Platform (GCP) для реалізації системи внутрішнього позиціонування (Indoor Positioning System, IPS). Відповідно до архітектури IoT систем (рис. 2.1) запропонована архітектура містить такі рівні.

1. Рівень сприйняття та периферійних вузлів (Perception & Edge Layer). Основою системи є використання технології Bluetooth Low Energy (BLE).

Джерела сигналу (Beacons): автономні активні мітки, що транслюють пакети оголошень (advertising packets).

Анкерні вузли (ESP32): стаціонарні мікроконтролери, оснащені модулями Bluetooth та Wi-Fi. Вони виконують роль сканерів, що реєструють показник сили отриманого сигналу (RSSI) від рухомих маячків. Вибір ESP32 обумовлений підтримкою стека протоколів TCP/IP та низькою вартістю розгортання мережі.

2. Протокол передачі та чергування повідомлень (Transport & Messaging Layer). Для забезпечення надійної передачі даних у реальному часі використовується протокол MQTT. Дані інкапсулюються у JSON-пакети, що містять ID сканера, ID маячка та значення RSSI.

Брокер повідомлень Google Cloud Pub/Sub виступає як асинхронна шина даних, що забезпечує високу пропускну здатність та демпфування пікових навантажень (load buffering) перед етапом обробки.

3. Рівень обробки даних (Cloud Processing Layer). Логіка реалізована за парадигмою Serverless Computing за допомогою Google Cloud Functions.

Алгоритмічна база: функція аналізує масив значень RSSI від кількох анкерних вузлів. Для конвертації фізичних величин у координати застосовуються математичні моделі поширення радіохвиль (наприклад, Log-Distance Path Loss Model) та алгоритми трилатерації для мінімізації похибки, спричиненої багатопроменевим поширенням сигналу. Результат обчислень: вектор стану об'єкта $P = (x, y, t)$, де x, y – просторові координати, а t – часова мітка.

4. Рівень зберігання та прикладного інтерфейсу (Storage & Application Layer). Persistence Layer: розраховані дані зберігаються в об'єктно-реляційній базі даних PostgreSQL. Це дозволяє проводити як оперативний моніторинг, так і ретроспективний аналіз траєкторій (data mining).

Application Layer: кінцеві користувачі (Client Apps) взаємодіють із базою даних через API для візуалізації місцеперебування об'єктів на цифровому плані приміщення.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підп.	Дата		

Відповідно до запропонованої архітектури системи розроблена структурна схема відстеження переміщень осіб у межах приміщення (рис. 2.7)

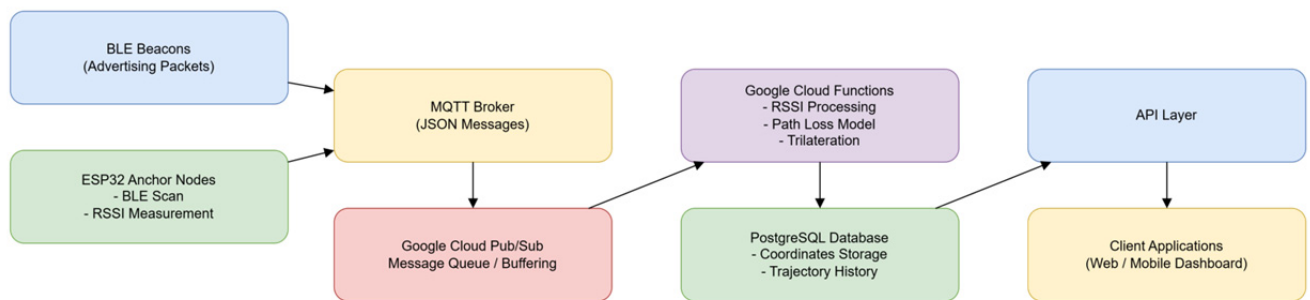


Рисунок 2.7 – Структурна схема системи

Структурна схема системи відображає багаторівневу IoT-архітектуру, яка складається з чотирьох основних логічних рівнів: Edge, Transport, Processing та Application. Передача даних здійснюється послідовно від джерел сигналу до кінцевих користувацьких застосунків.

Edge-рівень відповідає за збір первинних даних у фізичному середовищі. BLE Beacons виступають як рухомі або закріплені маячки, що періодично передають advertising-пакети з ідентифікатором пристрою. ESP32 Anchor Nodes виконують роль стаціонарних сканерів: приймають BLE-сигнали; вимірюють рівень RSSI; формують структуровані пакети даних (ID маячка, ID сканера, RSSI).

Transport-рівень забезпечує передачу даних у хмарну інфраструктуру. Дані від ESP32 передаються через MQTT-протокол у форматі JSON. Повідомлення надходять у MQTT Broker, який виконує роль проміжного вузла доставки. Далі дані потрапляють у Google Cloud Pub/Sub, який: працює як асинхронна черга повідомлень; забезпечує буферизацію навантаження; гарантує масштабовану доставку даних у хмарні сервіси.

На рівні Processing виконується основна аналітика та обчислення. Google Cloud Functions реалізують serverless-обробку даних: аналіз RSSI з кількох анкерних вузлів; застосування моделі поширення сигналу (Log-Distance Path Loss Model); виконання трилатерації для визначення координат об'єкта.

Оброблені дані зберігаються в PostgreSQL, де: фіксуються поточні координати; накопичується історія переміщень; підтримується подальший аналітичний аналіз траєкторій.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підп.	Дата		

Application-рівень забезпечує доступ користувачів до системи. API Layer надає інтерфейс для доступу до даних із бази. Client Applications (веб або мобільні) відображають позиції об'єктів у реальному часі, забезпечують візуалізацію на цифровому плані приміщення, дозволяють здійснювати моніторинг та аналіз переміщень.

Структурна схема демонструє IoT-пайплайн із розділенням відповідальності:

- Edge-рівень збирає дані;
- Transport забезпечує їх доставку;
- Cloud Processing виконує інтелектуальну обробку;
- Application надає користувацький доступ і візуалізацію.

Функціональний потік системи (рис. 2.8): генерація сигналів (BLE-маячки); збір і вимірювання RSSI (ESP32); передача даних (MQTT + Pub/Sub); обчислювальна обробка (Cloud Functions); збереження результатів (PostgreSQL); візуалізація та доступ користувача (API + клієнти).



Рисунок 2.8 – Функціональна IDEF0 діаграма

Блок-схема алгоритму роботи системи (рис. 2.9) містить такі етапи.



Рисунок 2.9 – Блок-схема алгоритму роботи системи

На початку BLE-маячки починають передавати advertising-пакети – це короткі радіосигнали з їхнім ідентифікатором. Далі ці сигнали приймаються ESP32 вузлами (anchor nodes), які працюють як сканери.

Після прийому сигналу кожен ESP32 вимірює рівень сигналу RSSI, що дає приблизне уявлення про відстань до маячка. На основі цих даних формується JSON-пакет, який містить ID маячка, ID вузла та значення RSSI.

Сформовані дані передаються через MQTT-брокер, який забезпечує надійну передачу повідомлень у хмару. Далі дані потрапляють у Google Cloud Pub/Sub, де відбувається їх прийом і чергування.

Отримання нового повідомлення активує Cloud Function – серверну функцію, яка обробляє дані. Спочатку виконується фільтрація та попередня обробка RSSI (видалення шуму, усереднення тощо). Потім застосовується модель затухання сигналу (Path Loss Model), щоб оцінити відстань до маячка.

На основі відстаней від кількох ESP32 вузлів виконується трилатерація – математичний метод визначення координат об'єкта. У результаті обчислюються координати (x , y).

Отримані координати зберігаються в базі даних PostgreSQL. Далі API надає доступ до цих даних клієнтським застосункам.

Клієнтський інтерфейс (веб або мобільний застосунок) візуалізує положення об'єкта в реальному часі або з історією переміщень.

Наприкінці алгоритм завершується, але передбачено зворотний зв'язок: дані з клієнта можуть знову запускати обробку (feedback loop), що забезпечує безперервне відстеження в реальному часі.

2.5 UML-діаграми системи

У діаграмі послідовності відображено взаємодію основних компонентів системи позиціонування на основі BLE iBeacon технології, MQTT-інфраструктури та хмарної обробки даних (рис. 2.10).

Актори системи:

- iBeacon-маячок – джерело сигналу, який періодично транслює пакети;
- користувач – ініціює перегляд звітів та аналітики.

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		39

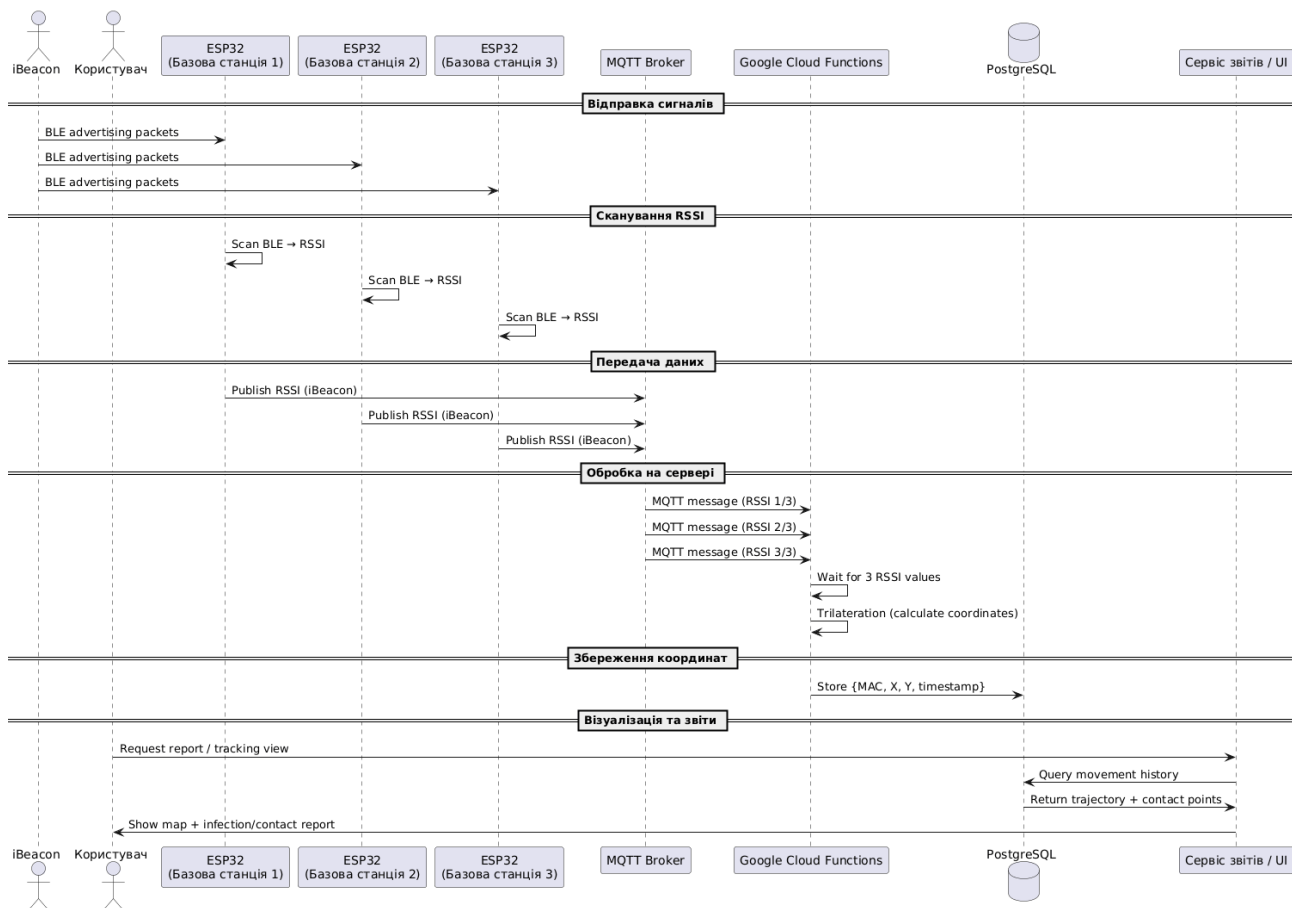


Рисунок 2.10 – Діаграма послідовності

Системні компоненти:

- ESP32 (базові станції) – три пристрої, розміщені в різних точках приміщення, які здійснюють прийом BLE-сигналів та вимірювання RSSI;
- MQTT-брокер – проміжний комунікаційний вузол для передачі телеметричних даних;
- Google Cloud Functions – серверна логіка, що виконує обробку даних і розрахунок координат;
- PostgreSQL – база даних для збереження історії координат iBeacon-міток;
- UI (сервіс звітів) – інтерфейс користувача для візуалізації результатів.

Процес починається з того, що iBeacon-маячок періодично надсилає BLE advertising-пакети. Ці сигнали приймаються трьома базовими станціями ESP32, кожна з яких виконує сканування Bluetooth-ефіру та визначає рівень сигналу (RSSI) для виявлених маячків.

Далі кожна базова станція передає отримані значення RSSI до MQTT-брокера, який забезпечує доставку повідомлень до серверної частини системи.

Google Cloud Functions, підписані на відповідний MQTT-топик, отримують RSSI-дані від усіх трьох станцій. Після отримання повного набору вимірювань виконується обчислення координат iBeacon-маячка методом трилатерації, що дозволяє визначити його положення в двовимірному просторі.

Розраховані координати разом із MAC-адресою та часовою міткою зберігаються у PostgreSQL, формуючи історію переміщення об'єктів.

На прикладному рівні Користувач взаємодіє з системою через UI-сервіс звітів, надсилаючи запит на перегляд історії переміщень або контактів конкретної мітки. UI отримує дані з бази та відображає траєкторію руху у графічному вигляді, включно з точками можливих контактів між мітками.

Приклад діаграми прецедентів для системи позиціонування IPS наведено на рисунку 2.11.



Рисунок 2.11 – Діаграма прецедентів

Діаграма прецедентів відображає взаємодію користувача з системою трекінгу iBeacon-маячків та основні функції, які реалізує система для забезпечення відстеження переміщення об'єктів у приміщенні.

У системі виділено два основні актори:

- iBeacon-маячок – пристрій, який транслює BLE-пакети з унікальним ідентифікатором та сигналом RSSI. Маячок не взаємодіє з системою напряму як користувач, але є джерелом даних для позиціонування;

- користувач – кінцевий користувач системи, який має доступ до перегляду аналітичної інформації та звітів про переміщення об'єктів.

Основні прецеденти системи включають:

- трансляція BLE сигналу – iBeacon-маячок періодично передає Bluetooth Low Energy пакети, що містять ідентифікаційні дані пристрою;

- збір RSSI даних – базові станції (внутрішня частина системи) здійснюють прийом сигналів від маячків та фіксують рівень RSSI для подальшої обробки;
- обробка та передача даних – отримані RSSI значення передаються через MQTT-протокол до серверної частини системи;
- розрахунок координат (трилатерація) – серверна функція виконує обчислення просторових координат маячка на основі даних щонайменше трьох базових станцій;
- збереження координат – обчислені координати разом із MAC-адресою маячка та часовою міткою зберігаються в PostgreSQL для подальшого аналізу;
- формування звіту – система агрегує історичні дані переміщення маячка та будує траєкторію його руху, а також визначає точки контактів між маячками;
- перегляд звіту (історія переміщень) – користувач отримує доступ до візуалізованої інформації про рух об'єктів у часі через інтерфейс системи.

Таким чином, діаграма демонструє повний цикл обробки даних: від збору сигналу BLE-маячком до формування аналітичного звіту для кінцевого користувача. Система реалізує принцип IoT-архітектури з розподіленою обробкою даних та централізованим зберіганням результатів.

2.6 Висновки до розділу

У результаті виконаного проектування сформовано архітектуру системи та створено фізичну схему взаємодії її складових. Було обґрунтовано вибір ключових компонентів, зокрема датчика BLE iBeacon та бездротового модуля ESP32.

Запропонована архітектура демонструє високу масштабованість завдяки використанню хмарних сервісів та низьку енергозатратність кінцевих пристроїв, що є критичним для довготривалих систем моніторингу в індустріальних та логістичних середовищах.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підп.	Дата		

3 РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Збір даних. MQTT

Будь-яка сучасна IoT-система зазвичай будується навколо двох базових процесів. Перший – це передавання телеметрії від пристроїв до хмарної інфраструктури з подальшим збереженням у сховищі. Такий підхід є типовим для більшості рішень у сфері Інтернету речей. У деяких випадках також реалізовується зворотна передача команд до обладнання, однак у межах цієї системи така функціональність не передбачена. Другий процес полягає в опрацюванні отриманих даних: це може бути розрахунок показників, формування аналітичних висновків або побудова наочних візуалізацій.

Етап збору та доставки інформації зазвичай реалізується за подібними принципами незалежно від конкретного застосування. Натомість аналітична складова значно варіюється залежно від задачі. Водночас без надійного отримання первинних даних будь-яка подальша обробка втрачає сенс. Саме тому особливу увагу приділяють створенню відмовостійкої, масштабованої та тісно інтегрованої з апаратною частиною підсистеми збору інформації.

Станом на 2026 рік платформа Google Cloud пропонує вже не один інструмент, а цілу екосистему інструментів для реалізації подібних задач. Варто зазначити, що сервіс Google Cloud IoT Core був офіційно виведений з експлуатації (ще у 2023 році), тому сучасні рішення зазвичай базуються на комбінації таких сервісів, як Google Cloud Pub/Sub, Cloud Run або Cloud Functions разом із брокерами MQTT сторонніх провайдерів (наприклад, HiveMQ чи EMQX).

Концепція serverless-обчислень залишається актуальною і навіть більш поширеною, ніж раніше. Вона дозволяє виконувати код у вигляді окремих функцій або контейнерів без необхідності керування інфраструктурою. Такі сервіси автоматично масштабуються, забезпечують високу доступність і спрощують розробку. За їх допомогою можна приймати потоки даних,

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		43

виконувати попередню обробку та передавати результати до баз даних або аналітичних систем.

Для зберігання інформації доступний широкий спектр рішень: від об'єктних сховищ, таких як Google Cloud Storage, до керованих баз даних, включаючи Cloud SQL (з підтримкою PostgreSQL) або аналітичні системи на кшталт BigQuery.

У межах розроблюваної системи позиціонування всередині приміщень було обрано передачу даних від Bluetooth-маячків із використанням протоколу MQTT. Він фактично став галузевим стандартом для IoT завдяки своїй легковаговості та ефективній моделі publish/subscribe. Пристрої публікують повідомлення у визначені топіки, а серверні компоненти асинхронно отримують їх через підписку (рис. 3.1).

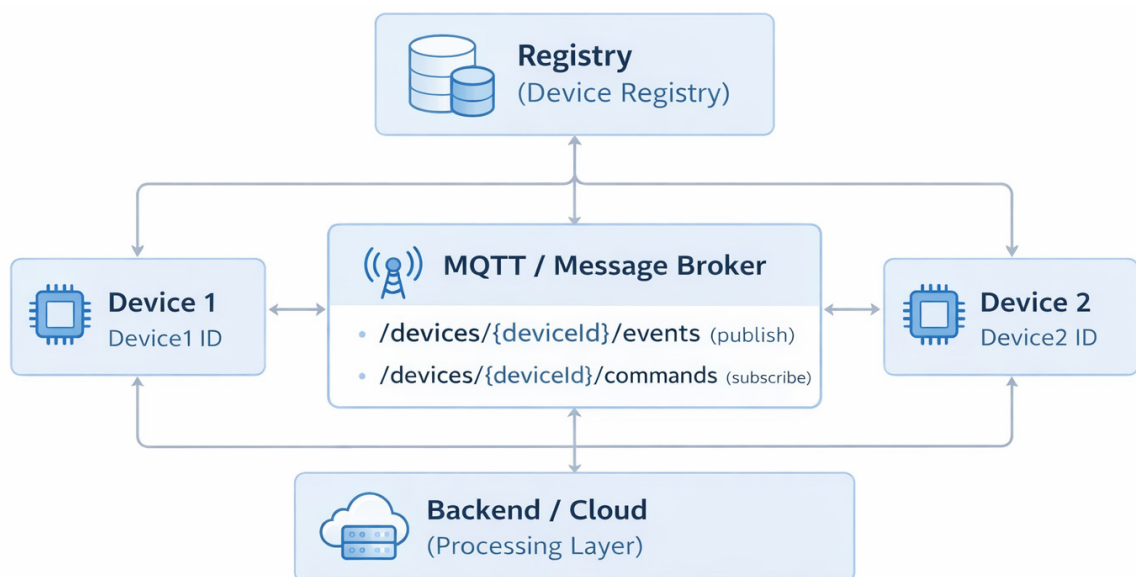


Рисунок 3.1 – Доступ пристроїв до MQTT в Google Cloud

У сучасних реалізаціях замість жорстко визначених структур топіків (як це було в IoT Core) застосовуються більш гнучкі схеми, які залежать від обраного брокера. Проте загальна логіка зберігається: пристрої передають телеметрію в окремі канали, а серверні служби агрегують ці дані для подальшої обробки.

У цій системі для збору інформації достатньо:

- створити логічну групу пристроїв (аналог реєстру);
- підключити до неї кілька вузлів збору даних.

Кожен вузол виконує роль базової станції, що фіксує сигнали від Bluetooth-маячків і передає їх у спільний канал повідомлень. Зворотний зв'язок із цими вузлами не використовується, тому механізми отримання команд не задіяні.

Як апаратна основа застосовується мікроконтролер ESP32, який має вбудовані модулі Wi-Fi та Bluetooth. Це робить його зручним вибором для IoT-проектів без потреби у додаткових компонентах.

Для пристроїв було створено прототип прошивки мовою C++, що виконує такі функції: сканування Bluetooth-середовища, вимірювання рівня сигналу (RSSI), агрегацію отриманих значень і передавання результатів через MQTT із використанням TLS-шифрування. Підключення до мережі здійснюється через Wi-Fi.

Розробка програмного забезпечення спрощується завдяки використанню Arduino Core від компанії Espressif Systems. Цей інструментарій надає доступ до великої кількості бібліотек і дозволяє швидко створювати прототипи навіть розробникам без глибокого досвіду роботи з мікроконтролерами. Блок-схема алгоритму отримання даних від маячків контролером ESP32 предсталена на рисунку 3.2.

Згідно цього алгоритму для ESP32 програмний код функції ініціалізації:

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

  WiFi.setHostname(HOSTNAME);
  WiFi.mode(WIFI_AP_STA);
  WiFi.begin(ssid, pass);
  while (WLFL.status() != WL_CONNECTED)
  {
    delay(1000);
  }
  configTime(-5*3600, 0, "pool.ntp.org", "time.nist.gov");
  now = time(nullptr);
  while (now < 1510592825) {
    delay(500);
    now = time(nullptr);
  }
  struct tm timeinfo;
  gmtime_r(&now, &timeinfo);
  net.setCACert(local_root_ca);
```

					БР.КІ-19.00.00.000 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підп.	Дата		

```

client.setServer(MQTT_HOST, MQTT_PORT);
client.setBufferSize(2048);
client.setCallback(receivedCallback);
mqtt_connect():
BLEDevice::init("");
}

```



Рисунок 3.2 – Блок-схема алгоритму сканування Bluetooth ефіру ESP32

Під час запуску пристрою виконується ініціалізація: встановлюється з'єднання з Wi-Fi мережею (з використанням попередньо заданих облікових даних), синхронізується системний час через NTP-сервер, після чого

налаштовується захищене TLS-з'єднання з MQTT-брокером. Коректний час є критично важливим для перевірки сертифікатів безпеки.

У процесі роботи виконується безперервний цикл, у якому контролюється стан мережових підключень і періодично запускається сканування Bluetooth-ефіру:

```
void ScanBeacons() {
    delay(1000);
    for (int i = 0; i < bufferIndex; i++) {
        buffer[i].count = 0;
        buffer[i].rssi = 0;
    }
    bufferIndex = 0;

    BLEScan* pBLEScan = BLEDevice::getScan();
    MyAdvertisedDeviceCallbacks cb;
    pBLEScan->setAdvertisedDeviceCallbacks (&cb);
    pBLEScan->setActiveScan(true);
    BLEScanResults foundDevices = pBLEScan->start(beaconScanTime);
    for (uint8_t i = 0; i < bufferIndex; i++) {
        buffer[i].rssi = buffer[i].rssi / buffer[i].count;
    }

    pBLEScan->stop();
    delay(1000);
}
```

Для кожного знайденого маячка накопичуються значення RSSI та кількість вимірювань, що дозволяє обчислити усереднений показник за певний інтервал:

```
class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {
public:
    void onResult(BLEAdvertisedDevice advertisedDevice) {
        if (lisIWhitelist(String(advertisedDevice.getAddress().toString().c_str())) {
            return;
        }
        extern uint8_t bufferIndex;
        extern BeaconData buffer[];
        if (bufferIndex >= 50) {
            return;
        }
        int index = FindElementIndex(String(advertisedDevice.getAddress().toString().c_str()));
        if (advertisedDevice.haveRSSI()) {
            buffer[index].rssi = buffer[index].rssi + advertisedDevice.getRSSI();
        } else {
            buffer[index].rssi = 0;
        }
    }
}
```

					БР.КІ-19.00.00.000 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        if (StringCbuffer[i].address).equals(mac)) {
            return i;
        }
    }
    bufferIndex += 1;
    return bufferIndex - 1;
}
};

```

Після цього сформовані дані відправляються у вигляді JSON-повідомлення до обраного топіка

```

void loop()
{
    boolean result;
    ScanBeacons();
    if (WiFi.status() != WL_CONNECTED)
    {
        while (WiFi.waitForConnectResult() != WL_CONNECTED)
        {
            WiFi.begin(ssid, pass);
            delay(10);
        }
    }
    else {
        if (client.connected()) {
            mqtt_connect();
        } else {
            client.loop();
        }
    }

    String payloadString = "{\"e\":[\"";
    for (uint8_t i = 0; i < bufferIndex; i++) {
        payloadString += "{\"m\":\"";
        payloadString += String(buffer[i].address);
        payloadString += "\",\"r\":";
        payloadString += String(buffer[i].rssi);
        payloadString += "\"";
        if (i < bufferIndex - 1) {
            payloadString += ",";
        }
    }
    payloadString += "],\"st\":\"";
    payloadString += String(WiFi.macAddress());
    payloadString += "\",\"coords\":{\"x\":\"";
    payloadString += String(STATION_X);
    payloadString += "\",\"y\":\"";
    payloadString += String(STATION_Y);
    payloadString += "\"}]\"";
    payloadString.getBytes(message_char_buffer, payloadString.length() + 1);
    result = client.publish(topicEvents.c_str(),
        message_char_buffer, payloadString.length(), false);
    delay(3000);
}

```

звідки вони можуть бути оброблені серверною частиною системи.

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		48

3.2 Обчислення координат об'єктів

Для визначення координат радіомітки було розроблено програму мовою JavaScript, що працює у середовищі Node.js та розгортається за допомогою Google Cloud Functions. У процесі обчислень застосовано підхід трилатерації, описаний у попередньому розділі. Обраний метод пояснюється його відносною простотою впровадження, відсутністю потреби у дорогому спеціалізованому обладнанні, а також достатньо високою точністю, яка в сучасних умовах залишається конкурентною порівняно з альтернативними алгоритмами позиціонування.

Тригер активується при надходженні нового повідомлення у відповідний топик, після чого серверна функція автоматично ініціює виконання алгоритму. Загальна логіка роботи хмарного обчислення наведена на блок-схемі алгоритму (рис. 3.3).

Визначення відстані між базовою станцією та маячком здійснюється за допомогою математичної залежності (3.1), що базується на вимірюванні рівня сигналу:

$$D = 10^{\frac{P_m - P_d}{10^n}}, \quad (3.1)$$

де P_m – еталонне значення RSSI на відстані одного метра від приймача.

P_d – фактичний рівень сигналу;

n – коефіцієнт згасання сигналу в середовищі.

Значення P_m необхідно попередньо калібрувати окремо для кожного типу Bluetooth-пристроїв, зважаючи на їхні апаратні особливості. Параметр n залежить від низки чинників, серед яких можуть бути перешкоди, матеріали приміщення, а також метеорологічні умови, включаючи вологість. У практичних системах цей параметр визначається експериментально або адаптивно коригується за допомогою алгоритмів машинного навчання.

Варто зазначити, що у сучасних рішеннях подібні обчислення часто доповнюються фільтрацією шуму (наприклад, за допомогою фільтра Калмана) для підвищення стабільності результатів.

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		49



Рисунок 3.3 – Блок-схема алгоритму розрахунку координат

Фрагмент програмної реалізації:

```

function calculateDistance (rssi) {
  let P=-60;
  let n = 3;
  let d = Math.pow(10, ((P-rssi) / (10*n)) );
  return d*px_meter;
}
  
```

Для розв’язання задачі визначення контактів між пристроями використовується розширення PostGIS для системи керування базами даних PostgreSQL. Це розширення забезпечує підтримку геопросторових типів даних і створення індексів, що значно пришвидшує виконання просторових запитів. З його допомогою можна працювати з такими об’єктами, як точки, лінії, полігони та растрові структури, а також виконувати операції аналізу — пошук найближчих об’єктів, визначення перетинів і обчислення відстаней.

Зм.	Арк.	№ докум.	Підп.	Дата

БР.КІ-19.00.00.000 ПЗ

Арк.

50

У підсумку для визначення фактів зближення радіоміток з урахуванням часових параметрів була реалізована серверна функція в Google Cloud Functions, яка формує та виконує SQL-запит із використанням можливостей PostGIS:

```
let dbClient = new pg.Client(config);
await dbClient.connect();
const query = `
  SELECT DISTINCT ON (ST_Distance(c1.geom, c2.geom))
    c1.id AS id1,
    c2.id AS id2,
    c1.beacon_mac AS mac1,
    c2.beacon_mac AS mac2,
    c1.time AS time1,
    c2.time AS time2,
    ST_AsText(c1.geom) AS p1,
    ST_AsText(c2.geom) AS p2,
    ST_Distance(c1.geom, c2.geom) AS distance
  FROM coordinates c1, coordinates c2
  WHERE c1.beacon_mac != c2.beacon_mac
    AND (c1.beacon_mac = $1 OR c2.beacon_mac = $1)
    AND c1.time BETWEEN c2.time - ($2 * INTERVAL '1 second')
      AND c2.time + ($2 * INTERVAL '1 second')
    AND ST_DWithin(c1.geom, c2.geom, $3)
  ORDER BY ST_Distance(c1.geom, c2.geom);
`;
const params = [params.mac, params.interval, params.distance];
let result = await dbClient.query(query, params);
```

У цьому запиті застосовується функція ST_Distance, що дозволяє обчислювати відстань між геометричними об'єктами.

MAC-адреса відповідної мітки, гранична дистанція взаємодії та часовий інтервал, у межах якого фіксується контакт, можуть змінюватися динамічно. Ці параметри передаються до серверної логіки як аргументи з клієнтського застосунку, що забезпечує гнучкість системи та можливість її адаптації до різних сценаріїв використання, включаючи сучасні IoT-рішення та системи моніторингу в реальному часі.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підп.	Дата		

3.3 Реалізація клієнтської частини

Завершальним кроком є розробка клієнтського застосунку, який забезпечує формування та візуалізацію звіту щодо переміщення осіб, що використовують Bluetooth-ідентифікатор, у межах приміщення на основі історії переміщень.

Для створення клієнтської частини було обрано підхід, орієнтований на мультиплатформеність. Такий спосіб передбачає застосування сучасних інструментів розробки, які дають змогу створювати програмне забезпечення одночасно для кількох операційних систем без необхідності повного переписування коду під кожен платформу.

Ключова ідея цього підходу полягає у повторному використанні єдиної кодової бази, що значно оптимізує витрати часу та людських ресурсів. У більшості випадків достатньо однієї команди інженерів, а для невеликих проєктів – навіть одного розробника. Дизайнери, своєю чергою, працюють із єдиною системою інтерфейсних компонентів, яка використовується на різних платформах. Це дозволяє зменшити загальні витрати на розробку, скоротити терміни виконання та підвищити узгодженість продукту. Додатковою перевагою є використання спільної бізнес-логіки, що спрощує тестування та знижує ризик появи помилок. Водночас недоліком такого підходу все ще може залишатися дещо нижча продуктивність у порівнянні з повністю нативними рішеннями, хоча у 2026 році ця різниця значною мірою зменшилася завдяки розвитку інструментів компіляції та оптимізації.

Як основний інструмент реалізації клієнтської частини було обрано фреймворк Flutter. Це рішення з відкритим вихідним кодом, створене компанією Google, яке активно розвивається і станом на 2026 рік підтримує не лише мобільні платформи (iOS та Android), а й вебсередовище, настільні операційні системи (Windows, macOS, Linux), а також вбудовані пристрої. Сучасні версії Flutter орієнтовані на високу продуктивність, покращену інтеграцію з нативними API та підтримку нових архітектур процесорів.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підп.	Дата		

Для написання програмного коду використовується мова Dart, яка також розроблена Google і з часом отримала суттєві оновлення, включаючи покращену систему типів, асинхронне програмування та ефективні механізми компіляції. Станом на 2026 рік Dart підтримує як JIT-компіляцію для швидкої розробки, так і AOT-компіляцію для досягнення максимальної продуктивності в релізних версіях застосунків.

На відміну від деяких альтернативних технологій, Flutter не покладається на JavaScript-рушії. Код, написаний на Dart, компілюється безпосередньо у нативний машинний код, що дозволяє досягати швидкодії, співставної з програмами, створеними на Swift, Kotlin або C++. Це забезпечує плавну роботу інтерфейсу та швидкий відгук застосунку навіть при високих навантаженнях.

Для відображення інтерфейсу Flutter використовує власний графічний рушій (у сучасних версіях – Skia або Impeller, залежно від платформи), що дозволяє повністю контролювати процес рендерингу. Завдяки цьому інтерфейс виглядає однаково на всіх пристроях. Хоча застосунок може мати дещо більший розмір через вбудовані компоненти рушія, це компенсується стабільністю відображення та високою швидкістю роботи. На відміну від підходів, які використовують WebView або проміжні шари для взаємодії з нативними елементами, Flutter забезпечує більш передбачувану поведінку та спрощує створення уніфікованого дизайну.

Особливістю цього фреймворку є підхід до побудови користувацького інтерфейсу. Якщо у традиційній Android-розробці логіка та представлення розділені (код і XML-розмітка), то тут усе описується програмно. Це дає змогу більш гнучко керувати поведінкою елементів та їхнім відображенням.

Інтерфейс формується за декларативним принципом із використанням віджетів. Уся структура представляє собою вкладену ієрархію компонентів, де кожен елемент є частиною іншого. Оновлення відбувається реактивно: перерисовуються лише ті частини, які зазнали змін, що позитивно впливає на продуктивність.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підп.	Дата		

Приклад оголошення віджету:

```
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: const Text('Заголовок'),  
    ),  
    body: const Center(  
      child: Text('Текст посередині'),  
    ),  
  );  
}
```

Подання цього віджету у вигляді дерева представлено на рисунку 3.4

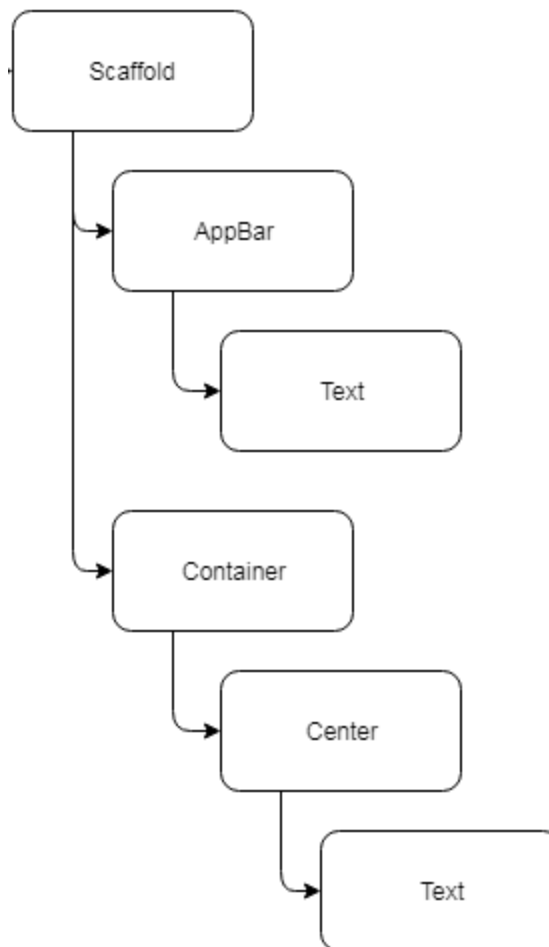


Рисунок 3.4 – Подання віджету

Разом із тим, зі зростанням складності проєкту виникає необхідність чіткого розмежування відповідальностей між різними частинами системи. Без належної організації коду застосунок може стати складним для підтримки та масштабування. Саме тому у сучасній розробці активно використовуються архітектурні підходи, що дозволяють структурувати програму.

З часом було сформовано низку перевірених рішень, відомих як архітектурні патерни програмного забезпечення. Серед них найбільш поширеними залишаються MVC, MVP, MVVM, а також сучасні варіації на кшталт Clean Architecture та Feature-first підходів.

У даному проєкті за основу було взято архітектуру MVVM, адаптовану відповідно до специфіки системи. Її складові виконують такі функції (рис. 3.5):

- View відповідає за опис інтерфейсу, взаємодіє з ViewModel та отримує дані для відображення, реагуючи на зміни стану;
- ViewState зберігає поточні параметри екрану та забезпечує автоматичне оновлення UI при їх зміні;

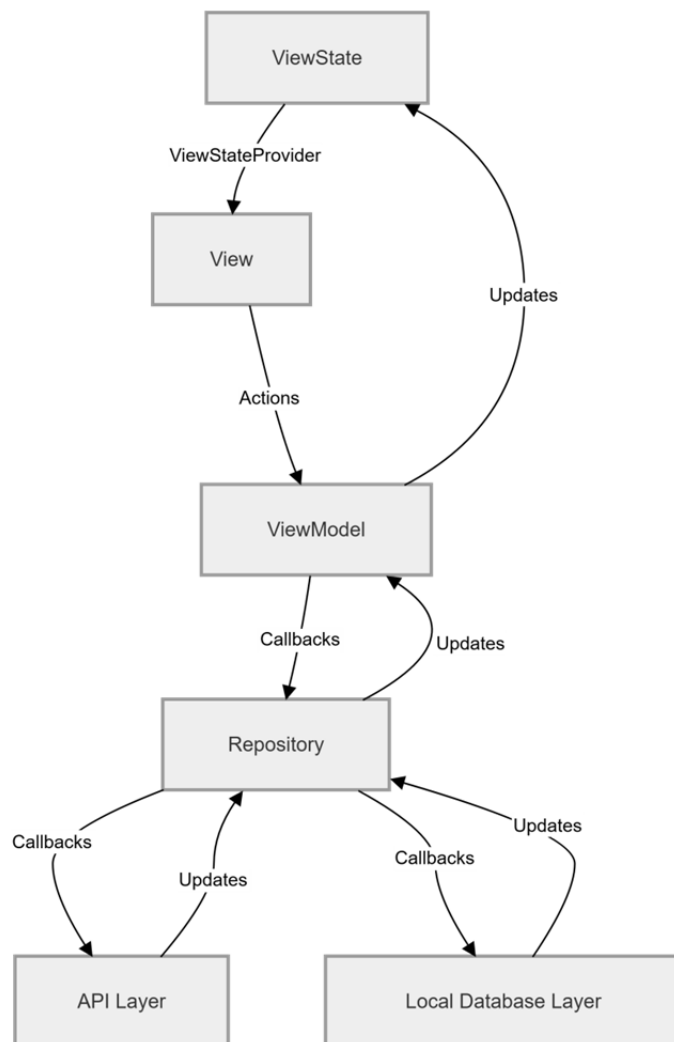


Рисунок 3.5 – Схема взаємодії компонентів

- ViewModel виступає посередником між інтерфейсом і бізнес-логікою, керує станом представлення та обробляє дії користувача;

– Repository інкапсулює роботу з даними, взаємодіє із зовнішніми сервісами, API або локальними сховищами (наприклад, базами даних чи кешем).

Такий підхід дозволяє створити гнучку, масштабовану та зручну для підтримки архітектуру, що відповідає сучасним вимогам до розробки програмних систем.

Інтерфейс початкової сторінки відображено на рисунку 3.6. На ньому видно мапу приміщення розміром 5х7 м в якому перебувають і переміщуються люди. На мапі, після завантаження інтерфейсу, відображаються траєкторії переміщення (сірим кольором) усі людей у приміщенні з моменту завантаження інтерфейсу. Якщо натиснути клавішу «Очистити шлях», усі траєкторії, побудовані до цього моменту стираються і починають будуватись нові, з поточного часу. В даній реалізації відображається мапа приміщення без внутрішні стін, перешкод для пересування, тощо. За потреби можна додати план приміщення (рис. 3.7) чи змінити розміри (масштаб) мапи та сітки.

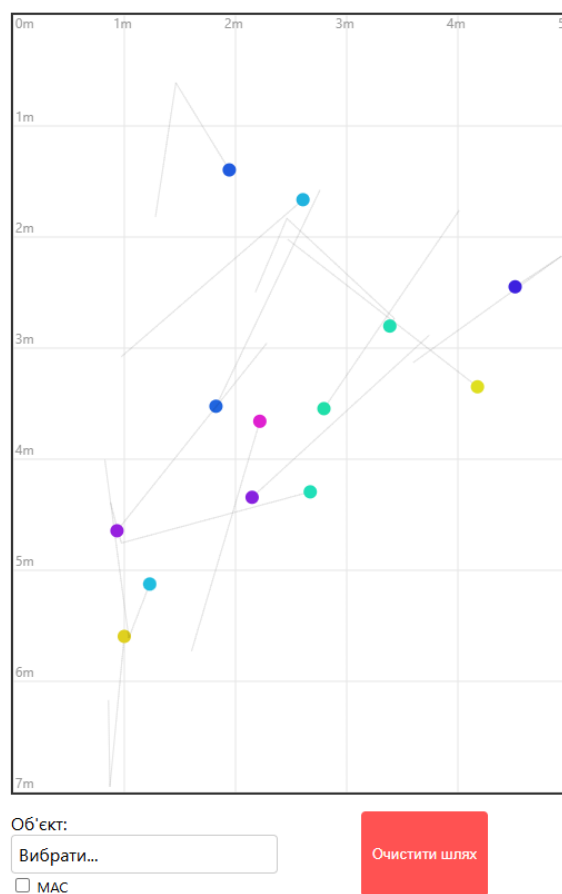


Рисунок 3.6 – Інтерфейс початкової сторінки

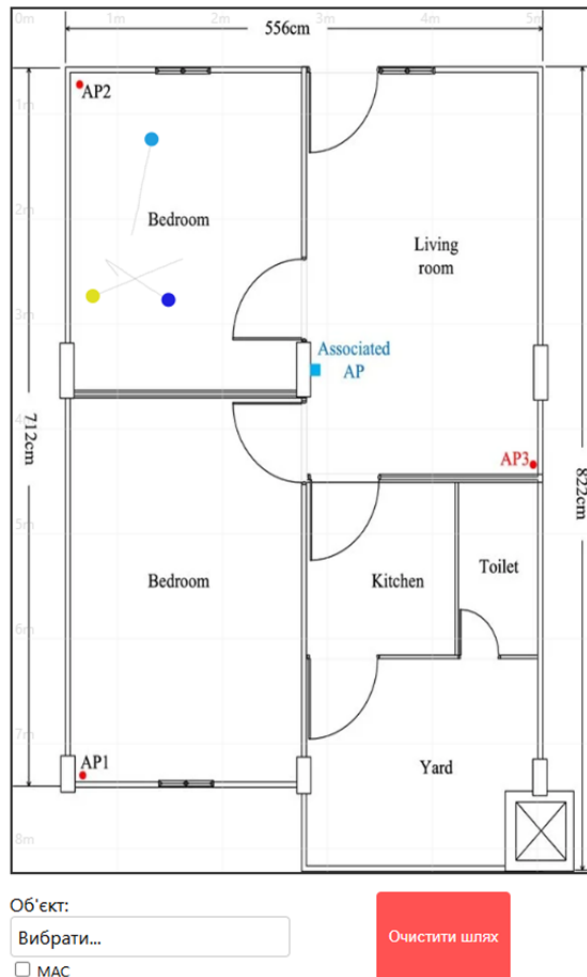


Рисунок 3.7 – Мапа-план приміщення

Кольорові точки – це позиції об'єктів. Різні кольори точок дозволяють відрізняти об'єкти.

Випадаючий список «Об'єкт» дозволяє обрати конкретний один об'єкт чи кілька для аналізу. Список із чекбоксами, наприклад P1, P2, P3 – увімкнення/вимкнення відображення окремих об'єктів (рис. 3.8). Для обраних об'єктів траєкторії руху уже будуть відображатись не сірим кольором, а тим, що відповідає кольору вибраного об'єкту (рис. 3.8).

У правому нижньому куті карти відображаються поточні координати об'єктів наприклад, P1: 2.80, 6.46, P1: 2.92, 5.16 (рис. 3.8). Це дозволяє точно відслідковувати положення у числовому вигляді.

При встановленні мітки «MAC» на мапі будуть відображатись MAC-адреси маячків, які відповідають обраним особам (рис. 3.9).

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		57

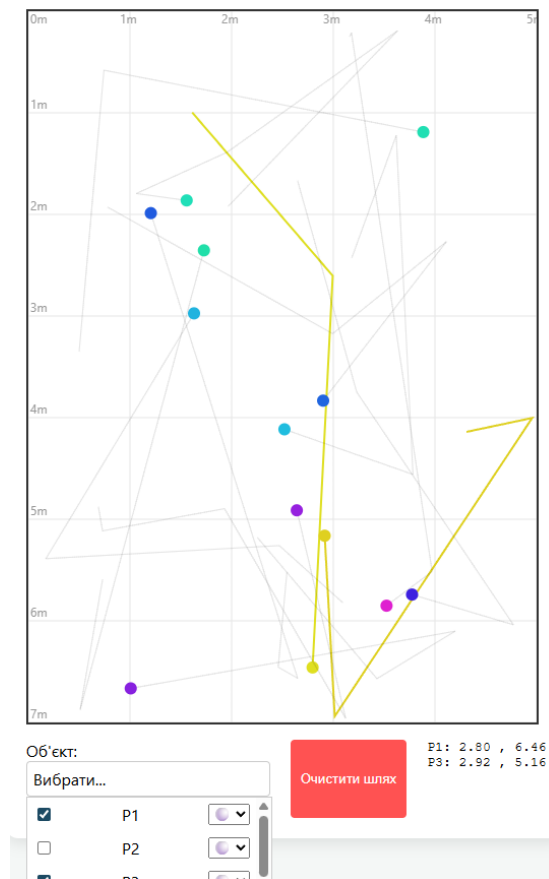


Рисунок 3.8 – Вибір об'єктів для відображення

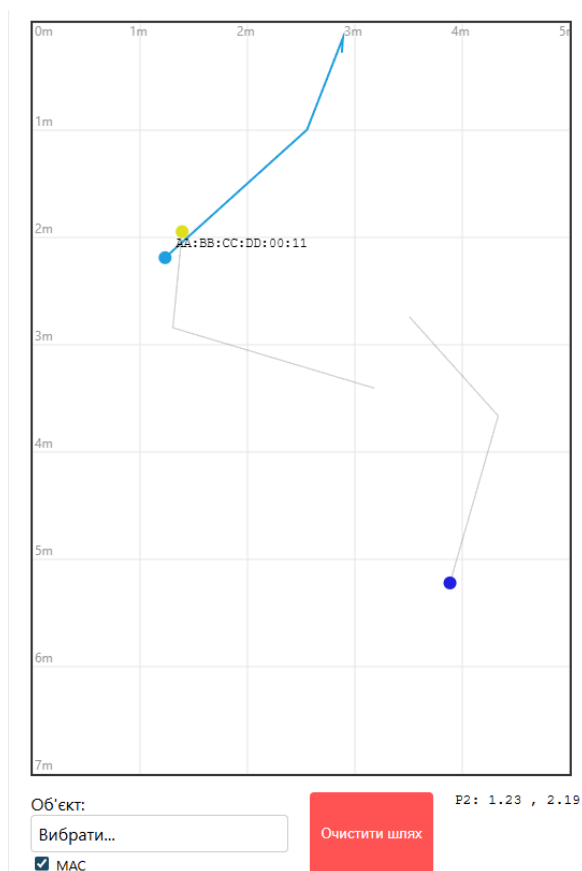


Рисунок 3.9 – Відображення MAC-адреси маячка

					БР.КІ-19.00.00.000 ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підп.	Дата		

Для різних об'єктів призначаються випадково різні кольори точок, які дозволяють розрізняти об'єкти. Проте також їх можна групувати, змінюючи колір точки у меню «Вибрати». У реалізованому варіанті інтерфейсу є змога розподіляти на дві групи та призначати, відповідно, червоний та зелений кольори точкам. Нехай здійснюється поділ на здорових осіб – зелений колір (P1), та контактних осіб (з підозрою на ГРВІ) – червоний (P2) (рис. 3.10). Тоді можна буде відстежити, чи був контакт між цими особами і, якщо був, коли саме.

Також можна знову повернути колір за замовчуванням для точки, вибравши відповідний пункт (P3, рис. 3.10).

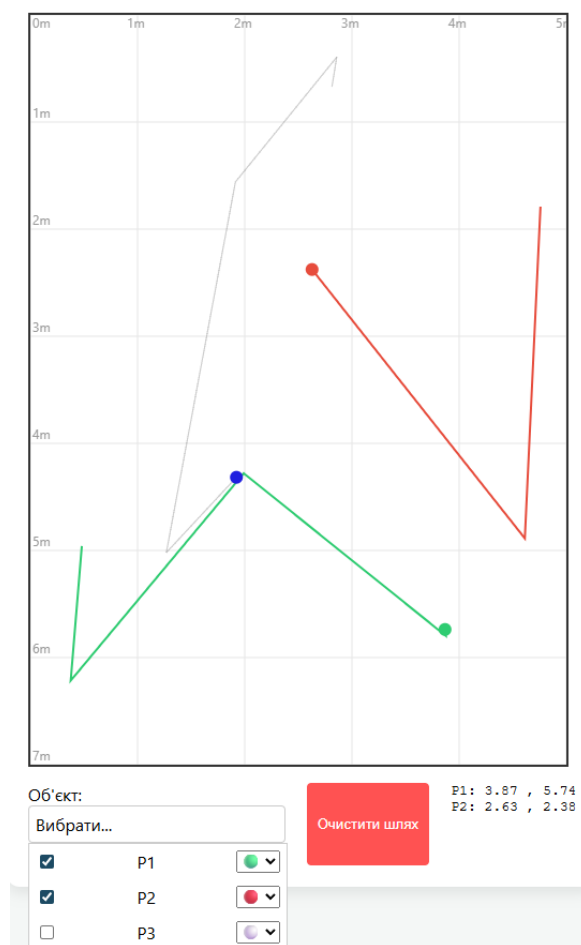


Рисунок 3.10 – Групування об'єктів

В запропонованому варіанті система буде:

- точки позиціонування;
- траєкторію руху.

Інтерфейс дозволяє:

- відслідковувати позицію об'єктів у реальному часі;
- аналізувати траєкторію руху;
- вмикати/вимикати відображення різних об'єктів;
- очищати та повторно будувати маршрут.

За потреби функціонал можна легко розширити.

Графічний інтерфейс системи IPS, який забезпечує візуалізацію координат об'єктів у приміщенні, відображення їх траєкторій, а також інтерактивне керування відображенням даних та очищенням результатів позиціонування.

3.4 Висновки до розділу

Розроблена системи визначення місцеположення в приміщенні, що ґрунтується на використанні алгоритму трилатерації. Створене рішення забезпечує фіксацію та збереження повної історії переміщень користувачів із використанням маячків у базі даних. У ході реалізації проєкту було опрацьовано взаємодію з iBeacon-пристроями, мікроконтролером ESP32, а також використано можливості хмарної платформи Google Cloud для обробки та зберігання інформації.

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		60

ВИСНОВКИ

У процесі виконання цієї роботи всі визначені цілі та завдання були успішно реалізовані. Було детально проаналізовано численні характеристики, підходи й технологічні рішення, які можуть застосовуватися для встановлення місцеположення об'єктів у межах закритих просторів.

Отримані результати дослідження засвідчили, що в реальних умовах для розв'язання задач внутрішнього позиціонування доцільно використовувати засоби бездротового зв'язку. Кожна з таких технологій має власні переваги та недоліки, що впливають на точність, стабільність і складність реалізації системи. Додатково було визначено дві ключові групи методів локалізації, які застосовуються разом із зазначеними технологіями: range-based підходи, що базуються на вимірюванні відстані, та range-free методи, які не потребують точного визначення дистанції між об'єктами.

Підсумком проведеної роботи стала розробка системи визначення місцеположення в приміщенні, що ґрунтується на використанні алгоритму трилатерації. Створене рішення забезпечує фіксацію та збереження повної історії переміщень користувачів із використанням маячків у базі даних. У ході реалізації проекту було опрацьовано взаємодію з iBeacon-пристроями, мікроконтролером ESP32, а також використано можливості хмарної платформи Google Cloud для обробки та зберігання інформації.

Запропонована система надає змогу оперативно визначати місцезнаходження осіб у межах контрольованого простору. Це, у свою чергу, відкриває перспективи для подальшого застосування, зокрема для вибіркового регулювання переміщення людей, підвищення рівня безпеки або оптимізації управління потоками в різних типах приміщень.

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		61

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Ониськів Р. Б. IoT система моніторингу та аналізу інформації про наявність людей в приміщенні : кваліфікаційна робота бакалавра : 123 Комп'ютерна інженерія / Ониськів Р. Б. – Тернопіль, 2023. – 60 с.

2 Tsai, H.-C., Chiu, C.-J., Tseng, P.-H., & Feng, K.-T. Refined Autoencoder-Based CSI Hidden Feature Extraction для Indoor Spot Localization. 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall). – 2018.

3 Kawdungta, R., Kawdungta, S., Torrungrueng, D., & Phongcharoenpanich, C. Switched Beam Multi-Element Circular Array Antenna Schemes для 2D Single-Anchor Indoor Positioning Applications. IEEE Access, 9, 58882-58892 – 2021.

4 Molina, B., Olivares, E., Palau, CE, & Esteve, M. A Multimodal Fingerprint-Based Indoor Positioning System for Airports. IEEE Access, 6, 10092-10106. 2018.

5 Патент на корисну модель №134569. Спосіб позиціювання за сукупністю сигналів від всенаправлених кутомірних радіомаяків та прогнозованої інформації / Остроумов І.В., Харченко В.П., Кузьменко Н.С. Національний авіаційний університет; заявка у 2018 12431; опубл. 27.05.2019.

6 Indoor Location Market Global forecast to 2026. URL: <https://www.marketsandmarkets.com/Market-Reports/indoor-location-market-989.html> (дата звернення: 15.03.2026).

7 R R. SambathKumar, S. Gowshameed and S. Arunmozhi, "Arithmetical Analysis of WSN based Indoor Positioning Localization Systems with Kalman Filtering," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), 2021.

8 P P. Spachos and K. N. Plataniotis, "BLE Beacons for Indoor Positioning at an Interactive IoT-Based Smart Museum," in IEEE Systems Journal, vol. 14, no. 3, pp. 3483-3493, 2020.

9 F. Betti Sorbelli, CM Pinotti, S. Silvestri i SK Das, "Measurement Errors в Range-based Localization Algorithms for UAVs: Analysis and Experimentation," in IEEE Transactions on Mobile Computing, 2020.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підп.	Дата		

10 Nemer, I., Sheltami, T., Shakshuki, E. та ін. Performance evaluation of range-free localization algorithms for wireless sensor networks. Pers Ubiquit Comput 25, 177-203, 2021.

11 N. Pakanon, M. Chamchoy and P. Supanakoon, "Study on Accuracy of Trilateration method for Indoor Positioning with BLE Beacons," 6th International Conference on Engineering, Applied Sciences and Technology (ICEAST), 2020.

12 Zafari, F.; Gkelias, A.; Leung, KK A Survey of Indoor Localization Systems and Technologies. IEEE Commun. Surv. Tutor. 2019.

13 Dong, Y.; Shan, F.; Dou, G.; Cui, Y. The Research and Application of Indoor Location Algorithm Based on Wireless Sensor Network. In Proceedings of the IEEE 3rd International Conference Communication Software and Networks, Xi'an, China, 27–29 May 2011.

14 Bharadwaj, R.; Parini, C.; Alomainy, A. Experimental Investigation of 3-D Human Body Localization Using Wearable Ultra-Wideband Antennas. IEEE Trans. Antennas Propag. 2015.

15 Chen, R. A Novel Method for Indoor Location Identification. In Proceedings of the IEEE 2nd International Symposium on Aware Computing, Tainan, China, 1-4 November 2010.

16 Chen, SW; Seow, CK; Tan, SY Elliptical Lagrange-Based NLOS Tracking Localization Scheme. IEEE Trans. Wirel. Commun. 2016.

17 Методи локального позиціонування. URL: <https://habr.com/company/realtrac/blog/301706/> (Дата звернення: 02.04.2023).

18 Li, Z.; Dimitrova, D.C.; Raluy, D.H.; Braun, T. TDOA for Narrow-band Signal with Low Sampling Rate and Imperfect Synchronization. In Proceedings of the IEEE 7th IFIP Wireless and Mobile Networking Conference (WMNC), Vilamoura, Portugal, 2014.

19 Yassin, A.; Nasser, Y.; Awad, M.; Al-dubai, A. Simultaneous Context Inference and Mapping using mm-Wave for Indoor Scenarios. In Proceedings of the IEEE International Conference on Communications (ICC), Paris, France, 2017.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підп.	Дата		

20 Кокорева О. В., Костюкович А. Є. Оцінка похибки вимірювань місцезнаходження абонента в мережі Wi-Fi // Програмні системи та обчислювальні методи. 2019.

21 Zafari, F.; Gkelias, A.; Leung, KK A Survey of Indoor Localization Systems and Technologies. IEEE Commun. Surv. Tutor. 2019.

22 Wang, X.; Gao, L.; Mao, S.; Pandey, S. CSI-based Fingerprinting for Indoor Localization: A Deep Learning Approach. IEEE Trans. Veh. Technol., 2016.

23 Hsieh, H.-Y.; Prakosa, S.W.; Leu, J.-S. Towards the Implementation of Recurrent Neural Network Schemes for WiFi Fingerprint-Based Indoor Positioning. In Proceedings of the 2018 IEEE 88th Vehicular Technology Conference (VTCFall), Chicago, IL, USA, 2018.

24 Ding, N.; Wagner, D.; Chen, X.; Pathak, A.; Hu, YC; Rice, A. Characterizing i modeling impact of wireless signal strength on smartphone battery drain. ACM Sigmetrics Perform. Eval. Rev. 2013.

25 Cidronali, A.; Maddio, S.; Giorgetti, G.; Manes, G. Analysis and Performance of Smart Antenna for 2.45-GHz Single-Anchor Indoor Positioning. IEEE Trans. Microw. Theory Tech. 2010.

26 Rusli, M.E.; Ali, M.; Jamil, N.; Din, M.M. An Improved Indoor Positioning Algorithm Based on RSSI-Trilateration Technique for Internet of Things (IOT). In Proceedings of the 2016 International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia, 2016.

27 Ren, J.; Wang, Y.; Niu, C.; Song, W.; Huang, S. A Novel Clustering Algorithm для Wi-Fi Indoor Positioning. IEEE Access, 2019.

28 Shi, S.; Sigg, S.; Chen, L.; Ji, Y. Accurate Location Tracking From CSIBased Passive Device-Free Probabilistic Fingerprinting. IEEE Trans. Veh. Technol., 2018.

29 Yan, D.; Kang, B.; Zhong, H.; Wang, R. Research on positioning system based on Zigbee communication. In Proceedings of the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 2018.

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		64

30 Thuong, NT; Phong, HT; Do, D.; van Hieu, P.; Loc, DT Android Application for Wi-Fi Based Indoor Position: System Design and Performance Analysis. In Proceedings of the 2016 International Conference on Information Networking (ICOIN), Kota Kinabalu, Malaysia, 2016.

31 30 Kim, B.; Kwak, M.; Lee, J.; Kwon, T.T. A Multit-proned Approach for indoor Positioning with Wi-Fi, Magnetic and Cellular Signals. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 2014.

32 Yu, N.; Zhan, X.; Zhao, S.; Wu, Y.; Feng, R. A Precise Dead Reckoning Algorithm Based on Bluetooth and Multiple Sensors. IEEE Internet Things J., 2018.

33 Sou, S.-I.; Lin, W.-H.; Lan, K.-C.; Lin, C.-S. Indoor Location Learning Over Wireless Fingerprinting System With Particle Markov Chain Model. IEEE Access, 2019.

34 C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola i C. Fischione, "A Survey of Enabling Technologies for Network Localization, Tracking and Navigation," IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3607 3644, 2018.

35 Sadowski, S.; Spachos, P. RSSI-Based Indoor Localization With the Internet of Things. IEEE Access, 2018.

36 Giovanelli, D.; Farella, E.; Fontanelli, D.; Macii, D. Bluetooth-Based Indoor Positioning Through ToF and RSSI Data Fusion. In Proceedings of IEEE International Conference Indoor Positioning and Indoor Navigation (IPIN), Nantes, France, 2018.

37 Sthapit, P.; Gang, H.-S.; Pyun, J.-Y.; Pyurr, J.-Y. Bluetooth Based Indoor Positioning Using Machine Learning Algorithms. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics—Asia (ICCE-Asia), Jeju, Korea, 2018.

38 Li, G.; Geng, E.; Ye, Z.; Xu, Y.; Zhu, H. An Indoor Positioning Algorithm Based on RSSI Real-time Correction. In Proceedings of the 2018 14th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 2018.

39 ZigBee vs Bluetooth - розуміємося на тонкощах. URL: <https://sprut.ai/client/article/3922> (Дата звернення: 10.04.2023).

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		65

40 Lo, L.; Li, C. Passive UHF-RFID Localization Based на Similarity Measurement of Virtual Reference Tags. IEEE Trans. Instrum. Meas., 2019.

41 A Dual-Band Low-Power-Consumption Active RFID Tag Based on a Meander FPCB Antenna for Subway Vehicle Management, Electromagn Eng Sci., 2021.

42 Skiljo, M.; Solic, P.; Blazevic, Z.; Perkovic, T. Analysis of Passive RFID Applicability in Retail Store: What Can We Expect? Sensors, 2020.

43 Li, J.-Q.; Feng, G.; Wei, W.; Luo, C.; Cheng, L.; Wang, H.; Song, H.; Ming, Z. PSOTrack: A RFID-Based System for Random Moveing Objects Tracking in Unconstrained Indoor Environment. IEEE Internet Things J., 2018.

44 Hanssens, B.; Plets, D.; Tanghe, E.; Oestges, C.; Gaillot, D.P.; Lienard, M.; Li, T.; Steendam, H.; Martens, L.; Joseph, W. An Indoor Variance-Based Localization Technique Utilizing the UWB Estimation of Geometrical Propagation Parameters. IEEE Trans. Antennas Propag., 2018.

45 Mekonnen, Z.W.; Slotke, E.; Luecken, H.; Steiner, C.; Wittneben, A. Constrained maximum likelihood positioning for UWB based human motion tracking. In Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation, Zurich, Switzerland, 2010.

46 UWB: Pros and Cons of Ultra-Wideband Technology. URL: <https://www.konsyse.com/articles/uwb-pros-and-cons-of-ultra-wideband-technology/> (Дата звернення: 13.04.2026).

47 Рубізова С.А. Огляд та порівняльне дослідження мікроконтролерів// Проблеми сучасної науки та освіти . 2017.

48 TECH GEEK. URL: <https://tech-geek.com/choosing-microcontroller/> (Дата звернення: 15.04.2026).

49 Perry Xiao, «Designing Embedded Systems and the Internet of Things (IoT) with the ARM mbed», John Wiley & Sons Limited, 2018. 313 p.

50 ESP8266 WI-FI MCU. Espressif, URL: <https://www.espressif.com/en/products/socs/esp8266> (Дата звернення: 13.04.2026).

51 WI-FI & Bluetooth MCU. Espressif, URL: <https://www.espressif.com/en/products/socs/esp32> (Дата звернення: 13.04.2026).

					БР.КІ-19.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		66

52 Choosing best hardware for your next IoT project. URL: <https://developer.ibm.com/technologies/iot/articles/iot-lp101-best-hardware-devicesiot-project/> (Дата звернення: 14.04.2026).

53 hIoTron. URL: <https://www.hiotron.com/iot-architecture-layers/> (Дата звернення: 20.04.2026).

54 Importance of Cloud Computing for IoT. URL: <https://www.einfochips.com/blog/importance-of-cloud-computing-for-large-scale-iiotsolutions/> (дата звернення: 16.04.2026).

55 Архітектура платформи Google Cloud. URL: <https://cloud.google.com/architecture> (дата звернення: 18.04.2026).

56 Технології інтернету речей. Навчальний посібник для студ. спеціальності 126 «Інформаційні системи та технології», спеціалізація «Інформаційне забезпечення робототехнічних систем» / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. 2021. 271 с.

					БР.КІ-19.00.00.000 ПЗ	Арк.
						67
Зм.	Арк.	№ докум.	Підп.	Дата		

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: "Розроблення мікроконтролерної системи детектування розташування людей в приміщенні на основі ESP32"

Обсяг пояснювальної записки 67 аркушів.

36 рисунки;

2 таблиць;

- додатки.

Дата завершення роботи: *11 червня 2026 р.*

Підпис студента-дипломника _____ *Плахетко Я. Ю.*