Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: *IoT система моніторингу та аналізу інформації про наявність людей в приміщенні*

Виконав: студент *IV курсу,* групи *СІ-41*
спеціальності *123 Комп'ютерна інженерія*
(шифр і назва спеціальності)

_____ *Ониськів Р.Б.*
(підпис) (прізвище та ініціали)

Керівник _____ *Баран І.О.*
(підпис) (прізвище та ініціали)

Нормоконтроль _____ _____
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ *Осухівська Г.М.*
(підпис) (прізвище та ініціали)

Рецензент _____ _____
(підпис) (прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет *комп'ютерно-інформаційних систем і програмної інженерії*
(повна назва факультету)

Кафедра *комп'ютерних систем та мереж*
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ *Осухівська Г.М.*
(підпис) (прізвище та ініціали)
«__» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня *Бакалавр*
(назва освітнього ступеня)

за спеціальністю *123 Комп'ютерна інженерія*
(шифр і назва спеціальності)

Студенту *Ониськіву Роману Богдановичу*
(прізвище, ім'я, по батькові)

1. Тема роботи *IoT система моніторингу та аналізу інформації про наявність людей в приміщенні*

Керівник роботи *Баран Ігор Олегович., к.т.н., доц.*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від *« 28 » 02 2023 року № 4/7-238*

2. Термін подання студентом завершеної роботи *22.06. 2023 р.*

3. Вихідні дані до роботи *Технічне завдання*

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ.
1. Аналіз технічного завдання.
2. Проектна частина.
3. Практична частина.
4. Безпека життєдіяльності, основи охорони праці.
Висновки

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

1. Основні методи локалізації.
2. Структура IoT рішень та Схема роботи протоколу MQTT
3. Алгоритми сканування BLUETOOTH ефіру для ESP32 та розрахунку координат
4. Лістинги фрагментів програмного коду.
5. Схеми розробленої IoT системи та взаємодії програмних компонент

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	*Гурик О.Я., доц. кафедри МТ*		

7. Дата видачі завдання 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	*Ознайомлення з завданням до кваліфікаційної роботи*	*01.03 – 05.03*	*Виконано*
2.	*Розробка технічного завдання*	*06.03 – 16.03*	*Виконано*
3.	*Підбір джерел про IoT системи моніторингу та аналізу*		
	інформації про наявність людей в приміщенні	*17.03 – 28.04*	*Виконано*
4.	*Опрацювання джерел про такі системи*	*29.04 – 10.05*	*Виконано*
5.	*Виконання дослідження щодо IoT системи моніторингу*		
	та аналізу інформації про наявність людей в приміщенні	*11.05 – 18.05*	*Виконано*
6.	*Розроблення програмного коду*	*19.05 – 26.05*	*Виконано*
7.	*Оформлення розділу «Аналіз технічного завдання»*	*27.05 – 30.05*	*Виконано*
8.	*Оформлення розділу «Проектна частина»*	*30.05 – 02.06*	*Виконано*
9.	*Оформлення розділу «Практична частина»*	*03.06 – 08.06*	*Виконано*
10.	*Виконання завдання до підрозділу «Безпека*		
	життєдіяльності, основи охорони праці»	*20.05 – 04.06*	*Виконано*
11.	*Оформлення кваліфікаційної роботи*	*09.06 – 12.06*	*Виконано*
12.	*Нормоконтроль*	*12.06 – 16.06*	*Виконано*
13.	*Перевірка на плагіат*	*16.06 – 18.06*	*Виконано*
14.	*Попередній захист кваліфікаційної роботи*	*17.06 – 19.06*	*Виконано*
15.	*Захист кваліфікаційної роботи*	*23.06*	

Студент _____ *Ониськів Р.Б.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Баран І.О.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

IoT система моніторингу та аналізу інформації про наявність людей в приміщенні // Кваліфікаційна робота бакалавра // Ониськів Роман Богданович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно–інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІ–41 // Тернопіль, 2023 // с. – 60, рис. – 31, табл. – 4, аркушів А1 – 5, бібліогр. – 59.

Ключові слова: ЛОКАЛІЗАЦІЯ, IPS, ХМАРНІ ОБЧИСЛЕННЯ, РАДІОМІТКА, МІКРОКОНТРОЛЕР, MQTT, FLUTTER

Кваліфікаційна робота присвячена розробці IoT-системи, котра утримує користувачів від можливих контактів з хворими людьми за даними історії їх позиціонування всередині приміщення.

Досліджено основні методи та технології для локалізації у приміщенні (трилатерації, триангуляції, радіовідбитків). Проведено аналіз та вибір параметрів для локалізації. Здійснено вибір метрик, котрі потрібні для здійснення розрахунку координат людей всередині приміщення. Виконано дослідження існуючих IoT платформ. Побудована архітектура системи. Виконано огляд існуючих протоколів передачі даних. Значна увага також приділена застосування хмарних обчислень з використанням можливостей платформи Google Cloud.

Побудовані схеми взаємодії компонентів та функціонування IoT системи. Наведені алгоритми та їх програмна реалізація Bluetooth сканування ефіру і розрахунку координат радіоміток. Здійснено визначення контактів переміщення людини та збереження історії. Забезпечено виведення звіту з графічним відображенням історії переміщення зараженої мітки та точках контактів з іншими мітками.

ANNOTATION

IoT system for monitoring and analyzing information about the presence of people in the room // Bachalor thesis // Onyskiv Roman // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Systems and Nets // Ternopil, 2023 // p.- 60, fig. – 31, table. – 4, Sheets A1 – 5, Ref. – 59.

Keywords LOCALIZATION, IPS, CLOUD COMPUTING, RADIO TRACKING, MICROCONTROLLER, MQTT, FLUTTER

The qualification work deals with the development of an IoT system that prevents users from possible contact with sick people based on the history of their positioning inside the room.

The main methods and technologies for indoor localization (trilateration, triangulation, radio footprints) were studied. The analysis and selection of parameters for localization was carried out. The selection of metrics required for calculating the coordinates of people inside the room was made. A study of existing IoT platforms was carried out. The system architecture is built. A review of existing data transfer protocols was performed. Considerable attention is also paid to the application of cloud computing using the capabilities of the Google Cloud platform.

Diagrams of the interaction of components and the functioning of the IoT system are built. Algorithms and their software implementation of Bluetooth air scanning and calculation of coordinates of radio tags are given. The contacts of human movement and preservation of history have been defined. A report is provided with a graphical display of the history of the infected tag's movement and points of contact with other tags.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ 7
ВСТУП 8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ 10
1.1 Основні методи локалізації 10
1.2 Параметри для локалізації в системах IPS 13
1.3 Технології бездротового зв'язку в системах IPS 16
1.3.1 Wi-Fi 16
1.3.2 BLE 17
1.3.3 ZigBee 18
1.3.4 RFID 18
1.3.5 UWB 19
РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА 20
2.1 Архітектура IoT систем 20
2.2 Мікроконтролери 21
2.2.1 PIC 21
2.2.2 AVR 22
2.2.3 STM 23
2.2.4 ESP8266 и ESP32 24
2.3 Плати відлагодження 26
2.3.1 Плати на основі мікроконтролера 26
2.3.2 Одноплатні комп'ютери 28
2.4 Протоколи передачі в IoT 29
2.5 Хмарні обчислення в IoT. GCP 30

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА 35
3.1 Google IoT Core. Створення MQTT -топіків. Збір даних 35
3.2 Збір значень RSSI з Bluetooth-маячків. Прошивка ESP32 38
3.3 Обчислення координат маячків 41
3.4 Знаходження перетинів з історії координат радіоміток 42
3.5 Клієнтська частина. Формування звіту 43
3.5.1 Flutter 44
3.5.2 Принцип побудови інтерфейсу 44
3.6 Схема взаємодії компонентів 45
3.7 Схема роботи системи 47
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ ... 49
4.1 Стихійні лиха та їх класифікація 49
4.2 Соціальне значення охорони праці. 51
ВИСНОВКИ 54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 55
ДОДАТКИ
Додаток А Технічне завдання

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

AOA (Angle Of Arrival) – кут прибуття сигналу.

BLE (Bluetooth Low Energy) – технологія цифрової бездротової передачі даних з наднизьким енергоспоживанням і малим радіусом передачі.

CSI (Channel State Information) – інформація про стан каналу.

GCP (Google Cloud Platform) —набір хмарних служб від Google.

GPS (Global Positioning System) – система глобального позиціювання.

IoT (Internet Of Things) – інтернет речей.

IPS (Indoor Positioning System) – система внутрішнього позиціювання.

MQTT (Message Queuing Telemetry Transport) – спрощений мережевий протокол.

OPS (Outdoor Positioning System) – система зовнішнього позиціювання.

OFDM (Orthogonal Frequency-Division Multiplexing) – мультиплексування з ортогональним частотним поділом каналів.

POA (Phase Of Arrival) – фаза прийнятого сигналу.

RFID (Radio Frequency Identification) – радіочастотна ідентифікація.

RSSI (Received Signal Strength Indication) — пристрій для вимірювання рівня потужності сигналу.

TDOA (Time Difference Of Arrival) – різниця часу прибуття сигналів.

TOA (Time Of Arrival) – оцінка часу прибуття сигналу.

UWB (Ultra-Wide Band) – надширока смуга.

Wi-Ti (Wireless Fidelity) – технологія бездротової локальної мережі.

ZigBee – бездротовий стандарт передачі даних.

БД – база даних.

МК – мікроконтролер.

ОС – операційна система

ПЗ – програмне забезпечення.

ВСТУП

В останні десятиліття стався швидкий розвиток бездротових систем геопозиціонування, особливо вони вплинули на IoT [1]. Такі сучасні системи активно використовуються не тільки для навігації, але і в різних сферах для визначення розташування клієнтів, пристроїв або речей, фінансового сектора, логістики або транспортного сектора [2].

На даний момент існує два види систем позиціонування – OPS та IPS.

Представниками OPS є, наприклад, GPS, Beidou. Такі системи дуже добре показують себе при позиціонуванні ззовні приміщення, проте не зовсім підходять для використання всередині з кількох причин, наприклад: відсутність прямої видимості супутників, перешкоди і шуми [3]. В даний час OPS в основному використовуються для навігації поза приміщеннями, де немає необхідності в більш точному позиціонуванні та похибка в кілька метрів є допустимою.

IPS застосовується в будівлях та закритих спорудах, в яких фактично недоступні супутникові навігаційні системи [4]. Вони дуже актуальні у світі. За прогнозами аналітиків, потенційний обсяг їх ринку у світі досягне 24 млрд доларів до 2027 року [5]. Вони можуть застосовуватися в різних сферах, наприклад: навігація для роботів, відстеження людей або об'єктів у реалі, моніторинг стану пацієнтів у охороні здоров'я, моніторинг складів, забезпечення безпеки підприємств і т. д. [6].

Пандемія COVID-19 також робить IPS дуже актуальними в наші дні. Так, карантин та інші примусові заходи призначені порятунку життів з допомогою стримування поширення вірусу. Як тільки такі обмеження знімаються, виникає необхідність перезапуску бізнес- процесів на очний режим. Це, у свою чергу, тягне за собою ризик зараження вірусом співробітників на робочих місцях, тим самим наражаючи на небезпеку життя і знову поміщаючи підприємства в карантин [5].

За допомогою відстеження розташування людей усередині приміщень

можна своєчасно знаходити контактних осіб та обмежувати їх переміщення вибірково, а не закривати все підприємство на карантин. Таким чином, можливе внесення контролю в процес поширення вірусу.

Можливість взаємодії досить дешевих давачів та пристроїв між собою, завдяки активному розвитку IoT технологій останнім часом [7], може допомогти у розробці такої непомітної та зручної для користувачів системи відстеження та попередження контактних осіб усередині приміщень.

Метою роботи є розробка системи, котра запобігає користувачів можливих контактів з хворими людьми за даними історії їх позиціонування всередині приміщення.

Завдання, необхідні для досягнення даної мети:

– вивчити основні методи та технології для локалізації у приміщенні;

– здійснити вибір метрик, необхідних розрахунку координат людей всередині приміщення;

– дослідити існуючі IoT платформи та вибрати конкретну для реалізації системи;

– забезпечити збереження історії переміщення людини;

– визначити контакти з історії переміщення та виведення результатів у вигляді звіту;

– реалізація прототипу системи.

Практична цінність роботи зумовлена можливістю своєчасного знаходження контактних осіб та подальшого вибіркового обмеження їхнього переміщення, що дозволить внести контроль у процес поширення вірусу.

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Основні методи локалізації

Основою всіх систем позиціонування всередині приміщень є методи локалізації, які можна поділити на дві основні категорії:

– з використанням діапазону передачі (range-based);

– без використання діапазону передачі (range-free) [8].

Range-based методи засновані на використанні значення відстані між маячком та базовими станціями для оцінки розташування. Їх перевага полягає у вищій точності в порівнянні з range-free методами. Представниками є трилатерація та тріангуляція.

У range-free методах, навпаки, розташування оцінюється без будь-яких вимірювань, а тільки шляхом визначення того, чи маячок і базова станція в межах досяжності [9]. Приклад такого методу є метод радіовідбитків. Зазвичай, системи, які використовують такі методи, коштують набагато дешевше, ніж системи, котрі базуються на Range-based методах.

Варто навести базові особливості основних підметодів, згаданих вище.

Метод трилатерації – популярний метод, який використовується для позиціонування. Ідея методу трилатерації показана на рис. 1.1, де (x_1, y_1), (x_2, y_2) і (x_3, y_3) - координати базових станцій 1, 2 і 3 відповідно. Параметри d_1, d_2, і d_3 — відстані між базовими станціями та маячком. Шукані координати маячка - (x_e, y_e) [10].

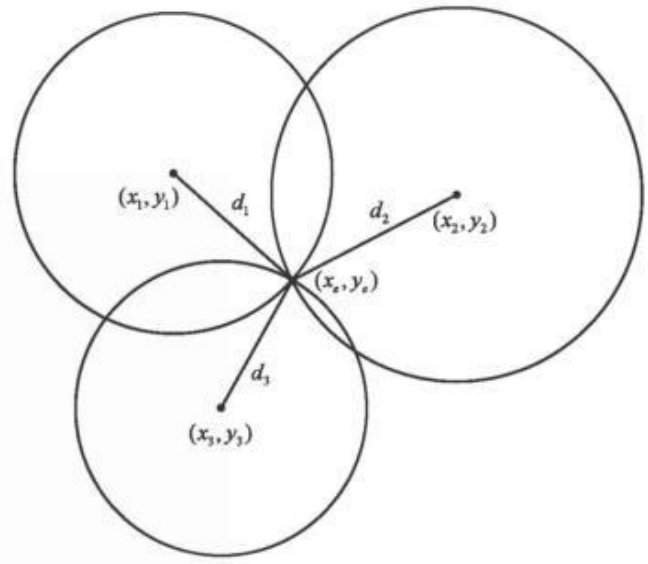

Рисунок 1.1 – Метод трилатерації

Координати маячка можуть бути отримані через розв’язок такої системи рівнянь (1.1):

$$\begin{aligned} &(x_e - x_1)^2 + (y_e - y_1)^2 = d_1^2, \\ &(x_e - x_2)^2 + (y_e - y_2)^2 = d_2^2, \\ &(x_e - x_3)^2 + (y_e - y_3)^2 = d_3^2, \end{aligned} \tag{1.1}$$

При триангуляції визначається напрямок від опорних вузлів до цільового і положення цільового вузла буде на перетині променів, проведених від опорних вузлів у напрямку цільового. Визначення розташування маячка відбувається шляхом вимірювання кутів сигналів, що прийшли від базових станцій до маячку [33].

Для локалізації цільового вузла достатньо двох вузлів опорних точок (рис. 1.2).

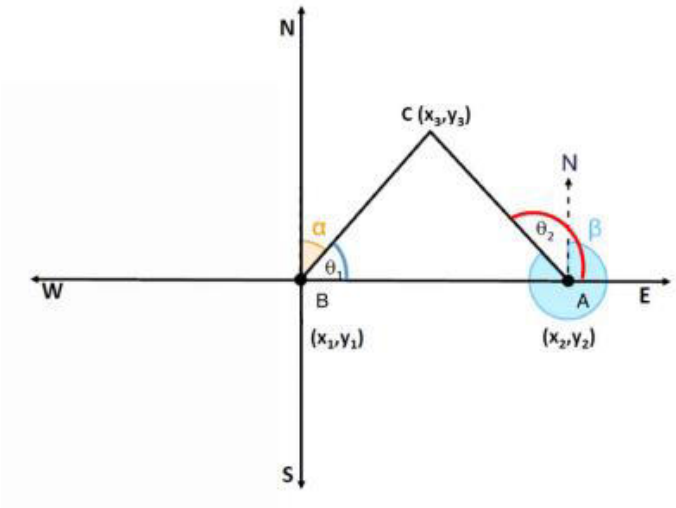

Рисунок 1.2 – Метод триангуляції

Знаючи кути θ_1, θ_2, координати (x_1, y_1) та (x_2, y_2) можна знайти координати шуканої точки (x_3, y_3) наступним чином:

$$\tan(\theta_1) = \frac{(y_3 - y_1)}{(x_3 - x_1)}, \quad (1.2)$$

$$\tan(\theta_2) = \frac{(y_3 - y_2)}{(x_3 - x_2)}, \quad (1.3)$$

Із формул (1.2) та (1.3) можна виразити x_3 та y_3 так:

$$x_3 = \frac{y_1 - y_2 + x_2 \tan\theta_2 - x_1 \tan\theta_1}{\tan\theta_2 - \tan\theta_1} \quad (1.4)$$

$$y_3 = \frac{y_1 \tan\theta_2 - y_2 \tan\theta_1 + (x_2 - x_1)\tan\theta_2 \tan\theta_1}{\tan\theta_2 - \tan\theta_1} \quad (1.5)$$

Метод радіовідбитків ґрунтується на даних про рівень сигналу - RSSI. В основі методу – створення так званого "відбитка" приміщення (карти його покриття).

Тобто приміщення розбивається на сектори, у кожному з яких вимірюються значення RSSI у радіомаячків, які заносяться до БД. Далі при

порівнянні значень RSSI з існуючими записами в БД, можна оцінити положення маячка в приміщенні (рис. 1.3).

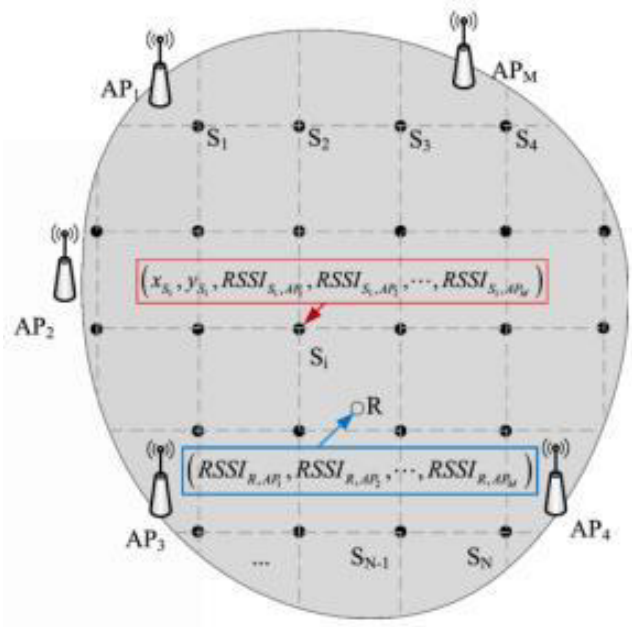

Рисунок 1.3 – Метод радіовідбитків

1.2 Параметри для локалізації в системах IPS

Необхідно розглянути основні параметри, які можна використовувати в системах позиціонування всередині приміщень. Такими параметрами є RSSI, TOA, AOA або TDOA, POA та CSI.

Їхня класифікація представлена на рис. 1.4.

RSSI є порівняльним виміром величини сигналу, котрий прийнятий. RSSI метрика, що часто застосовується для оцінки відстані між метою і базовою станцією, яка не вимагає складних обчислень. Відстань між двома вузлами обчислюється за втратою потужності сигналу. Для оцінки відстані достатньо лише пари вузлів [11], але для визначення координат вже потрібно використовувати метод трилатерації разом із системою з трьох базових станцій і маячка.

Виробники використовують різні шкали виміру рівня RSSI. Так, у WIFI чіпсетів компанії Atheros RSSI варіюється від 0 до 60 dBm, а Cisco від 0 до 100 dBm [12].

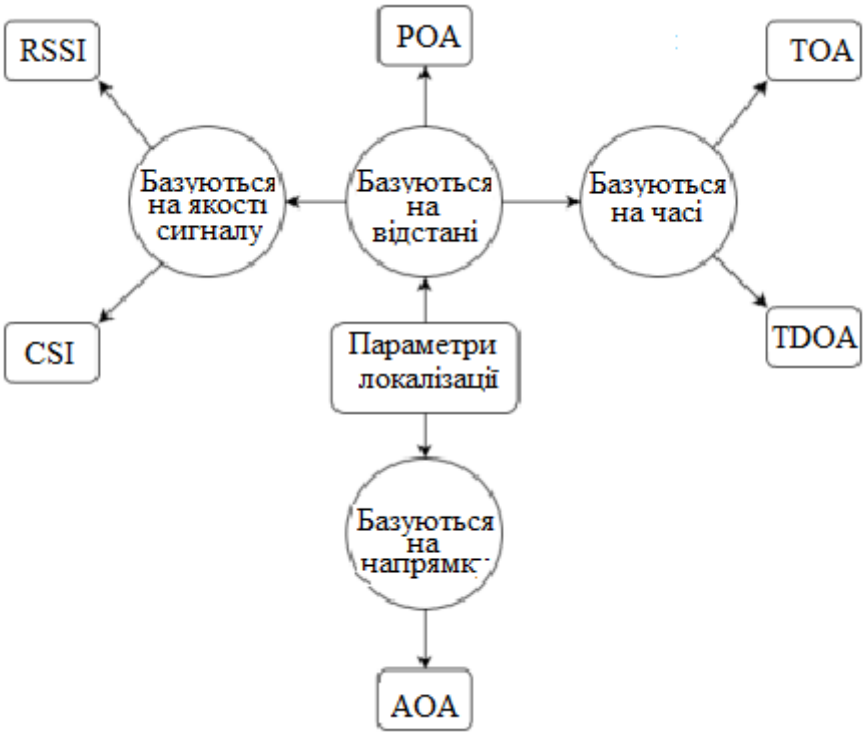

Рисунок 1.4 – Основні параметри для локалізації

TOA описується як перший період, упродовж якого сигнал досягає приймача. За його допомогою можна оцінити відстань до вузла, обчисливши час трансляції бездротового радіосигналу [13,14].

Для визначення місця розташування маячка потрібно використовувати мінімум 3 базові станції у відкритому полі зору для підвищення точності [15].

TDOA – ще один параметр, що дозволяє розрахувати інформацію про відстань між двома вузлами мережі. Суть полягає у вимірі різниці часу передачі сигналу від маячка до базової станції з відомими координатами та синхронізованим часом. Маючи цю різницю часу отримання сигналу можна розрахувати відстань від маячка до базових станцій [16].

Базові станції прослуховують пакети від маячка і його розташування може бути розраховане шляхом порівняння різниці між часом прибуття пакетів до всіх базових станцій [17].

AOA – визначення розташування за допомогою даного параметра базується на визначенні напрямку джерела сигналу. У цьому методі використовуються спеціальні базові станції, на яких встановлені антени, що крутяться, або антени з фазованими гратами або ж просто кілька антен, спрямованих в різні сторони. Після отримання напрямку сигналу можна визначити координати його джерела. Зі збільшенням кількості базових станцій зростає точність визначення координат. [16].

Складність та необхідність у спеціалізованому обладнанні є основними перешкодами для широкого поширення систем позиціонування на основі AOA [18].

POA – параметр який пов'язаний із затримкою поширення та відстанню через довжину хвилі та швидкість світла [19]. Підходи на основі POA використовують фазу або різницю фаз несучого сигналу для оцінки відстані між передавачем та приймачем. Існує ряд методів для оцінки відстані між передавачем та приймачем за допомогою POA [20].

З розвитком нових технологій у системах бездротового зв'язку, у мобільній передачі даних (LTE) та в системах Wi-Fi почали застосовувати метод OFDM. OFDM перетворює інформацію на кількох піднесучих в одній смузі. Згідно із стандартом IEEE 802.11, приймачеві сигналу потрібно знати значення CSI для декодування даних.

CSI - це частотна характеристика каналу кожної піднесучої в системі OFDM в частотному полі. CSI використовує вдесятеро більше даних, ніж традиційний RSSI, при комунікації між відправником і одержувачем [21].

Використання CSI показує хорошу стабільність і може забезпечити більш високу точність позиціонування, ніж методи на основі RSSI [14,21].

1.3 Технології бездротового зв'язку в системах IPS

У цьому підрозділі будуть розглянуті основні технології, які можуть бути використані у системах позиціонування усередині приміщень. Серед інших це Wi-Fi, ZigBee, RFID, BLE, UWB, LoRa, Sigfox, NFC.

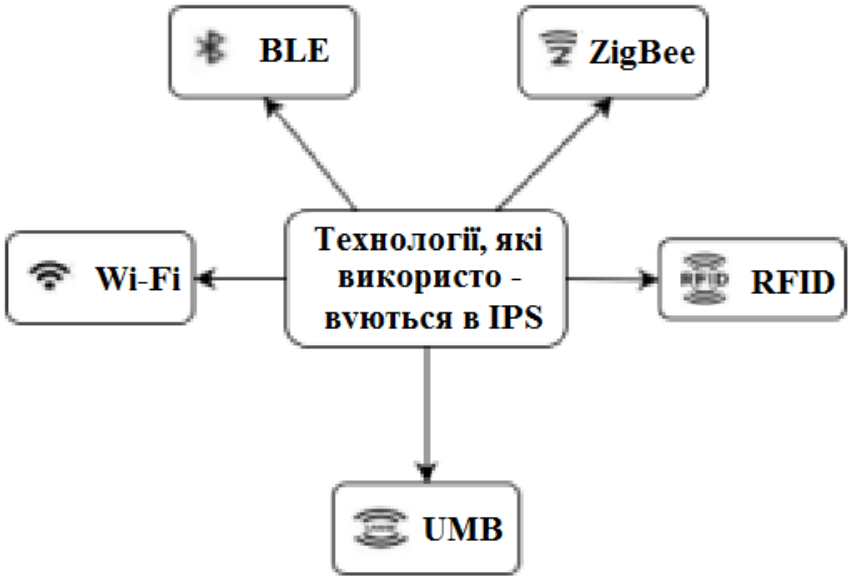

Рисунок 1.5 – Технології, які можуть бути застосовані для IPS

1.3.1 Wi-Fi

Є добре відомою технологією широкосмугового зв'язку, особливо для схем міжмашинної взаємодії [22]. Пристрої Wi Fi, зазвичай, передають сигнал на частоті 2,4 ГГц, проте тепер для передачі широко використовується і частота 5 ГГц через меншу кількість перешкод, і більш стабільного з'єднання на підвищеній швидкості [23].

Для локалізації за допомогою Wi-Fi застосовуються різні алгоритми, у тому числі алгоритм на основі AOA (триангуляція) [24], алгоритм трилатерації [25], алгоритм фінгерпринтінгу на основі RSSI [26] та на основі CSI [27].

Wi-Fi може здатися гарним вирішенням проблеми позиціонування всередині приміщень через повсюдне розміщення точок доступу, низьку вартість порівняно з іншими бездротовими технологіями, низького енергоспоживання та

відсутність додаткових вимог до обладнання [28,29], однак точність позиціонування становить у середньому 2 -3 метри, що залишає бажати кращого. Крім того, реалізація системи позиціонування всередині приміщення на основі Wi-Fi може сильно залежати від таких факторів довкілля, як географія приміщення, скупчення великої кількості людей, а також погоди [30].

1.3.2 BLE

BLE, заснований на стандарті IEEE 802.15, на сьогоднішній день підтримується майже всіма смарт-пристроями. Bluetooth 4.0 був представлений в 2010 [31] і працює на частотах від 2,4 ГГц до 2,4835 ГГц.

BLE розроблено для бездротової передачі даних на дуже короткі відстані. Таким чином, при використанні систем на основі BLE, точність позиціонування всередині приміщення, в порівнянні з Wi-Fi, зазвичай набагато вище. Радіус роботи BLE зазвичай становить не більше 10 метрів, що потребує мінімальних витрат на енергію та ідеально підходить для приміщень [32].

Хоча система позиціонування на основі Bluetooth вимагає додаткових апаратних пристроїв, на відміну від системи на основі Wi- Fi, але точність позиціонування при цьому може досягати 1 метра [29]. Крім того, BLE при розробці IoT рішень вибирають через такі переваги, як низька вартість, низьке енергоспоживання [34], невеликий розмір і простота розгортання [35, 36]. Радіус роботи систем на основі BLE можна збільшити аж до 100 м, регулюючи потужність мовлення, що створює можливість для ширшого діапазону позиціонування. У системах на базі Bluetooth в основному застосовуються методи трилатерації та фінгерпринтінгу.

У 2013 році компанія Apple Inc. представила нову технологію iBeacon Технологія iBeacon була створена на основі технології BLE, яка може передавати дані безпосередньо зі смартфонів, і має меншу потужність та вартість, ніж традиційні технології Bluetooth та Wi-Fi [34,37]. Крім того, запуск відкритого стандарту Google EddystoneTM у 2015 році призвів до появи нових та покращених форматів мовлення, які сприяли розвитку інтересу до поширення та

впровадження платформ радіомаяків Bluetooth [31].

Все вище перелічене робить технологію BLE дуже цікавою для застосування в системах позиціонування всередині приміщень.

1.3.3 ZigBee

Це технологія з невеликим радіусом дії, її основі стандарт IEEE 802.15.4. Володіє низькою швидкістю передачі даних, робоча частота 2,4 ГГц,.

ZigBee класифікує три типи пристроїв: координатор, маршрутизатор та кінцевий пристрій. Останній дешевше у виробництві, ніж координатор чи маршрутизатор [11]. Технологія ZigBee має трохи більший радіус роботи, ніж технологія BLE, тому вона здатна передавати дані далі, використовуючи мережу ретрансляційних вузлів [34].

Однак, у порівнянні з BLE, ZigBee має трохи більш високе енергоспоживання [34] і технологія не набула такого сильного поширення в IoT рішеннях. Крім того, тому що використовується пориста топологія мережі, то вузли, що маршрутизують повідомлення, завжди повинні бути активними. Кінцеві пристрої з хоч і є складовою мережі, не можуть її продовжувати, тому що не передають дані від інших вузлів [38].

1.3.4 RFID

Це бездротова безконтактна технологія, яка забезпечує автоматичну ідентифікацію шляхом передавання інформації з RFID -мітки на зчитувач із застосуванням електромагнітного сигналу. Як правило, до складу технології RFID входять зчитувальний пристрій, мітки та обчислювальний пристрій [39].

RFID базуються на технології активних та пасивних міток [40, 41]. Перші володіють більшим діапазоном виявлення за рахунок високого енергоспоживання та вищої вартості. Другі застосовуються для локалізації статичних точок на коротких відстанях і застосовні тільки для невеликого простору, хоч і є більш поширеними.

Недолік RFID полягає в тому, що він не зовсім застосовний для визначення

розташування в реальному часі при використанні пасивних міток, тоді як активні мітки RFID занадто енерговитратні. Крім того, через дифракцію і відображення передача сигналу RFID в приміщенні є утрудненою і сильно схильною до шумів [42].

1.3.5 UWB

Приваблива технологія, що швидко розвивається в бездротових сенсорних мережах, яка забезпечує високі швидкості передачі для невеликих відстаней через свою широку смугу пропускання, що також забезпечує більш високу точність і, отже , точніше позиціонування кожного цільового пристрою в мережі [43]. Передача сигналів провадиться в діапазоні від 3,1 до 10,6 ГГц.

Через природу великої смуги пропускання UWB -сигнали забезпечують найкращий захист від перешкод. Крім того, UWB має менший вплив на людський організм через малу потужність передачі [44].

Основним недоліком даної технології є її слабка поширеність, з чого випливає висока ціна компонентів для її застосування в системах позиціонування. Також, ще одним недоліком є занадто широкий спектр частот, на якому вона працює: через це UWB може створювати перешкоди для інших систем, які працюють на цих частотах [45].

РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА

Так як IoT нерозривно пов'язаний з технологіями розташування всередині приміщень [46], особливо важливо розібратися в його основних компонентах. У цьому розділі буде розібрано наступні аспекти IoT систем: архітектура, МК та налагоджувальні плати, а також протоколи передачі даних.

2.1 Архітектура IoT систем

Технологія IoT дає можливість втілити різні ідеї в життя, багато компаній намагаються впровадити IoT рішення у свої бізнес-процеси, але на практиці виявляється, що досить важко реалізувати систему з безлічі пристроїв та змусити її працювати. Тому важливо дотримуватись певної архітектури при реалізації IoT рішення.

Насправді архітектура IoT рішень зазвичай зводиться до чотирьох рівнів (рис.. 2.1) [47].

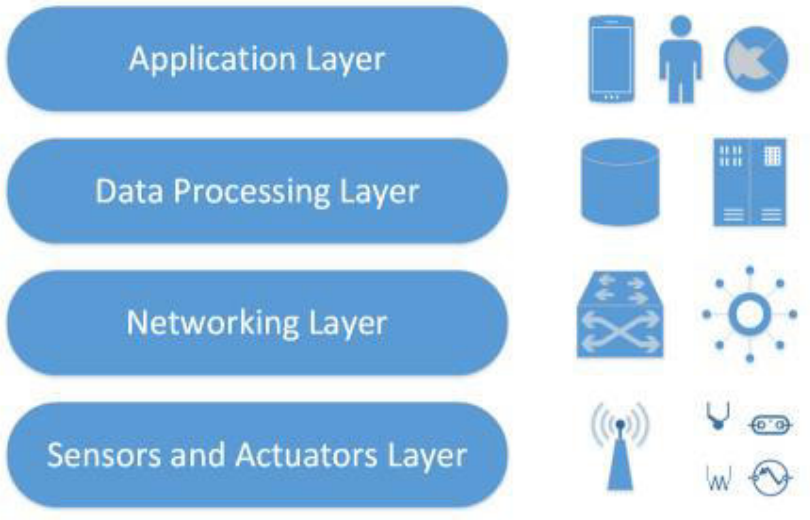

Рисунок 2.1 – Архітектура IoT рішень

На найнижчому рівні (рівень пристроїв - hardware level) архітектури знаходяться різні давачі та сенсори, а також МК.

Після збирання даних пристроями та їх початкової обробки необхідно передати їх на сервер. І тому існують різні технології зв'язку (LoRa, NB-IoT, GSM, WI-FI тощо.). Це відбувається на мережному рівні.

На наступному рівні (рівень обробки даних) відбувається робота з даними на сервері. Виконується певна логіка та ведеться спілкування з рівнем пристроїв, а також із прикладним рівнем.

Сервер може бути виділеною машиною, так і бути хмарним рішенням, яке надає нам послугу за певну плату (SaaS).

2.2 Мікроконтролери

У IoT рішеннях часто застосовуються МК, а розробка та налагодження таких рішень простіше проводиться на спеціальних платах налагодження.

МК прямо чи опосередковано впливають на наше повсякденне життя. Сьогодні вони застосовуються майже у всій електроніці, що вбудовується: пральні машини, кухонна техніка, заправні станції, паркувальні апарати, ліфти і т. д. Саме МК роблять всю цю техніку і пристрої "розумними" [48].

На сьогоднішній день у світі популярні 4 основні типи МК: PIC, AVR, ARM та ESP [49].

2.2.1 PIC

Зовнішній вигляд МК PIC наведений на рис. 2.2

Головні параметри, якими розрізняють їх моделі - це розрядність, набір інтерфейсів і форм-фактор чіпа. Восьмирозрядні МК PIC поділяються на чотири групи: PIC18, enhanced mid-range, mid-range, baseline. На рис. 2.3 далі наведено докладну інформацію.

Рисунок 2.2 – Приклад PIC-мікроконтролера

	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 134B	Up to 368B	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/I²C™ · 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
Total Number of Devices	16	58	29	193
Families	PIC10, PIC12, PIC16	PIC12, PIC16	PIC12F1XXX, PIC16F1XXX	PIC18

Рисунок 2.3 – Характеристики МК PIC

Програмують ці МК, як правило, на асемблері та на С. Існує безліч середовищ розробки: MPASM та MPLAB, MicroC, JALedit.

Зазвичай на даних МК збирають прості пристрої, наприклад мигалки або таймери.

2.2.2 AVR

МК такого типу (рис. 2.4) виробляє компанія Atmel. Це МК, з яких збирають Arduino. Дані МК поділяються три основні групи: : tinyAVR, megaAVR, XmegaAVR. Їх основні характеристики наведені в табл. 2.1.

Рисунок 2.4 – Зовнішній вигляд МК AVR

Таблиця 2.1 – Порівняння характеристик МК AVR

	Flash -пам'ять	RAM	ROM	Число GPIO пінів
TinyAVR	16 Кбайт	512 Байт	512 Байт	4-18
MegaAVR	16 Кбайт	16 Кбайт	4 Кбайт	23-86
XmegaAVR	384 Кбайт	32 Кбайт	4 Кбайт	44-100

Завдяки поширенню плат прототипування Arduino, з цими МК легко розпочати роботу новачкам. Вони можуть програмувати їх як на Assembler, і на C / C++. Також можна використовувати різні генератори коду, наприклад Scratch, де логіка описується у вигляді схем. Для роботи є різні IDE: Atmel Studio, IAR AVR, WinAVR, Arduino IDE Для прошивки слід використовувати спеціальні програматори. На ринку існують різні версії за різними цінами. Деякі плати розробки вже мають вбудовані програматори на борту.

2.2.3 STM

Ці МК поділяються на дві серії - STM8 та STM32. Як можна здогадатися з назв, такі чіпи бувають 8- та 32-бітові.

STM8 – функціональний аналог AVR, тільки дешевше. Існують три серії: STM8L з ультранизьким енергоспоживанням, STM8S для індустріальної апаратури та STM8A, іменовані «високонадійними». Периферія у всіх така сама, як у AVR. З позитивних сторін цих МК – низьке енергоспоживання та низька ціна. STM8 використовує свою архітектуру.

STM32 – старший представник мікроконтролерів STM. Його характеристики набагато вищі і коливаються у великих межах залежно від серії. Програмуються практично будь-якою мовою, навіть є варіанти програмувати їх на JavaScript.

STM32 базується на архітектурі ARM. Варто зазначити, що компанія ARM не виробляє МК, а розробляє лише архітектуру. Ліцензію на неї купують кінцеві виробники та використовують так, як їм заманеться. Найбільшого поширення якраз і набули чіпи компанії STMicroelectronics.

Під даними МК існують ОС. Найбільш популярною з них є ARM Mbed OS [50].

2.2.4 ESP8266 і ESP32

Починаючи з 2014 року на ринку МК став набирати популярності ESP8266 китайської компанії Espressif.

ESP8266 є продовженням розвитку МК. На відміну від них системи на кристалі можуть мати вже вбудовані інтерфейси бездротового зв'язку Bluetooth чи Wi-Fi і, як правило, бувають набагато потужнішими за МК, зберігаючи при цьому порівняні розміри.

На борту ESP8266 знаходиться 32-бітовий 80 МГц процесор Tensilica L106, 32 Кбайт IRAM (пам'ять для програми), 80 Кбайт DRAM, 16 контактів GPIO, а також безліч різних периферійних пристроїв, включаючи SPI, I2C, UART та ADC. Найголовніша перевага ESP8266 – вбудований Wi-Fi, проте Bluetooth відсутня [51].

ESP8266 можна програмувати в середовищі Arduino IDE як автономний МК, також можна використовувати його як Wi-Fi модуль, керований AT - командами через інтерфейс UART з іншого МК.

Сам по собі ESP8266 є лише чіпом, без необхідного для його роботи наповнення. Тому найчастіше використовуються плати-модулі, які поєднують у собі сам чіп, вбудовану flash - пам'ять, антену та інші необхідні компоненти. Приклад популярного модуля ESP-01 з чіпом ESP8266 показаний на рис. 2.5.

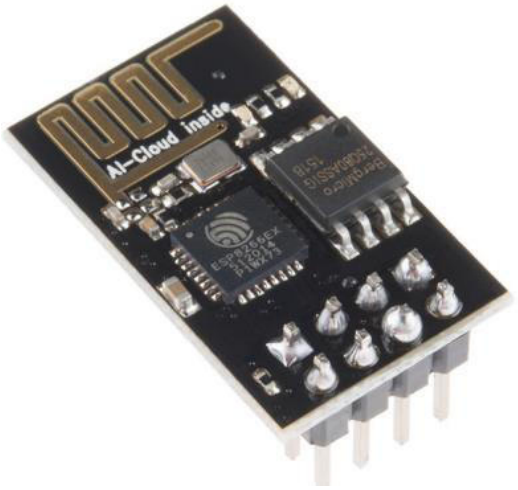

Рисунок 2.5 – Приклад модуля з чипом ESP8266 – ESP-01

ESP32 – це оновлена версія ESP8266. Ядром ESP32 є двоядерний 32-розрядний процесор Tensilica LX106 із частотою 160 МГц (або 240 МГц залежно від моделі) із співпроцесором із наднизьким енергоспоживанням. Він обладнаний 520 Кбайт RAM, декількома роз'ємами I/O, включаючи DAC, ACD. Також на борту є інтерфейси GPIO, I2C, SPI, I2S, UART, CAN [52].

ESP32, на відміну від ESP8266, також має вбудований Bluetooth v4.2, який розширює можливості підключення та підтримує режим BLE (Bluetooth із низьким енергоспоживанням).

У ESP32 інтегрована апаратна безпека у вигляді криптографічного прискорення та flash - носія з апаратним шифруванням, що робить ESP32 безпечною платформою для IoT проектів.

Зовнішній вигляд модуля показано на рис. 2.6

Рисунок 2.6 – Приклад модуля ESP32 DevKit v1

В загальному ESP32 виявляється краще за всіма параметрами ніж ESP8266. Порівняльна характеристика цих чіпів наведена в табл. 2.2.

Таблиця 2.2 – Порівняння параметрів ESP32 и ESP8266

	ESP8266	ESP32
Процесор	80 MHz 32-bit RISC	dual-core 240 MHz 32-Bit RISC
Пам'ять	32 Кбайт + 80 Кбайт	520 Кбайт
Інтерфейси	16 GPIO, UART, I2C, I2S, ADC	> 16 GPIO, UART, I2C, I2S, ADC, DAC, CAN
Bluetooth	Ні	Так
Wi-Fi	Так	Так
Апаратне шифрування	Ні	Так
Підтримка flash- пам'яті	До 16 Мбайт	До 16 Мбайт

2.3 Плати відлагодження

Є друкованими платами із встановленим на них МК з усім необхідним йому стандартним обв'язуванням. Також як відлагоджувальна плата можуть виступати одноплатні комп'ютери.

На плати також поміщають:

– схеми зв'язку з комп'ютером,

– розводку для підключення плат розширення,

– макетну область для монтування схем користувача.

Такі плати є зручними для виконання швидкого прототипування та «обкатування» вузлів, перевірки ідей та методів [53].

2.3.1 Плати на основі мікроконтролера

На рис. 2.7 наведено приклад такої плати на МК AVR

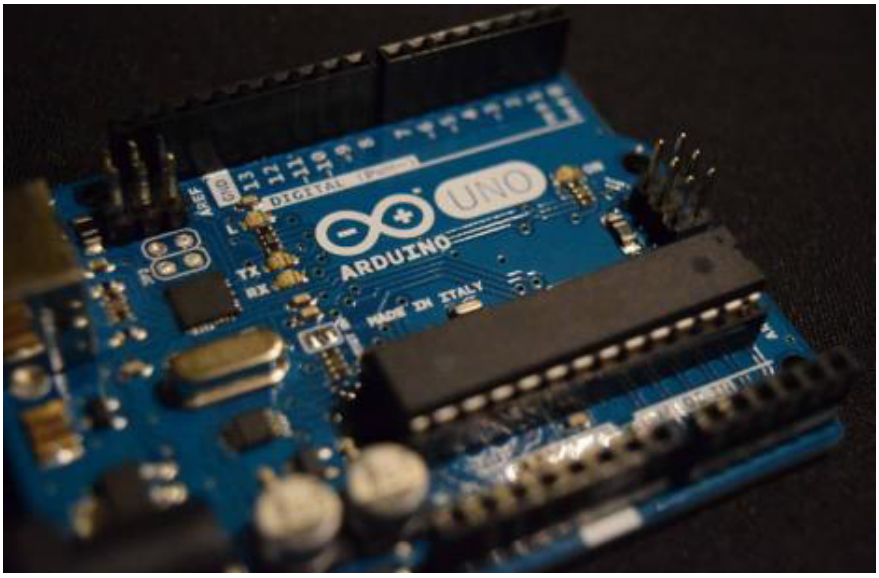

Рисунок 2.7 – Плата відлагодження на МК AVR

В табл. 2.3 для порівняння наведено кілька моделей налагоджувальних плат на основі МК.

Таблиця 2.3 – Порівняльні характеристики налагоджувальних плат на основі МК

Характеристика	Модель		
	Arduino UNO	Particle Electron	ESP8266-01
Піни GPIO	6 аналогових, 14 цифрових	12 аналогових, 30 цифрових	2 цифрових, 1 аналоговий
Логічний рівень	5В	3.3В	3.3В
Мікроконтролер	ATMega328P	32-bit STM32F205 ARM Cortex M3	32-bit Tensilica L106
Частота	16 MHz	120 MHz	80 MHz
Пам'ять	32 kB flash, 1 kB EEPROM	1 Mb flash, 128 kB RAM	1 Mb flash, 768 kB RAM
Мережеві інтерфейси	За промовчанням немає	Вбудований 2G/3G модем	Вбудований wi-fi, Bluetooth
Ціна	20 $	39$	10 $

2.3.2 Одноплатні комп'ютери

Такий пристрій є повноцінним комп'ютером. На одній друкованій платі у нього зібрано мікропроцесор, ОЗП, I/O системи та інші периферійні пристрої для функціонування комп'ютера.

Вони виробляються різних форм-факторів, наприклад для розробників. Такі версії мають більш зручний доступ до портів. Щоб не доводилося використовувати паяльник для розробки прототипу пристрою.

Також, існують промислові версії, вже для фінальної інтеграції у пристрої. Вони, як правило, мають менший розмір, для прикладу Raspberry Pi 4 (рис. 2.8).

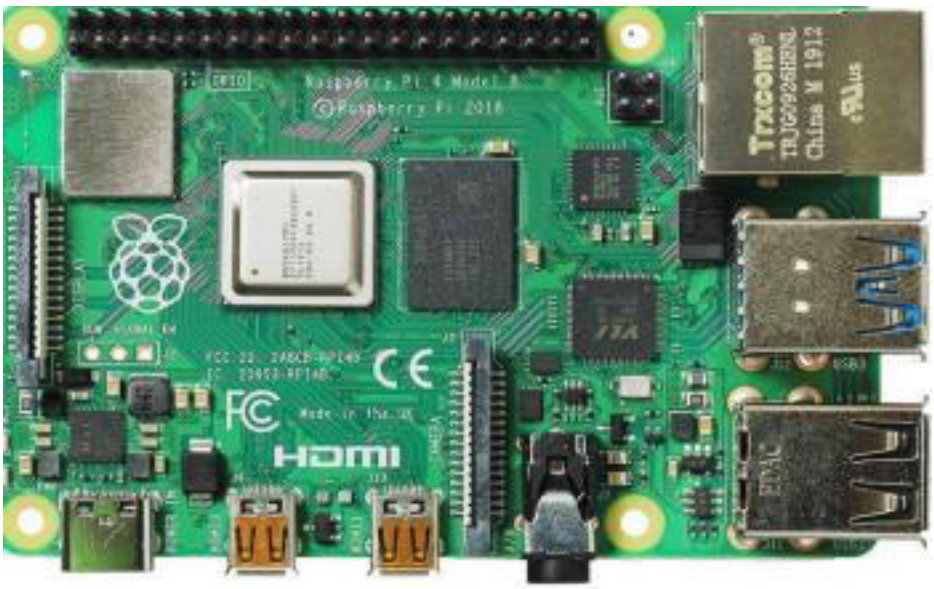

Рисунок 2.8 – Raspberry Pi 4

Для порівняння в табл. 2.4 наведено параметри кількох моделей одноплатних комп'ютерів.

Серед сильних їх сторін можна виділити високу обчислювальну потужність, велику кількість портів GPIO, багато вбудованих мережевих інтерфейсів, проте є й слабкі сторони: порівняно з платами на МК – це висока ціна та підвищене енергоспоживання.

Таблиця 2.4 – Порівняння параметрів одноплатних комп'ютерів

Характеристика	Модель		
	Raspberry Pi 4	BeagleBone Black	Qualcomm DragonBoard 410c
Піни GPIO	40 цифрових, 11 аналогових	65 цифрових, 7 аналогових	12 цифрових
Логічний рівень	3.3V	5V	1.8V
Процесів	ARM Cortex A72	AM335X ARM Cortex A8	ARM Cortex A53
Частота	1.5 GHz	1 GHz	1.2 GHz
Пам'ять	1-4GB flash, 1GB RAM	4 GB flash, 512 MB RAM	8 GB flash, 1GB RAM
Мережеві інтерфейси	Wifi, Ethernet, Bluetooth	Ethernet, зовнішні Wifi, Bluetooth	Wifi, Bluetooth, GPS
Ціна	35$	55 $	75$

2.4 Протоколи передачі даних в IoT

У світі IoT систем використовують безліч різних протоколів: DDS, XMPP, CoAP, MQTT, KMS, AMQP та інші. Також можуть використовуватися стандартні протоколи, наприклад HTTP. Конкретний протокол вибирається залежно від поставленої задачі та пристрою [54].

На автономних пристроях, які працюють кілька років без заміни акумулятора, використовують протоколи, що дозволяють відправляти дані з мінімальними енерговитратами, такі як MQTT. Якщо потрібна двостороння комунікація з пристроями, краще підходять протоколи, що дозволяють обладнанню здійснювати обмін повідомленнями в реалі, один з них – DDS.

DDS використовують для систем реального часу. Принцип роботи побудовано на обміні повідомленнями напряму.

Протоколи AMQP, JMS, XMPP та MQTT мають інший принцип роботи. Повідомлення передаються не безпосередньо, а через сервер-брокер, який перебирає всю обробку повідомлень. Він може бути розгорнутий на сервері в

дата-центрі або у хмарі.

З протоколів такого типу популярність набирає MQTT, оскільки він розроблений спеціально для інтернету речей, не вимагає постійного та стабільного з'єднання та не перевантажує канали зв'язку. Спрощена схема його роботи наведена на рис. 2.9.

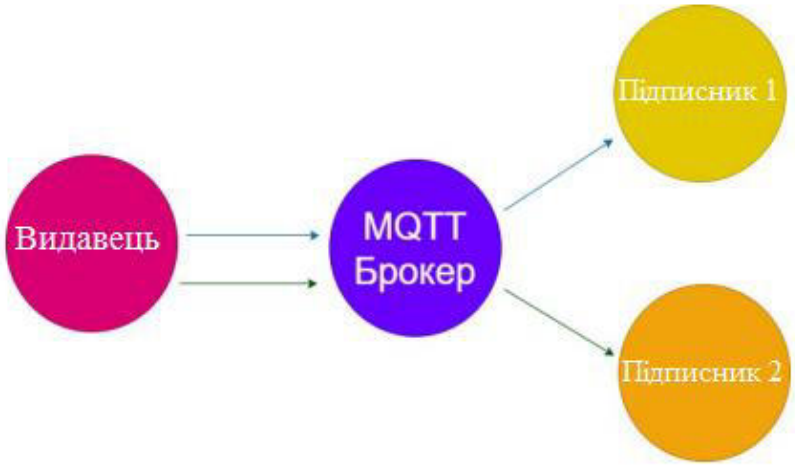

Рисунок 2.9 – Схема роботи протоколу MQTT

Протокол CoAP - аналог HTTP, але призначений спеціально для IoT-обладнання, оскільки відрізняється малим розміром заголовків, що підходить для мереж з обмеженими можливостями.

Принцип роботи простий: він спрямований на взаємодію точка- точка (клієнт-сервер). Клієнт звертається до сервера і надсилає йому прості команди, як це відбувається і в HTTP.

2.5 Хмарні обчислення в IoT. GCP

Застосування хмар дозволяє компаніям зберігати дані та керувати ними через хмарні платформи, забезпечуючи масштабованість при “доставці” ПЗ як послуги. Хмарні обчислення також дозволяють передавати та зберігати дані через інтернет або за прямим посиланням, що забезпечує постійне передавання даних між пристроями, додатками та хмарою.

У сфері IoT звичайна ситуація, коли генерується безліч даних за секунду. Хмари дають змогу зберігати та аналізувати ці дані, щоб бізнес міг отримати максимальну вигоду від інфраструктури IoT. Рішення IoT має поєднувати та забезпечувати зв'язок між речами, людьми та процесами, і хмарні обчислення є надзвичайно важливими у цьому процесі [55].

IoT не обмежується лише функціями збору даних, зберігання та аналітики. Концепція допомагає модернізувати різні процеси за допомогою підключення різних пристроїв та обладнання, включаючи застаріле, до Інтернету, що зменшує бар'єри між IT та OT командами завдяки єдиному уявленню про системи та дані.

Завдяки хмарам організаціям не потрібно розгортати складне обладнання, налаштовувати мережі та інфраструктуру для IoT рішень. Хмарні обчислення також дозволяють бізнесу масштабувати інфраструктуру в залежності від своїх потреб без встановлення додаткового обладнання та інфраструктури. Це не тільки допомагає прискорити процес розробки, а може скоротити витрати. Бізнесу немає необхідності витрачати гроші на купівлю та налаштування серверів та іншої інфраструктури, оскільки вони платять лише за ресурси, що споживаються.

Одним із представників сервісів для хмарних обчислень є GCP. До нього входить понад 40 інструментів конфігурації серверів, управління даними та інфраструктурою та інших сервісів для розробників (рис. 2.10). GCP дотримується вимог до безпеки та зберігання особистих даних користувачів.

Сервіси, що входять до складу платформи, та дані, що зберігаються, розміщуються на серверах Google. Налаштування сервісів, що використовуються, проводиться в консолі управління. Через неї запускаються віртуальні машини, створюються БД та здійснюється підключення до інших сервісів. Також налаштовувати, керувати та моніторити стан хмарних служб можна через спеціальний інтерфейс –CLI або мобільний додаток.

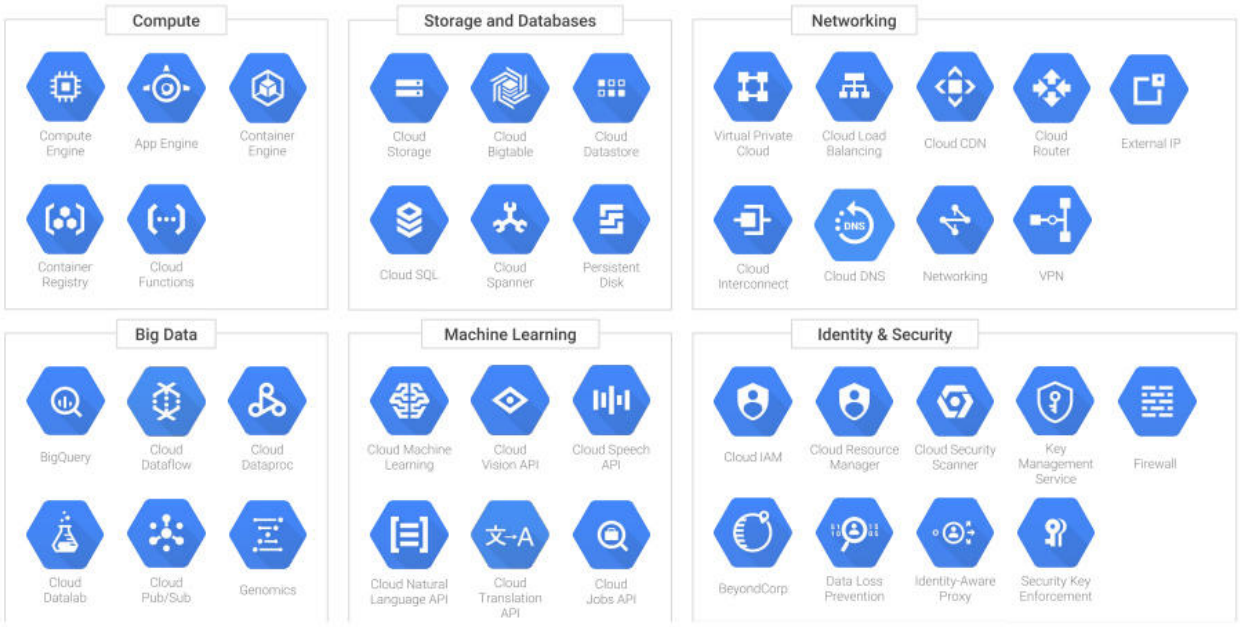

Рисунок 2.10 – Структура сервісів GCP сьогодні

В Google - хмарі можуть працювати як фізичні, так і юридичні особи. Після створення аккаунту необхідно прив'язати банківську картку для оплати послуг та пройти ідентифікацію. Після прив'язування картки новим користувачам первісний грант на 60 днів, який допоможе ознайомитися з можливостями сервісу та покрити первісні витрати.

До платформи входять, окрім інших, і такі категорії сервісів:

– мережа та інфраструктура – інструменти для масштабування інфраструктури;

– платформи даних - все, що потрібно для обробки, зберігання та візуалізації даних;

– бізнес-інструменти - інструменти для автоматизації бізнес- процесів команд. Вони містять послуги для управління проектами та зберігання документів;

– контейнерна розробка - все необхідне розробки мікросервісів і контейнерних додатків;

– безсерверні обчислення - інструменти для зберігання та створення програм без використання віртуальних машин;

– безпека - захист від кібератак та інструменти, що забезпечують

збереження даних;

– ML – інструменти для роботи з машинним навчанням.

Платформа призначена для надання єдиної множини апаратних ресурсів. Вона немає виділених фізичних серверів управління. Більшість сервісів розгорнуто на віртуальних машинах, аналогічних машинам кінцевих користувачів. Такий підхід забезпечує просту масштабованість платформи: з мінімальними витратами можна автоматично додавати потужності інфраструктури від одного сервера до цілого центру обробки даних, на якому працюють тисячі машин [56].

На рис. 2.11 показано архітектуру GCP.

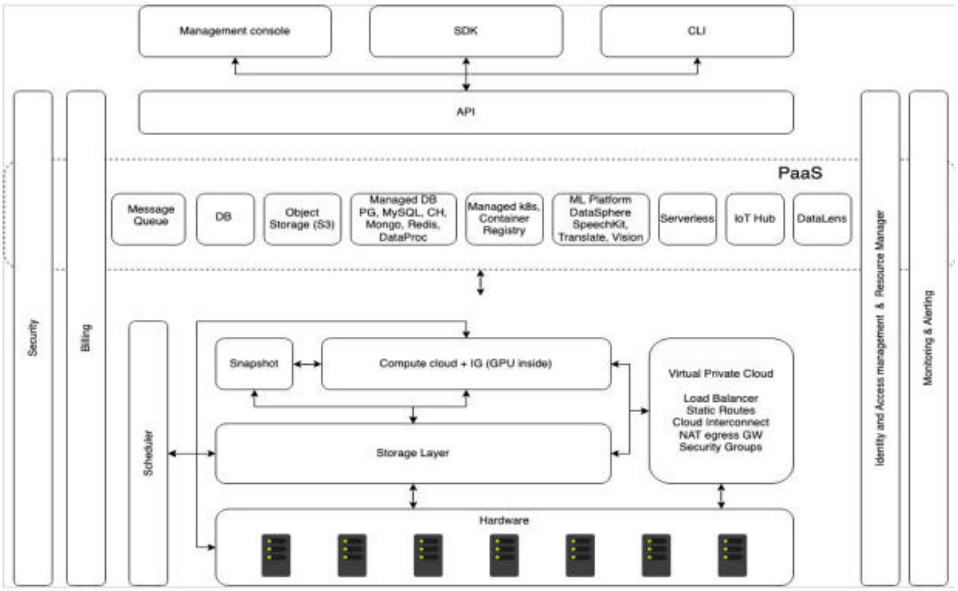

Рисунок 2.11 – Архітектура GCP

Основою платформи є storage layer – системне сховище даних. Воно працює на основі пропрієтарної системи керування БД. Завдяки ньому можна ефективно розміщувати дані платформи, що дозволяє горизонтально масштабувати систему з її зростанням. За допомогою механізму реплікації охоплюються всі апаратні сховища у хмарі.

Вище storage layer йде compute cloud – компонент, який керує віртуальними машинами зовнішніх користувачів та самої системи. Віртуалізація забезпечується за допомогою KVM. VirtIO використовується для віртуалізації

пристроїв.

Одним із найважливіших компонентів є планувальник (scheduler). Він визначає, на якому фізичному сервері буде розгорнуто віртуальну машину. Цей шар також включає інші служби інфраструктури.

На рівні вище розташовані платформні сервіси (PaaS). Вони включають, наприклад, IoT hub, DataLens, Object Storage і так далі.

У правій частині на схемі видно IAM (Identity and Access Management) та ресурс manager. Вони дозволяють активувати доступ на основі ролей кожного ресурсу. Кожен користувач має роль, яка дає доступ до певних операцій.

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

У цьому розділі будуть розглянуті етапи реалізації різних частин системи:

– збір метрик для визначення розташування Bluetooth маячків у приміщенні з використанням МК ESP32,

– подальше відправлення цих даних у хмару Google Cloud для розрахунку координат,

– аналіз результатів історії переміщення радіомаячків для знаходження потенційно контактних осіб.

Крім того, буде описано реалізацію клієнтської частини для відображення звіту про контакти у вигляді кросплатформної програми на фреймворку Flutter.

3.1 Google IoT core. Створення MQTT -топіків. Збір даних

Будь-яке IoT рішення, як правило, зводиться до двох етапів:

– передача даних у хмару, та потім у сховище. Це завдання вирішується при побудові будь-яких IoT систем. Іноді потрібно надсилати команди назад до пристроїв, але не у випадку системи, що розробляється;

– аналіз ухвалених даних. Наприклад, визначення будь-яких показників, побудова висновків із них чи навіть візуалізація цих даних.

Перший етап приблизно однаковий для всіх рішень. Він складається за подібними принципами для всіс систем. Друга частина вже унікальна для кожного розв'язуваного завдання. Однак, якщо ми не маємо самих даних, то й аналізувати нічого. Тому важливо побудувати досить стійку від відмови і надійну частину системи, яка взаємозв’язана із апаратною складовою для збору даних і, при необхідності, може масштабуватися.

Google Cloud вже пропонує практично готове рішення для цієї мети.

На рис. 3.1 зображено схему збору даних, запропоновану Google Cloud.

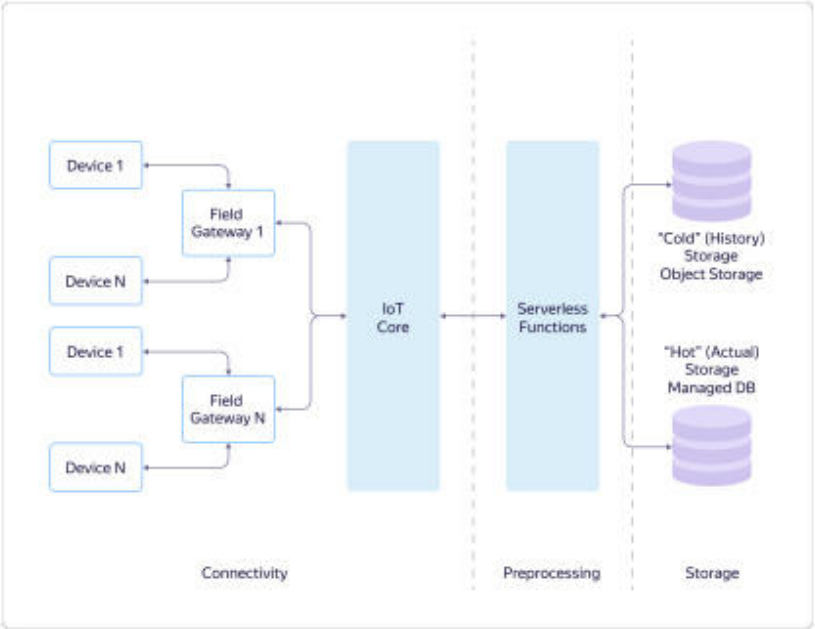

Рисунок 3.1 – Схема збору даних з IoT пристроїв за допомогою технологій Google Cloud

IoT Core - це стійкий до відмов і надійний MQTT брокер з можливістю масштабування.

Google Cloud Functions дозволяє запускати код у вигляді функцій у виділеному оточенні, яке безвідмовне, безпечно та масштабується автоматично, при цьому немає потреби обслуговувати віртуальні машини. Такий підхід зараз дуже перспективний і називається serverless. За допомогою таких функцій можна перехопити дані з IoT core, обробити їх та передати до БД.

Для зберігання даних Google пропонує безліч рішень як своїх, наприклад – Google Cloud Storage, так і сторонніх – PostgreSQL

У розроблюваній системі відстеження розташування всередині приміщення було прийнято рішення передавати дані для визначення координат Bluetooth -маячків через протокол MQTT в Google IoT Core.

MQTT став стандартом передачі в IoT. Він використовує поняття названих черг (топиків), куди з пристрою можна записувати дані, а з іншого боку, наприклад з сервера - асинхронно їх отримувати, за допомогою підписки на події

в цій черзі повідомлень.

Для кожного облікового запису в Google Cloud передбачена можливість створювати кілька реєстрів, у кожен з яких може включатися безліч пристроїв. Схема доступу до топиків в IoT core наведена на рис. 3.2:

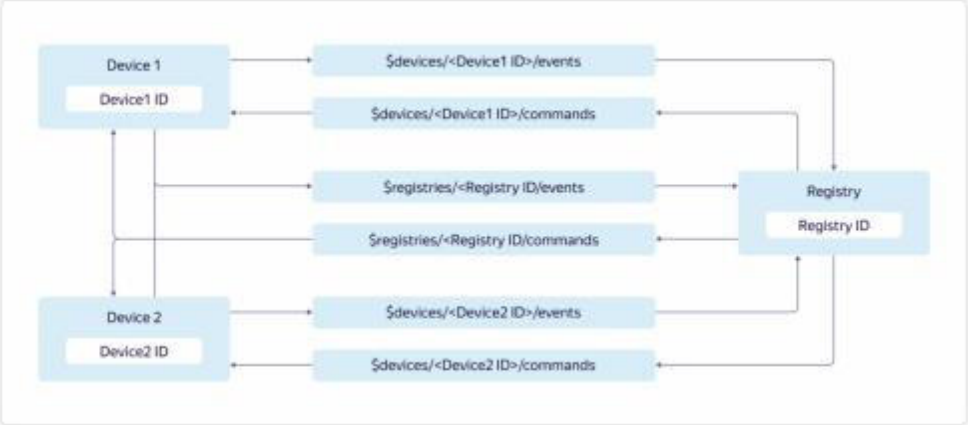

Рисунок 3.2 – Доступ пристроїв до MQTT -топиків в IoT Core

Пристрої мають право писати дані лише у свої event-топики, а також у загальний event-топік біля реєстру: $devices/ <Device1 ID> /events $registries/ <Registry ID> /events.

Пристрої також можуть підписуватись на повідомлення у своїх command-топіках, а також на повідомлення в загальному command-топіку біля реєстру: devices/ <Device1 ID> /commands $registries/ <Registry ID> / commands.

Реєстр, у свою чергу, може писати дані у всіх command-топік на пристроях і в свій command-топік: $devices/ <Device1 ID> /commands $devices/ <Device2 ID> /commands $registries/ <Registry ID> / commands.

Реєстр також може підписатися на повідомлення у всіх event-топіках пристроїв та у своєму власному event-топіку: $devices/ <Device1 ID> / events $devices/ <Device2 ID> / events $registries/ <Registry ID> / events.

У випадку системи, що розробляється все, що потрібно для збору даних:

– створити реєстр;

– зареєструвати три пристрої у ньому.

Кожен пристрій у реєстрі представлятиме базову станцію, з якої

здійснюється збір даних. Базові станції надсилатимуть дані до загального топіка подій біля реєстру: $ registries / <ID_створеного_реєстру> / events.

Зворотній зв'язок з базовими станціями не передбачається, тому немає необхідності використовувати топіки самих пристроїв, або підписуватися на топіки реєстру з самих базових станцій.

3.2 Збір значень RSSI з Bluetooth-маячків. Прошивка ESP32

Як апаратна платформа для базових станцій в системі, що розробляється, використовується МК ESP32, про який було написано докладніше в попередньому розділі.

Для базових станцій був розроблений прототип прошивки мовою C++, який дозволяє сканувати ефір Bluetooth -маячків, зчитувати їх RSSI та передавати ці дані за протоколом MQTT з використанням SSL -шифрування в Google Cloud IoT Core. Підключення до мережі інтернет здійснюється через WI-FI.

На рис. 3.3 наведена елементна база системи.

Рисунок 3.3 – Фото МК ESP32 та iBeacon Bluetooth маячків

Всі необхідні інтерфейси вже вбудовані в ESP32, а для розробки прошивки був використаний Arduino Core, який офіційно поставляється компанією Espressif, що полегшує розробку ПЗ для їх МК, тому що дозволяє

використовувати великий набір бібліотек Arduino . Це дозволяє не так глибоко занурюватися в апаратну платформу та почати працювати з МК звичайного програміста, який раніше не мав такого досвіду роботи. На рис. 3.4 представлений алгоритм.

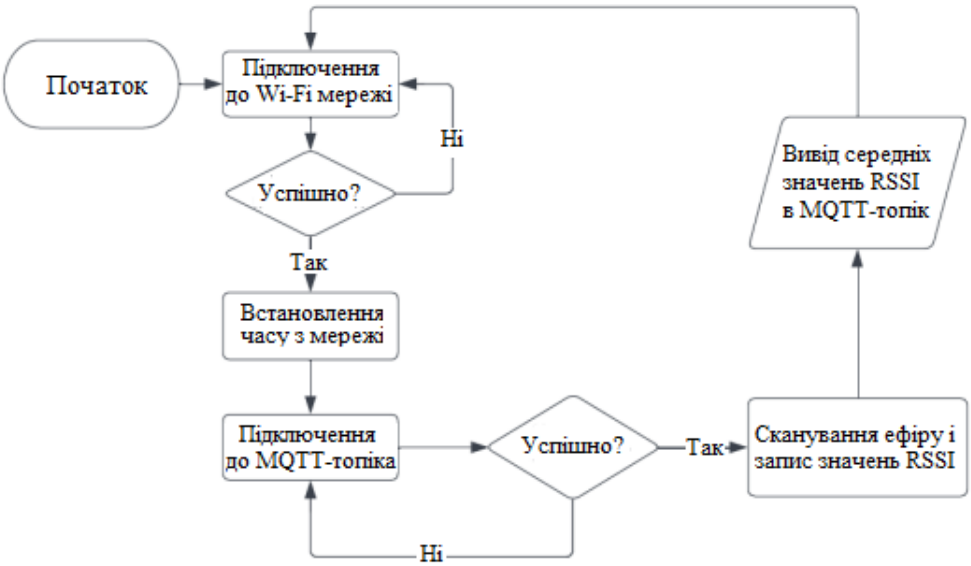

Рисунок 3.4 – Алгоритм ПЗ сканування Bluetooth ефіру для ESP32

Далі розглянемо деякі частини реалізації алгоритму докладніше. На рис. 3.5 наведено програмний код функції ініціалізації для ESP32.

```
157 void setup()
158 {
159   Serial.begin(9600);
160
161   WiFi.setHostname(HOSTNAME);
162   WiFi.mode(WIFI_AP_STA);
163   WiFi.begin(ssid, pass);
164   while (WiFi.status() != WL_CONNECTED)
165   {
166     delay(1000);
167   }
168   configTime(-5 * 3600, 0, "pool.ntp.org", "time.nist.gov");
169   now = time(nullptr);
170   while (now < 1510592825) {
171     delay(500);
172     now = time(nullptr);
173   }
174   struct tm timeinfo;
175   gmtime_r(&now, &timeinfo);
176
177   net.setCACert(local_root_ca);
178   client.setServer(MQTT_HOST, MQTT_PORT);
179   client.setBufferSize(2048);
180   client.setCallback(receivedCallback);
181   mqtt_connect();
182   BLEDevice::init("");
183 }
```

Рисунок 3.5 – Лістинг функції ініціалізації для ESP32

При запуску МК спочатку відбувається підключення до Wi-Fi-мережі. SSID мережі та пароль заздалегідь записані в константи. Далі завантажується час із віддаленого сервера та встановлюється локально. Коректний час потрібен для встановлення SSL-з'єднання з MQTT-брокером. Потім відбувається встановлення сертифіката і запускається MQTT підключення.

На рис. 3.6 зображено функцію для сканування Bluetoth- ефіру. У ній створюється екземпляр класу BLEScan і відбувається конфігурація об'єкта. При кожному результаті сканування спрацьовує callback -функція, у якій обробляються результати. Далі розраховується середнє значення RSSI у буфері.

```
185 void ScanBeacons() {
186   delay(1000);
187   for (int i = 0; i < bufferIndex; i++) {
188     buffer[i].count = 0;
189     buffer[i].rssi = 0;
190   }
191   bufferIndex = 0;
192
193   BLEScan* pBLEScan = BLEDevice::getScan();
194   MyAdvertisedDeviceCallbacks cb;
195   pBLEScan->setAdvertisedDeviceCallbacks(&cb);
196   pBLEScan->setActiveScan(true);
197   BLEScanResults foundDevices = pBLEScan->start(beaconScanTime);
198   for (uint8_t i = 0; i < bufferIndex; i++) {
199     buffer[i].rssi = buffer[i].rssi / buffer[i].count;
200   }
201
202   pBLEScan->stop();
203   delay(1000);
204 }
```

Рисунок 3.6 – Лістинг функції сканування Bluetooth - ефіру ESP32

```
116 class MyAdvertisedDeviceCallbacks : public BLEAdvertisedDeviceCallbacks {
117   public:
118
119     void onResult(BLEAdvertisedDevice advertisedDevice) {
120
121       if (!isInWhiteList(String(advertisedDevice.getAddress().toString().c_str()))) {
122         return;
123       }
124
125       extern uint8_t bufferIndex;
126       extern BeaconData buffer[];
127       if (bufferIndex >= 50) {
128         return;
129       }
130
131       int index = findElementIndex(String(advertisedDevice.getAddress().toString().c_str()));
132       if (advertisedDevice.haveRSSI()) {
133         buffer[index].rssi = buffer[index].rssi + advertisedDevice.getRSSI();
134       } else {
135         buffer[index].rssi =  0;
136       }
137
138       strcpy (buffer[index].address, advertisedDevice.getAddress().toString().c_str());
139       buffer[index].count = buffer[index].count + 1;
140     }
141
142     int findElementIndex(String mac) {
143       for (int i = 0; i < bufferIndex; i++) {
144         if (String(buffer[i].address).equals(mac)) {
145           return i;
146         }
147       }
148
149       bufferIndex += 1;
150       return bufferIndex - 1;
151     }
152 };
```

Рисунок 3.7 – Лістинг Callback -функції обробки результатів сканування

У callback –функції (рис. 3.7) відбувається збереження суми зчитаних значень RSSI та кількості вимірювань по кожному маячку для подальшого розрахунку середнього значення RSSI за один інтервал сканування.

У функції loop (рис. 3.8) написаний код, який повторюватиме МК вічно. За фактом це аналог вічного циклу while (true). У цій функції описано код перевірки підключення до Wi-Fi та MQTT -брокера та подальше ініціювання сканування Bluetooth -ефіру. Наприкінці функції відбувається відправлення даних з буфера в MQTT -топік реєстру Google Cloud IoT Core у вигляді JSON -рядка.

```
206 void loop()
207{
208   boolean result;
209   ScanBeacons();
210
211   if (WiFi.status() != WL_CONNECTED)
212   {
213     while (WiFi.waitForConnectResult() != WL_CONNECTED)
214     {
215       WiFi.begin(ssid, pass);
216       delay(10);
217     }
218   }
219   else {
220     if (!client.connected()) {
221       mqtt_connect();
222     } else {
223       client.loop();
224     }
225   }
```

```
227   String payloadString = "{\"e\":[";
228   for (uint8_t i = 0; i < bufferIndex; i++) {
229     payloadString += "{\"m\":\"";
230     payloadString += String(buffer[i].address);
231     payloadString += "\",\"r\":\"";
232     payloadString += String(buffer[i].rssi);
233     payloadString += "\"}";
234     if (i < bufferIndex - 1) {
235       payloadString += ',';
236     }
237   }
238
239   payloadString += "],\"st\":\"";
240   payloadString += String(WiFi.macAddress());
241   payloadString += "\",\"coords\":{\"x\":\"";
242   payloadString += String(STATION_X);
243   payloadString += "\",\"y\":\"";
244   payloadString += String(STATION_Y);
245   payloadString += "\"}}";
246
247   payloadString.getBytes(message_char_buffer, payloadString.length() + 1);
248   result = client.publish(topicEvents.c_str(),
249     message_char_buffer, payloadString.length(), false);
250   delay(3000);
251 }
```

Рисунок 3.8 – Лістинг функції loop

3.3 Обчислення координат маячків

Для обчислення координат маячка був написаний код JavaScript для середовища NodeJS, яке запускається в Google Cloud Functions. Для розрахунку був використаний метод трилатерації, який розглянуто в першому розділі даної роботи. Цей метод був обраний, тому що він найбільш простий у реалізації і не вимагає складного обладнання, а також не поступається точності іншим методам.

Обчислення ініціюються під час спрацювання тригера появи нового повідомлення на MQTT -топіці подій в IoT Core. Алгоритм хмарної функції представлений рис. 3.9.

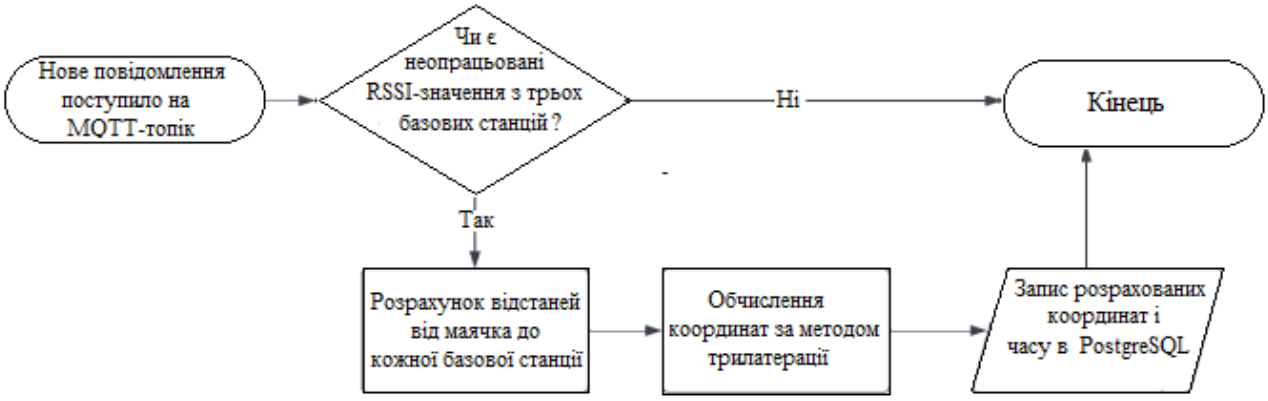

Рисунок 3.9 – Алгоритм розрахунку координат

Розрахунок відстані до маячка від базової станції проводиться за такою формулою (3.1):

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

Де P_m - це еталонне значення RSSI, коли Bluetooth-радіомітка знаходиться за 1 метр від базової станції, це значення має бути попередньо відкаліброване для різних типів радіоміток. P_d - це отриманий рівень RSSI від радіомітки при поточному вимірі. n – параметр загасання сигналу, який залежить від багатьох факторів, наприклад – вологість повітря. Підбирається дослідним шляхом.

Програмний код функції наведений на рис. 3.10.

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

Рисунок 3.10 – Лістинг функції розрахунку відстані до радіомітки з RSSI

3.4 Знаходження перетинів з історії координат радіоміток

Для вирішення задачі знаходження контактів було використано розширення PostGIS для PostgreSQL. Воно додає підтримку географічних

об'єктів і просторових індексів БД, завдяки чому з'являється можливість виконувати запити розташування в SQL. PostGIS дозволяє зберігати ряд географічних та просторових даних - точок, ламаних ліній, полігонів, растр. Всі ці дані можуть бути використані для різних операцій - пошук, знаходження найближчих точок і т.д.

У результаті, як розв'язання задачі знаходження координат зближення маячків з урахуванням часу була написана функція Google Cloud Functions, в якій виконується SQL -запит, що використовує вбудовані в PostGIS функції (рис. 3.11).

```
let dbClient = new pg.Client(conString);
dbClient.connect();
let result = await dbClient.query(
`SELECT DISTINCT ON (distance)
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 = '${params.mac}' OR
    c2.beacon_mac = '${params.mac}') AND
    (c1.time BETWEEN c2.time - (interval '${params.interval}s') AND c2.time + (interval '${params.interval}s')) AND
    ST_Distance(c1.geom,c2.geom) <= ${params.distance}`
);
```

Рисунок 3.11 – Лістинг функції отримання точок зближення маячків

У запиті використовується функція ST_Distance, яка входить до розширення PostGIS. MAC-адреса зараженої мітки, дистанція зближення та інтервал часу, коли перетин можна враховувати, можуть бути змінені динамічно. Ці дані передаються у функцію як аргументи з клієнтської частини.

3.5 Клієнтська частина. Формування звіту

Фінальним етапом роботи було написання клієнтської програми для можливості відображати звіт про знайдені контакти зараженої людини з Bluetooth -міткою з іншими людьми в приміщенні з історії його пересування.

Для реалізації клієнтської частини було обрано кросплатформний підхід. Він має на увазі використання спеціальних інструментів, які дозволяють створювати програми відразу для декількох операційних систем.

Головна ідея такого підходу полягає у використанні одного й того ж коду на різних платформах, завдяки чому витрачається менше людських ресурсів для реалізації програми. Для розробки потрібна лише одна команда програмістів, а в деяких випадках достатньо одного програміста. Дизайнер працює лише над одним набором графіки, що використовується на всіх платформах. Все це знижує кількість робочих годин та бюджет проекту. Крім того, використання єдиної логіки програми для всіх платформ зменшує ймовірність виникнення програмних помилок та полегшує процес тестування. Мінусом же кроссплатформенного додатка є нижча продуктивність, порівняно з нативним.

3.5.1 Flutter

Як фреймворк для реалізації клієнтської частини програми був обраний Flutter. Він розроблений Google і є проектом з відкритим вихідним кодом, який написаний на C/C++. Призначений для створення якісних та високопродуктивних додатків для iOS та Android. Крім того, на ньому можуть бути розроблені веб-додатки та графічні програми під настільні ОС Windows, MacOS, Linux.

Для формування коду на Flutter застосовується мова програмування Dart, яка також була розроблена в Google ще в жовтні 2011 року і отримала широку популярність з виходом фреймворку. Flutter цікавий своєю простотою та досить високою продуктивністю, порівнянною з нативними програмами.

На відміну від інших фреймворків, Flutter не застосовує JavaScript. Dart компілюється в бінарний код, тому операційну швидкість можна порівняти з Objective-C, Swift, Java і Kotlin.

3.5.2 Принцип побудови інтерфейсу

Flutter використовує свій графічний рушій - Skia для відтворення всього інтерфейсу програми. В результаті додаток виходить трохи важчим, так як у собі містить компоненти для роботи графічного рушія, проте працює значно швидше. Інші фреймворки малюють інтерфейс усередині WebView, або використовують

"прошарки" для взаємодії з нативними компонентами системи, що негативно позначається на продуктивності програми та, на відміну від Flutter, ускладнює розробку однакового інтерфейсу для всіх платформ. Та ж кнопка виглядає по-різному на Android та iOS.

Характерною рисою Flutter є спосіб побудови інтерфейсу, який значно відрізняється від використовуваних у нативній розробці на Android та iOS. Наприклад, Android логіка і інтерфейс розділені: логіка задається кодом, а інтерфейс - верстkою в xml. На Flutter все це задається за допомогою коду.

Для побудови інтерфейсу Flutter застосовує декларативний підхід на основі віджетів. Весь інтерфейс є ієрархічною структурою: всі елементи вкладені один в одного, при цьому відсутня можливість прямого доступу до елементів. Віджети перемальовуються за потребою — тільки коли в них щось змінилося, що додає значного приросту до швидкості роботи інтерфейсу.

Приклад оголошення віджету та його подання у вигляді дерева показано на рис. 3.12.

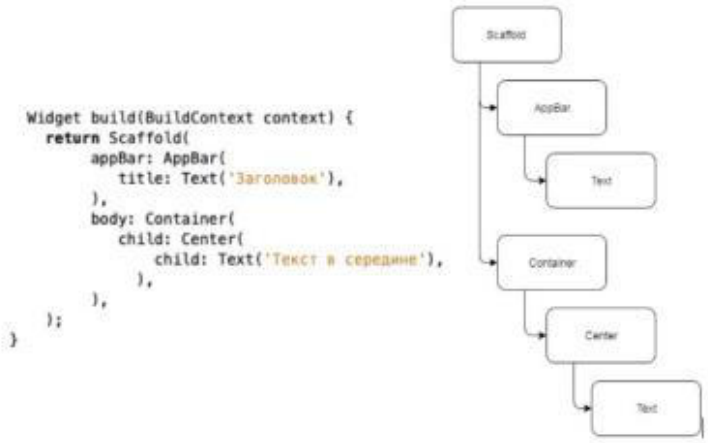

Рисунок 3.12 – Приклад побудови віджету та його подання у вигляді дерева

3.6 Схема взаємодії компонентів

На певному етапі опису інтерфейсу мобільного додатка розробник зможе ефективно декларувати код без його поділу на зони відповідальності. Це виходить із постійного ускладнення коду, і якщо не зуміти коректно розділити

візуальне відображення, логіку роботи з ним та бізнес-логіку, то додаток буде практично непідтримуваним для подальшої розробки.

Для коректного поділу зон відповідальності з часом були спроектовані різні шаблони, патерни, які дозволили командам розробки не винаходити щоразу велосипед, а використовувати і реалізовувати вже продумані ідеї щодо опису архітектури додатку. Так сформувалися шаблони проектування архітектури програми (Software architectural patterns). На даний момент є різні шаблони: MVC, MVVM, MVP.

В якості основи за реалізацію шаблону проектування архітектури програми був обраний MVVM, згодом трохи змінений під особливості проекту (рис. 3.13):

– View декларативно описує візуальне відображення, зберігає посилання на ViewModel та ViewStateProvider, взаємодіє з бізнес-логікою через ViewModel, отримує дані та оновлює відображення за допомогою ViewStateProvider;

– у ViewState зберігається стан екрану - різні змінні. При зміні ViewState відбувається реактивне оновлення View;

– ViewModel зберігає та оновлює ViewState екрана, є провідником між екраном та бізнес-логікою. Не зберігає посилання на сервіси та бізнес-логіку, містить необхідні колбеки, що передаються через Repository.

– Repository оперує даними. Може працювати з різними API та іншими джерелами, наприклад локальними БД.

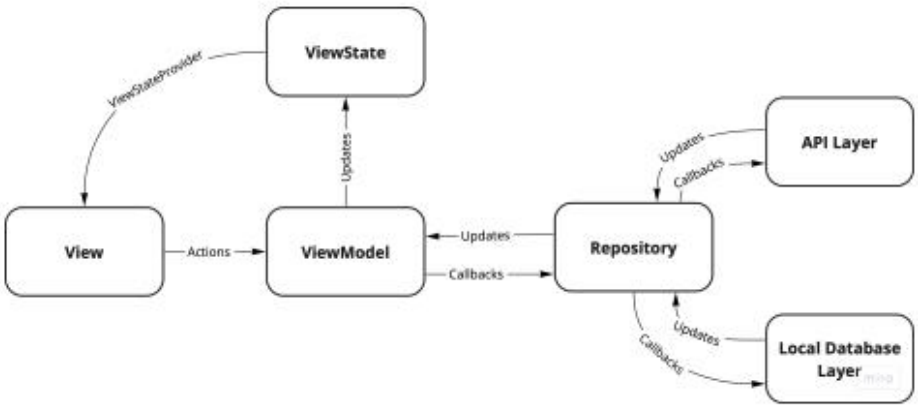

Рисунок 3.13 – Схема взаємодії компонентів

3.7 Схема роботи системи

Схема розробленої IPS системи представлена на рис. 3.14.

Робота системи складається з кількох етапів:

– як пристрої відстеження використовуються BLE iBeacon маячки, які транслюють з інтервалом від часток секунди до декількох секунд пакети оголошення;

– базові станції на основі МК ESP32, котрі які розставлені по трьох кутах приміщення, сканують Bluetooth -ефір;

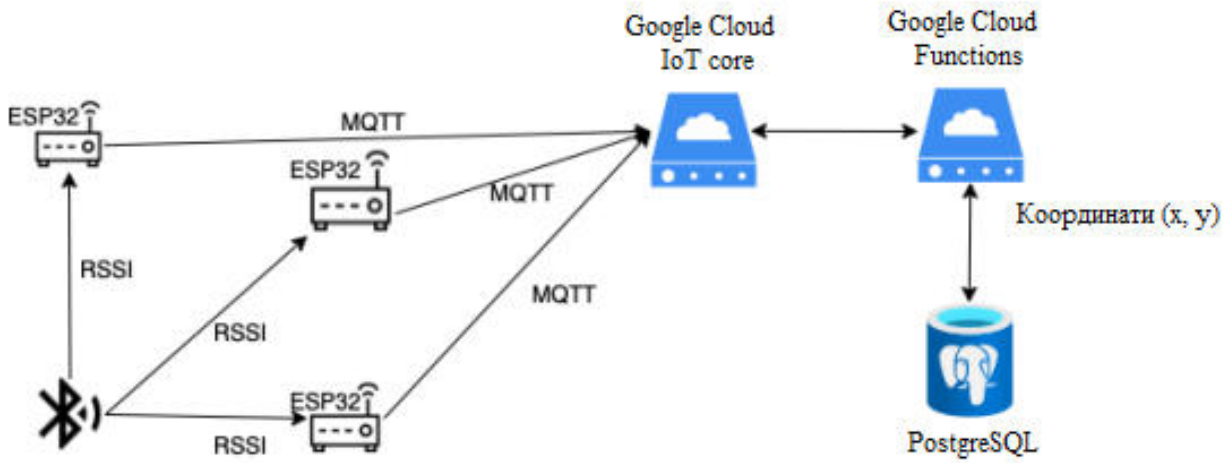

Рисунок 3.14 – Схема розробленої IoT системи

– зафіксовані значення RSSI iBeacon маячків передаються MQTT-брокеру. Як прототип був використаний Google Cloud IoT core;

– сервер, яким є Google Cloud Functions, підписаний на MQTT -топік брокера і при отриманні трьох значень RSSI від базових станцій розраховує координати методом трилатерації;

– розраховані координати записуються в БД - PostgresSQL у вигляді mac -адреса iBeacon -маячка, x і y координати, час фіксації координат;

– виведення звіту з графічним відображенням історії переміщення зараженої мітки та точках контактів з іншими мітками (рис. 3. 15).

Рисунок 3.15 – Вивід звіту

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Стихійні лиха та їх класифікація

Стихійні дії сил природи, поки що не повною мірою підвладні людині та щорічно завдають державі і населенню величезних збитків. Стихійні лиха - це такі явища природи, що викликають екстремальні ситуації, порушують нормальну життєдіяльність населення, роботу безлічі об'єктів. Стихійні лиха є трагедією для будь-якої держави.Через стихійні лиха страждає економіка країни, бо при цьому руйнуються виробничі підприємства, знищуються матеріальні цінності, гинуть люди.

Стихійні лиха - небезпечні природні явища, як правило раптового походження, хоча іноді і прогнозовані за допомогою метеорогогії, але на інтенсивність яких люди впливати не можуть. Їх можна класифікувати: за швидкістю переміщення - землетруси, зсуви, цунамі, снігопади, ожеледі - швидкі; підвищення рівня води в ріках через інтенсивні опади або танення снігу, льоду (повіні), звільнення внутрішньої енергії Землі, виверження вулканів - повільні. Часто виникають потужні, високошвидкісні потоки повітря через швидкий перепад значень атмосферного тиску (урагани, смерчі, циклони). Стихійні лиха речовинного характеру можуть ініціювати виникнення різноманітних полів, які негативно впливають на здоров'я, самопочуття людини. [57].

Стихійні явища часто виникають в комплексі, що значно посилює їх негативний вплив. Небезпечні природні явища визначаються трьома основними групами процесів - ендогенні, екзогенні та гідрометеорологічні. Стихійні лиха, які характерні для України, за структурою можна поділити на прості, що включають один елемент – наприклад, сильний вітер, зсув або землетрус

та складні. Вони складаються з декількох процесів однієї групи або кількох груп. Найбільші збитки спричиняють повені - 40%, на другому місці - циклони (20%), на третьому - посухи та землетруси (15%). Деякі стихійні лиха (пожежі, обвали, зсуви і навіть землетруси) можуть виникати в результаті дій самих людей, тобто мають антропогенне походження, але наслідки їх завжди є діями сил природи. Для кожного стихійного лиха характерна наявність властивих йому вражаючих чинників, що несприятливо впливають на стан здоров'я, життя людини [20].

Причинами стихійних лих можуть бути:

– швидке переміщення речовини (землетрусу, зсуви);
– вивільнення внутриземної енергії (вулканічна діяльність, землетруси);
– підвищення рівня вод річок, ставків і морів (повені, цунамі);
– вплив надзвичайно сильного вітру (урагани, торнадо, циклони).

Важливо своєчасно провести роботи, спрямовані на локалізацію природного лиха, щоб зменшити зони руйнувань, звести до мінімуму кількість загиблих та постраждалих.

В Україні найчастіше спостерігаються такі надзвичайні ситуації природного характеру:

– небезпечні геологічні явища (зсуви, обвали, осипки, просадки земної поверхні);
– небезпечні метеорологічні явища (зливи, урагани, сильні снігопади, сильний град, ожеледь);
– небезпечні гідрологічні явища (повені, паводки);
– природні пожежі лісових та торф’яних масивів;
– масові інфекції та хвороби людей, тварин, рослин.

В останні роки кількість стихійних лих в Україні та в світі в цілому значно збільшилася. Найчастіше в Україні виникають такі природні катастрофи як землетруси, повені, посухи (на Півдні України), лісові пожежі в літню пору року, снігові замети, зсуви поверхні.

Є серйозні підстави вважати, що масштабність впливу лиха й катастроф на соціальні, економічні, політичні та інших процесів сучасного нашого суспільства

та їх драматизм вже перевищили такий рівень, який дозволяв ставитися до них як до локальних збоїв у розміреному функціонуванні державних та громадських структур [58].

Отже, перед людиною та громадськістю в XXI в. вимальовується нова мета - глобальна безпека. Досягти цього можна, в першу чергу, за допомогою зміни світогляду людини, а також покращення системи профілактичних заходів у боротьбі зі стихійними лихами, а саме: вдосконалення рятувальних служб та рятувальної техніки, проведення попереджувальних заходів та пропагандистської роботи з громадянами щодо правил поведінки та дій під час стихійних лих. Це допоможе в майбутньому зменшити кількість загиблих та постраждалих від природних катастроф, а також зменшить матеріальні збитки, що були завдані стихійним лихом.

Природні лиха з часом нікуди не зникнуть. Будуть виникати землетруси в геологічно активних районах, будуть виникати повені, а штормові припливи стануть, раз у раз затопляти морські узбережжя, не обійдеться і пожеж. Людина безсила запобігти природним процесам, але тільки в наших силах зменшити кількість жертв і матеріальних втрат.

4.2 Соціальне значення охорони праці

Соціальне значення охорони праці полягає в сприянні росту ефективності суспільного виробництва шляхом безперервного вдосконалення і поліпшення умов праці, підвищення їх безпеки, зниження виробничого травматизму і профзахворювань [59].

Соціальне значення охорони праці проявляється в зростанні продуктивності праці, збереженні трудових ресурсів і збільшенні сукупного національного продукту.

Охорона праці полягає в сприянні росту ефективності виробництва, яке досягається шляхом безперервного вдосконалення і поліпшення умов праці, підвищення їх безпеки, зниження виробничого травматизму і профзахворювань.

Зростання продуктивності праці відбувається в результаті збільшення фонду робочого часу завдяки скороченню внутрішньо-змінних простоїв шляхом ліквідації мікротравм або зниження їх кількості, а також завдяки запобіганню передчасного стомлення шляхом раціоналізації і покращення умов праці та введенню оптимальних режимів праці і відпочинку та інших заходів, які сприяють підвищенню ефективності використання робочого часу.

Важливим питанням є зростання продуктивності праці, яка відбувається в результаті збільшення фонду робочого часу завдяки скороченню внутрішньозмінних простоїв шляхом ліквідації мікротравм або зниження їх кількості, а також завдяки запобіганню передчасного стомлення шляхом раціоналізації і покращення умов праці та введенню оптимальних режимів праці і відпочинку та інших заходів, які сприяють підвищенню ефективності використання робочого часу [59].

Особливої уваги заслуговує те, що збереження трудових ресурсів і підвищення професійної активності працюючих відбувається завдяки покращенню стану здоров’я і подовженню середньої тривалості життя шляхом покращення умов праці, що супроводжується високою трудовою активністю і підвищенням виробничого стажу. Підвищується професійний рівень також завдяки зростанню кваліфікації і майстерності. Відповідно і збільшення сукупного національного продукту відбувається завдяки покращенню вищеперелічених показників та їх складових компонентів [57].

Збереження трудових ресурсів і підвищення професійної активності працюючих відбувається завдяки покращенню стану здоров'я і подовженню середньої тривалості життя шляхом покращення умов праці, що супроводжується високою трудовою активністю і підвищенням виробничого стажу. Підвищується професійний рівень також завдяки зростанню кваліфікації і майстерності.

Збільшення сукупного національного продукту відбувається завдяки покращенню вищеперелічених показників та їх складових компонентів. Крім

того, соціальне значення охорони праці проявляється в зростанні продуктивності праці, збереженні трудових ресурсів.

За даними досліджень, комплекс заходів з поліпшення умов праці може забезпечити приріст продуктивності праці на 15-20%. Так, нормалізація освітлення робочих місць збільшує продуктивність праці на 6-13% та скорочує брак на 25%. Раціональна організація робочого місця підвищує продуктивність праці на 21%, раціональне фарбування робочих приміщень – на 25%.

Збільшення ефективного фонду робочого часу може бути досягнуто за рахунок скорочення тимчасової непрацездатності працівників внаслідок хвороб та виробничого травматизму.

ВИСНОВКИ

У ході виконання роботи було вирішено всі поставлені завдання. Було вивчено різні параметри, технології та методи, які можуть бути використані для визначення розташування всередині приміщень.

Проведене дослідження показало, що на практиці для вирішення задачі позиціонування всередині приміщень можуть застосовуватися різні технології бездротового зв'язку, кожна з яких володіє як позитивними , так і негативними характеристиками. Крім того, були виділені дві основні категорії методів локалізації, які можна застосовувати у зв'язку з цими технологіями зв'язку: range-based та range-free.

Результатом дослідження є розробка системи позиціонування всередині приміщення на основі методу трилатерації, яка записує всю історію переміщень людей з маячками в БД. При розробці системи було проведено роботу з iBeacon маячками, МК ESP32 та платформою Google Cloud.

Розроблена система дозволить своєчасно знаходити контактних осіб для подальшого вибіркового обмеження їхнього переміщення, що дає змогу внести контроль у процес поширення різних вірусів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 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

2 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.

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

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

5 Indoor Location Market Global forecast to 2026. URL: https://www.marketsandmarkets.com/Market-Reports/indoor- location-market-989.html (дата звернення: 15.03.2023).

6 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.

7 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.

8 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

9 F. Betti Sorbelli, CM Pinotti, S. Silvestri і SK Das, "Measurement Errors в Range-based Localization Algorithms for UAVs: Analysis and Experimentation," *in*

IEEE Transactions on Mobile Computing, 2020

10 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

11 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 Confernce Communication Software and Networks,* Xi'an, China, 27–29 May 2011.

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

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

14 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/

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

16 Методи локального позиціонування. URL: https://habr.com/ru/company/realtrac/blog/301706/ (Дата звернення: 02.04.2023).

17 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

18 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 Confencence on Communications (ICC)*, Paris, France, 2017

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

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

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

22 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.

23 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

24 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

25 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.

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

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

28 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

29 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

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

31 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

32 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 Acces*s, 2019.

33 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

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

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

36 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

37 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

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

39 Lo, L.; Li, C. Passive UHF-RFID Localization Based на Similarity

Measurement of Virtual Reference Tags. *IEEE Trans. Instrum. Meas*., 2019

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

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

42 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

43 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

44 Mekonnen, Z.W.; Slottke, 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.

45 UWB: Pros and Cons of Ultra-Wideband Technology. URL: https://www.konsyse.com/articles/uwb-pros-and-cons-of-ultra- wideband-technology/ (Дата звернення: 20.04.2023).

46 J. Rezazadeh, R. Subramanian, K. Sandrasegaran, X. Kong, M. Moradi та F. Khodamoradi, "Novel iBeacon Placement for Indoor Positioning in IoT," *in IEEE Sensors Journal*, vol. 18, no. 24, pp. 10240-10247, 2018

47 hIoTron. URL: https://www.hiotron.com/iot-architecture-layers/ (Дата звернення: 20.04.2023).

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

49 TECH GEEK. URL: https://tech-geek.com/choosing-microcontroller/ (Дата звернення: 25.04.2023).

50 Perry Xiao, «Designing Embedded Systems and the Internet of Things

(IoT) with the ARM mbed», John Wiley & Sons Limited, 2018. 313 p.

51 ESP8266 WI-FI MCU. Espressif, URL: https://www.espressif.com/en/products/socs/esp8266 (Дата звернення: 13.05.2023).

52 WI-FI & Bluetooth MCU. Espressif, URL: https://www.espressif.com/en/products/socs/esp32 (Дата звернення: 13.05.2023).

53 Choosing best hardware for your next IoT project. URL: https://developer.ibm.com/ technologies/iot/articles/iot-lp101- best-hardware-devices-iot-project/ (Дата звернення: 14.05.2023).

54 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.

55 Importance of Cloud Computing for IoT. URL: https://www.einfochips.com/blog/importance-of-cloud-computing-for-large-scale-iot-solutions/ (дата звернення: 16.05.2023).

56 Архітектура платформи Google Cloud. URL: https://cloud.google.com/architecture (дата звернення: 18.05.2023).

57 Стеблюк М.І. Цивільна оборона: Підручник. Знання, 2006. 487 с.

58 Толок А.О., Крюковська О.А. Безпека життєдіяльності: Навч. посібник. 2011. 215 с.

59 Основи охорони праці: Підручник.; 3-те видання, доповнене та перероблене / За ред. К. Н Ткачука. К.: Основа, 2011. 480 с.

Додаток А.

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

"Затверджую"

Завідувач кафедри КС

_____ Осухівська Г.М.

"___" _____2023 р

ІОТ СИСТЕМА МОНІТОРИНГУ ТА АНАЛІЗУ ІНФОРМАЦІЇ ПРО НАЯВНІСТЬ ЛЮДЕЙ В ПРИМІЩЕННІ

ТЕХНІЧНЕ ЗАВДАННЯ

на _*10*_ аркушах

Вид робіт: Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»	«ВИКОНАВЕЦЬ»
Керівник кваліфікаційної роботи	Студент групи СІ-41
_____ к.т.н., доц. Баран І.О.	_____ Ониськів Р.Б.
«____» _____ 2023 р.	«____» _____ 2023 р.

Тернопіль 2023

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «IoT система моніторингу та аналізу інформації про наявність людей в приміщенні».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.625.00.00

1.2 Виконавець

Студент групи СІ-41, факультету комп’ютерно-інформаційних систем і програмної інженерії, кафедри комп’ютерної інженерії, Тернопільського національного технічного університету імені Івана Пулюя, Ониськів Роман Богданович.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№ 4/7-238 від 28.02.2023 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 28.02.2023 р.

Плановий термін завершення виконання кваліфікаційної роботи – 22.06.2023 р.

1.5 Порядок оформлення та пред’явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ГОСТ, ЕСКД, ЕСПД та ДСТУ.

Пред’явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи.

Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред’явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

IoT система моніторингу та аналізу інформації про наявність людей в приміщенні призначена для запобігання в можливих контактів користувачі з хворими людьми за даними історії їх позиціонування всередині приміщення.

До складу IoT системи повинні входити як апаратна складова, так і програмна. У кваліфікаційній роботі бакалавра необхідно дослідити основні методи та технології для локалізації у приміщенні; провести аналіз та вибір параметрів для локалізації; здійснити вибір метрик, котрі потрібні для здійснення розрахунку координат людей всередині приміщення. Виконати дослідження існуючих IoT платформ. Спроектувати архітектуру. Виконати огляд існуючих протоколів передачі даних. Застосувати хмарні сервіси з використанням можливостей платформи Google Cloud.

Доцільність створення IoT зумовлена можливістю своєчасного знаходження контактних осіб та подальшого вибіркового обмеження їхнього переміщення, що дозволить вести контроль у процес поширення вірусу.

2.2 Мета створення системи

Основна мета проектування IoT системи моніторингу та аналізу інформації про наявність людей в приміщенні полягає розробці системи, котра запобігає особам у набутті можливих контактів з хворими людьми за даними історії їх позиціонування всередині приміщення.

Для того, щоб досягти поставленої мети роботи, необхідно розв’язати наступні задачі:

– вивчити основні методи та технології для локалізації у приміщенні;

– здійснити вибір метрик, необхідних розрахунку координат людей всередині приміщення;

– дослідити існуючі IoT платформи та вибрати конкретну для реалізації системи;

– забезпечити збереження історії переміщення людини;

– визначити контакти з історії переміщення

– здійснити виведення результатів у вигляді звіту;

– реалізувати прототипу системи.

2.3 Характеристика об'єкту

2.3.1 Основні задачі та функції об’єкту

IoT система моніторингу та аналізу інформації про наявність людей в приміщенні може використовуватись підприємствами і установами державної та приватної форм власності, в яких потрібно забезпечити контроль за особами в приміщенні.

Ефективне вирішення проблеми передбачає використання на рівні апаратного забезпечення:

– мікроконтолера ESP32;

– iBeacon Bluetooth маячків;

– середовища передачі даних;

– сервера обробки даних.

На рівні програмного забезпечення, IoT система повинна:

– мати модуль опрацювання даних;

– мати базу даних;

– використовувати спеціалізовані сервіси з платформи Google Cloud.

При проектуванні складових системи, зокрема апаратного і програмного забезпечення, необхідно проаналізувати предметну область та розробити концептуальні схеми взаємодії та розподілу прав доступів до даних.

IoT система моніторингу та аналізу інформації про наявність людей в приміщенні повинна своєчасно знаходити контактні особи та обмежувати їх переміщення вибірково, а не закривати все підприємство на карантин.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

IoT система моніторингу та аналізу інформації про наявність людей в приміщенні повинна забезпечувати формування часової мітки для запису даних у сховище, швидко та надійно реагувати на не авторизоване проникнення сторонніх. В цілому, у проектованій системі повинні бути забезпечені:

- надійність роботи апаратної частини;
- точність детектування особи;
- продуктивність роботи програмного забезпечення;
- паралельний доступ до бази даних різним користувачам;
- розмежування прав доступу до бази даних;
- часова ефективність та ефективність використання ресурсів системи;
- надання зручного користувацького інтерфейсу для роботи з радіомітками та відповідним програмно-апаратним забезпеченням.

3.1.1 Вимоги до структури та функціонування системи

До структури та функціонування IoT система моніторингу та аналізу інформації про наявність людей в приміщенні входять:

- мікроконтролер ESP32 із вбудованим Wi Fi модулем;
- BLE iBeacon маячки;
- системне та прикладне програмне забезпечення Arduino Core;
- Google Cloud IoT core
- сервер баз даних від Google Cloud Functions;
- клієнтська частина, що забезпечує зв’язок між користувачами та базою даних.

В цілому, концептуальна модель IoT система моніторингу та аналізу інформації про наявність людей в приміщенні повинна описувати предметну область, а саме визначення позиціювання та можливість локалізації особи в приміщенні без використання діапазону передачі даних. Клієнтська частина програмного забезпечення відповідає за можливості обліку даних та забезпечення їх захисту.

Функціональні вимоги, що висуваються до системи, виглядають наступним чином:

- можливість зчитування та запису даних від iBeacon Bluetooth маячків;
- можливість вводу, редагування та знищення даних про особу;
- можливість запобігання неавторизованому доступу (логічного);
- можливість формування звітів;
- можливість керування правами доступу до інформаційних ресурсів;
- розподіл прав доступу;
- часова ефективність на рівні 3 с.
- масштабованість програмної та апаратної складових системи.

3.1.2 Вимоги до способів та засобів зв’язку між компонентами системи

Як пристрої відстеження використовуються BLE iBeacon маячки, які транслюють з інтервалом від часток секунди до декількох секунд пакети оголошення, базові станції на основі МК ESP32, котрі які розставлені по трьох кутах приміщення, сканують Bluetooth –ефір. зафіксовані значення RSSI iBeacon маячків передаються MQTT- брокеру. Як прототип був використаний Google Cloud IoT core; розраховані координати записуються в БД - PostgresSQL у вигляді mac -адреса iBeacon -маячка, x і y координати, час фіксації координат; виведення звіту з графічним відображенням історії переміщення зараженої мітки та точках контактів з іншими мітками. Загалом, структура та архітектурне рішення при проектуванні IoT системи відповідає архітектурі «клієнт-сервер».

3.1.3 Вимоги по діагностуванню системи

Діагностика IoT системи моніторингу та аналізу інформації про наявність людей в приміщенні відбувається у відповідності до затвердженого розкладу профілактичних заходів.

3.1.4 Перспективи розвитку, модернізація системи

Перспективами розвитку та модернізації IoT система моніторингу та аналізу інформації про наявність людей в приміщенні є можливий перехід на нові версії мікроконтролера, адаптація та інтеграція з додатковими пристроями аутентифікації та авторизації користувачів. Існуюча апаратна складова IoT системи при цьому не повинна зазнавати значних змін, а програмне забезпечення системи повинно передбачати гнучкість та здатність до масштабування.

У випадку модернізації системи шляхом інтеграції нових пристроїв і як наслідок одержання нових даних, система керування базами даних повинна забезпечити архівування історичної інформації або додавання нових даних без порушення цілісності існуючих.

3.1.5 Вимоги до надійності системи

IoT система моніторингу та аналізу інформації про наявність людей в приміщенні повинна бути захищена на кількох рівнях: фізичному, операційної системи та на рівні доступу до бази даних. Фізичний рівень захисту повинен забезпечувати надійність щодо доступу до апаратного забезпечення, зокрема мікроконтролера та під’єднаних до нього компонентів, а також сервера.

На рівні операційної системи повинен бути організований доступ на основі визначених прав доступу до використання програмного забезпечення, що функціонує та керує даними при використанні BLE iBeacon маячків, для визначення особи у приміщення.

Доступ до інформації, що зберігається у базі даних, повинен бути авторизованим на рівні системи керування базами даних. Лише користувачі з наділеним правом доступу та адміністратори мають можливість вносити зміни у базу даних в межах їхньої компетенції.

3.1.6 Вимоги до функцій та задач, які виконує система

Функціональні вимоги та задачі, які повинна реалізовувати IoT система моніторингу та аналізу інформації про наявність людей в приміщенні полягають в наступному:

– можливість зчитування унікального ідентифікатора;

– формування зворотного зв’язку при успішному чи невдалому детектуванню особи;

– передача даних від мікроконтролера до сервера, на якому розгорнуто базу даних;

– забезпечення зв’язку клієнтської частини з базою даних;

– надання точних та адекватних результатів на запит користувачів;

– забезпечення часової ефективності роботи системи;

– забезпечення контролю над доступом до інформації про перебування осіб в приміщенні;

– забезпечення зручності використання програмного продукту;

– візуалізація графіків результатів;

– можливість розгортання та створення резервних копій бази даних.

3.1.7 Вимоги до апаратного забезпечення

Вимоги до апаратного забезпечення:

– мікроконтролер ESP32 із вбудованим Wi Fi модулем;

– BLE iBeacon маячки;

Вимоги до клієнтських робочих станцій:

– процесор – тактова частота > 1,8 ГГц з кількістю логічних ядер > 2;

– RAM – не менше 4 Гб;

– об’єм жорсткого диску – не менше 240 Гб.

Вимоги до сервера:

– процесор – 2,2 ГГц або більш потужний з кількістю логічних ядер >8;

– RAM – 16 ГБ або більше;

– об’єм дискового простору – 1Тб.

3.1.8 Вимоги до програмного забезпечення

Системне програмне забезпечення Arduino Core.

Програмне забезпечення робочих станцій – Windows 10, .NET Framework та інше програмне забезпечення.

Програмне забезпечення сервера – Windows Server 2012, MS SQL Server 2019.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

– пояснювальної записки;

– графічного матеріалу:

1 Основні методи локалізації.

2 Структура IoT рішень та Схема роботи протоколу MQTT.

3 Алгоритми сканування BLUETOOTH ефіру для ESP32 та розрахунку координат.

4 Лістинги фрагментів програмного коду.

5 Схеми розробленої IoT системи та взаємодії компонент

*Примітка: У комплект документації можуть вноситися міни та доповнення в процесі розробки.

5 Техніко-економічні показники

Планована собівартість IoT система моніторингу та аналізу інформації про наявність людей в приміщенні повинна становити не більше 40 000 грн.

*Примітка: собівартість системи може змінюватись під час розрахунку в процесі розробки.

6 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1.	Ознайомлення з завданням до кваліфікаційної роботи	01.03 – 05.03
2.	Розробка технічного завдання	06.03 – 16.03
3.	Підбір джерел про IoT системи моніторингу та аналізу інформації про наявність людей в приміщенні	17.03 – 28.04
4.	Опрацювання джерел про такі системи	29.04 – 10.05

5.	Виконання дослідження щодо IoT системи моніторингу та аналізу інформації про наявність людей в приміщенні	11.05 – 18.05
6.	Розроблення програмного коду	19.05 – 26.05
7.	Оформлення розділу «Аналіз технічного завдання»	27.05 – 30.05
8.	Оформлення розділу «Проектна частина»	30.05 – 02.06
9.	Оформлення розділу «Практична частина»	03.06 – 08.06
10.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	20.05 – 04.06
11.	Оформлення кваліфікаційної роботи	09.06 – 12.06
12.	Нормоконтроль	12.06 – 16.06
13.	Перевірка на плагіат	16.06 – 18.06
14.	Попередній захист кваліфікаційної роботи	17.06 – 19.06
15.	Захист кваліфікаційної роботи	23.06

7 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.