

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ПАРУБОК Євген Вадимович

**Програмно-апаратний модуль обробки даних із пристроїв Інтернету речей /
Software-hardware module for processing data from Internet of Things devices**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент
групи КН-43
Є. В. Парубок

Науковий
керівник:
к.т.н., доцент,
О.Р. Осолінський

Кваліфікаційну роботу допущено до
захисту

«___» _____ 2025 р.

В.о завідувача кафедри
_____ Н.М. Васильків

Тернопіль – 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
«_____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ПАРУБОК Євген Вадимович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Програмно-апаратний модуль обробки даних із пристроїв Інтернету речей / Software-hardware module for processing data from Internet of Things devices

керівник роботи О.Р. Осолінський

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

затверджені наказом по університету від __ грудня _____ р. № _____.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- провести аналіз сучасних технологій Інтернету речей (IoT) та архітектури систем збору та обробки даних і типові сценарії застосування;
- здійснити огляд апаратних і програмних рішень для побудови IoT-модулів;
- розробити архітектуру програмно-апаратного модуля збору та обробки даних з врахуванням підтримки локального і віддаленого доступу до інформації;
- розробити алгоритми функціонування ключових компонентів модуля;
- реалізувати веб-інтерфейс для локального моніторингу параметрів у режимі реального часу, використовуючи HTML, JavaScript та вбудований сервер на ESP32;
- протестувати працездатність програмно-апаратного модуля в умовах реального середовища.

5. Перелік графічного матеріалу в роботі:

- розширена архітектура системи;
- діаграма послідовності програмно – апаратного модуля;

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2025 р.	

Студент _____ Є. В. Парубок
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ О.Р. Осолінський
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему “Програмно-апаратний модуль обробки даних із пристроїв Інтернету речей” на здобуття освітнього ступеня “бакалавр” зі спеціальності 122 “Комп’ютерні науки” освітньої програми “Комп’ютерні науки” написана обсягом 71 сторінок і містить 18 рисунків, 4 таблиці, 47 джерел та 4 додатки.

Метою роботи є розробка програмно-апаратного модуля обробки даних із пристроїв Інтернету речей з підтримкою збереження, візуалізації та віддаленого доступу до даних у реальному часі.

В роботі застосовано методи системного аналізу, логічного структурування, програмування мікроконтролерних систем, технології Інтернету речей (IoT), протоколів передачі даних (MQTT, HTTP), а також засоби розробки інтегрованого програмного забезпечення для ESP32.

Запропоновано архітектуру модульної IoT-системи, яка включає сенсорний блок, модуль збору й збереження даних, модуль бездротової передачі інформації, веб-інтерфейс для взаємодії з користувачем та серверну частину з функцією обробки запитів і роботи через MQTT-брокер.

Розроблено програмно-апаратний модуль на основі мікроконтролера ESP32 з підтримкою роботи сенсора MPU6050, microSD-карти, OLED-дисплея та мережевого підключення. Реалізовано програмну частину системи, включаючи веб-сервер, обробку MQTT-команд, логіку запису файлів CSV, обробку конфігураційних файлів та механізми індикації стану.

Проведено тестування модулів, діагностику типових помилок та верифікацію стабільності роботи. Результати підтвердили функціональність усіх компонентів та відповідність вимогам до модульних IoT-систем із підтримкою дистанційного керування та збору даних.

Ключові слова: ІНТЕРНЕТ РЕЧЕЙ, ESP32, MQTT, WI-FI, MPU6050, MICROSD, ВЕБ-ІНТЕРФЕЙС, СЕНСОРНА СИСТЕМА, ІОТ-МОДУЛЬ, ПРОГРАМНО-АПАРАТНА СИСТЕМА.

ANNOTATION

The qualification work on the topic " Software-hardware module for processing data from Internet of Things devices " for obtaining the degree of "Bachelor" in specialty 122 "Informatics" of the educational program "Informatics" is written on 71 pages and contains 18 figures, 4 tables, 47 sources and 4 appendices.

The aim of this work is to develop a hardware-software module for processing data from Internet of Things (IoT) devices with support for data storage, visualization, and remote real-time access.

The work employs methods of systems analysis, logical structuring, microcontroller programming, Internet of Things (IoT) technologies, data transmission protocols (MQTT, HTTP), as well as tools for developing integrated software for ESP32.

A modular IoT system architecture is proposed, which includes a sensor unit, a data collection and storage module, a wireless data transmission module, a web interface for user interaction, and a server-side component for processing requests and communicating via an MQTT broker.

A hardware-software module was developed based on the ESP32 microcontroller, supporting operation with the MPU6050 sensor, microSD card, OLED display, and network connection. The software part of the system has been implemented, including a web server, MQTT command processing, CSV file logging, configuration file handling, and system status indication mechanisms.

Module testing, diagnosis of common errors, and verification of system stability were conducted. The results confirmed the functionality of all components and their compliance with the requirements for modular IoT systems with remote control and data acquisition capabilities.

Keywords: INTERNET OF THINGS, ESP32, MQTT, WI-FI, MPU6050, MICROSD, WEB INTERFACE, SENSOR SYSTEM, IOT MODULE, HARDWARE-SOFTWARE SYSTEM.

ЗМІСТ

Вступ	6
1 Стан та аналіз предметної області	9
1.1 Опис предметної області	9
1.2 Огляд і аналіз існуючих рішень	18
1.3 Системи моніторингу та контролю для розумних міст на основі IoT	21
1.4 Постановка задачі	24
2. Алгоритмічне та інформаційне забезпечення модуля обробки даних із пристроїв IoT	26
2.1 Розробка архітектури програмно-апаратного модуля	26
2.2 Алгоритми функціонування компонент модуля	29
2.3 Інформаційне забезпечення програмно-апаратного модуля	34
3 Програмно-технологічне забезпечення модуля обробки даних із пристроїв інтернету речей	39
3.1 Вибір програмних і апаратних засобів	39
3.2 Реалізація ключових модулів	42
3.3 Інтерфейс взаємодії з користувачем	45
3.4 Тестування функціональності	47
Висновки	53
Список використаних джерел	55
Додаток А Розширена архітектура системи	60
Додаток Б Діаграма послідовності програмно – апаратного модуля	61
Додаток В Код програмно-апаратного модуля	62
Додаток Г Апробація отриманих результатів	66

ВСТУП

Світ переживає технологічні зміни, які глибоко впливають на спосіб нашого життя. Досягнення, засновані, зокрема, на тому, що називається Інтернетом речей, або IoT, впливатимуть на різні види діяльності, такі як промисловість, міста, лікарні, офіси та наші сфери. Ці нові технології революціонізують те, як виробляються та споживаються товари, як керуються ресурси, як люди пересуваються містами, охорону здоров'я та навіть те, як люди проводять дозвілля вдома [1,2]. Діловий світ виділяє три основні напрямки для практичних застосувань Інтернету речей: розумне життя, розумні міста та промисловість 4.0.

Індустрія 4.0, яку також називають розумною промисловістю, спрямована на автоматизацію виробництва товарів за допомогою Інтернету речей і кіберфізичних систем за допомогою спілкування між людьми, машинами та ресурсами [3]. Основна мета розумної індустрії – забезпечити індивідуальне налаштування виробництва товарів автономним способом і дозволити контролювати життєвий цикл продукту. Інші концепції, що використовуються в галузях промисловості, які є похідними від ринку, створеного технологією IoT, це Інтернет послуг (IoS), який включає в себе оптимізацію транспорту та логістики, включаючи транспортування сировини та розповсюдження кінцевої продукції розумним способом, та Інтернет енергії (IoE), який визначає, як природні ресурси використовуються ефективно та чистим способом.

Концепція IoT також є ядром для розумних міст, метою якого є створення стійкої моделі для міст [4]. Збираючи та обробляючи дані, отримані від мережі датчиків, можна оптимізувати процеси, заощаджуючи енергію та зменшуючи споживання та відходи. Міста можуть отримати користь від IoT у кількох сферах, таких як транспорт, громадська безпека, охорона здоров'я, управління ресурсами та надання послуг. У транспорті можна використовувати технологію для керування потоком транспорту та налаштування громадського транспорту в режимі реального часу відповідно до його потреб. Це також робить можливим розробку спільних автономних транспортних засобів. Усі ці дії можуть призвести

до покращення якості повітря та води, зменшивши проблеми зі здоров'ям, пов'язані із забрудненням. Будинки можуть отримати вигоду в сферах безпеки та управління енергією.

Поступовий розвиток інфраструктури обчислювальної електроніки призвів до розробки серверів з компетенціями для інтенсивного накопичення та управління даними, відомих як великі дані, що відкриває двері для концепції IoT та інтелектуальних пристроїв [5]. IoT дозволяє побутовій техніці збирати та надсилати дані для обчислення в хмарі. Це дає можливість невеликим пристроям з обмеженими можливостями обробки та потужності використовувати величезні обсяги даних, зібраних іншими пристроями, і використовувати інформацію, отриману шляхом масивної обробки. Цей процес дозволяє цим пристроям приймати розумні рішення, тобто вони здатні спеціально адаптуватися до звичок своїх користувачів.

Метою кваліфікаційної роботи є розробка програмно-апаратного модуля обробки даних із пристроїв Інтернету речей.

Завдання кваліфікаційної роботи полягають у послідовному виконанні наступних етапів:

- провести аналіз сучасних технологій Інтернету речей (IoT) та архітектури систем збору та обробки даних і типові сценарії застосування;
- проаналізувати системи моніторингу та контролю для розумних міст на основі IoT;
- сформулювати постановку задачі;
- розробити архітектуру програмно-апаратного модуля збору та обробки даних з врахуванням підтримки локального і віддаленого доступу до інформації;
- розробити алгоритми функціонування ключових компонентів модуля;
- розробити інформаційне забезпечення програмно-апаратного модуля
- провести вибір програмних і апаратних засобів
- реалізувати ключові модулі;
- реалізувати інтерфейс з користувачем;
- протестувати працездатність програмно-апаратного модуля.

Об'єкт дослідження - процеси взаємодії між сенсорними пристроями, мікроконтролерними системами та хмарними або локальними сервісами в архітектурі Інтернету речей.

Предмет дослідження - системний аналіз, логічне структурування, модульне проектування, алгоритмічна декомпозиція, аналіз діаграм станів та послідовності.

Результати кваліфікаційної роботи апробовані та опубліковані у матеріалах студентської науково-практичної конференції “Інтелектуальні інформаційні технології в прикладних дослідженнях” (ІТAR – 2025), м. Тернопіль, Україна, 27-29 травня 2025 р.

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 СТАН ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Інтернет речей (IoT) — це технологічна революція та зміна в тому, як люди сприймають навколишній світ і спілкуємося з ним. Зараз сфера ІТ розвивається дуже динамічно, яка суттєво змінює повсякденне життя шляхом інтеграції все більшої кількості пристроїв і систем у глобальну мережу. Вони варіюються від простих побутових пристроїв, таких, як термометри чи датчики вологості, до передових систем моніторингу та контролю в промисловості. IoT пронизує всі сфери життя, надаючи нові можливості та завдання [6]. В останні роки спостерігається чітка тенденція збільшення кількості домашніх пристроїв, оснащених функціями IoT. Концепція розумного дому стає все більш поширеною, де всі елементи — від освітлення та опалення до кухонних приладів і систем безпеки — об'єднані в єдину мережу, що забезпечує дистанційне керування та моніторинг. Однак це лише верхівка айсберга, оскільки потенціал IoT також глибоко проникає в промисловість [7] [8], транспорт, охорону здоров'я [9], сільське господарство та багато інших економічних секторів. В таблиці 1.1 узагальнено типові рішення IoT.

Технології IoT використовують різні протоколи зв'язку, від популярних бездротових стандартів, таких як Wi-Fi [10], Bluetooth [11] або Zigbee [12], до дротових протоколів, таких як Mod bus [13] або CAN [14]. Ця різноманітність дозволяє гнучко адаптуватися до різних вимог середовищ, одночасно відкриваючи нові можливості і застосування. Окрім згаданих вище бездротових і дротових стандартів, технології IoT також використовують кілька інших протоколів зв'язку, які розширюють можливості та сферу їх застосування. Одним із таких важливих протоколів є GSM [15], який забезпечує зв'язок між пристроями IoT та мобільними мережами. Завдяки цьому пристрої можна віддалено контролювати та контролювати навіть у місцях без доступу до мереж Wi-Fi або Bluetooth. З розвитком мобільних технологій стандарт 5G [16] також стає все більш важливим.

Таблиця 1.1 - Ключові сфери застосування Інтернету речей (IoT)

Ключові сфери	Опис
Розумні міста	Інтелектуальне управління освітленням, моніторинг відходів і контроль дорожнього руху для розумного міського управління.
Розумні будинки	Енергоефективні розумні пристрої, дистанційний моніторинг побутової техніки, автоматизація безпеки.
Індустрія 4.0	Прогнозне технічне обслуговування, моніторинг машин у реальному часі, автоматизація процесів та інтеграція промислового IoT.
Охорона здоров'я	Носимі пристрої для моніторингу здоров'я, підключення медичних пристроїв, телемедичні рішення.
Точне землеробство	Розумне зрошення, моніторинг стану врожаю, аналіз ґрунту та відстеження кліматичних умов.
Транспорт і логістика	GPS-відстеження транспортних засобів, моніторинг умов транспортування, автоматизація складів.
Роздрібна торгівля	Управління запасами, розумні полиці, автоматизовані системи оплати та аналіз поведінки клієнтів.
Екологічний моніторинг	Контроль якості повітря і води, відстеження кліматичних змін, виявлення катастроф.
Ланцюг постачання та складування	Розумні RFID-мітки, відстеження в реальному часі, автоматизовані логістичні рішення.
Розваги та спорт	Носимі пристрої для відстеження фізичної активності, інтеграція віртуальної реальності, персоналізовані тренування.
Банківська справа та фінанси	Смарт-аутентифікація платежів, біометрична безпека та виявлення шахрайства.
Освіта	Розумні навчальні середовища, моніторинг студентів у реальному часі, інтерактивні освітні інструменти.
Розумні енергомережі	Інтелектуальний розподіл енергії, відстеження споживання та автоматизоване управління мережею.

Крім того, 5G, пропонує високу швидкість передачі даних і низькі затримки, що робить його ідеальним рішенням для додатків IoT, які вимагають високої пропускної здатності та негайної реакції. Завдяки 5G стають можливими нові додатки IoT, такі як передові системи моніторингу в реальному часі та автономні

транспортні засоби. Ще один важливий протокол, який набирає популярності в IoT, — це LoRa [16]. LoRa — це протокол бездротового зв'язку, який забезпечує передачу даних на великі відстані, споживаючи мінімальну енергію. Завдяки цьому LoRa використовується у випадках, коли необхідно контролювати та моніторити пристрої на великих територіях, наприклад, коли ведеться моніторинг полів та ферм. В таблиці 1.2 підсумовано протоколи передачі даних, який використовується IoT.

Таблиця 1.2 - Протоколи передачі даних IoT

№	Протокол	Опис
1	HTTP	Основний комунікаційний протокол Інтернету для передачі веб-даних.
2	Zigbee	Низькоенергетичний бездротовий протокол для сенсорних мереж.
3	MQTT	Полегшений протокол зв'язку, мінімізує трафік для IoT-пристроїв.
4	Wi-Fi	Бездротова мережева технологія для локальних мереж (LAN).
5	GSM	Стандарт мобільного зв'язку для голосових викликів та SMS.
6	Bluetooth	Короткорадіусний бездротовий протокол для обміну даними.
7	Ethernet	Провідна технологія передачі даних у локальних мережах.
8	RFID	Технологія радіочастотної ідентифікації для контролю доступу.
9	NFC	Технологія ближнього зв'язку (до 10 см) для мобільних платежів.
10	Sigfox	Глобальний LPWAN-протокол для передачі малих даних на великі відстані.
11	NB-IoT	Технологія вузькополосних IoT-мереж для стабільних з'єднань.
12	LoRa	LPWAN-технологія для передачі даних на великі відстані.

Однак із розширенням IoT виникають нові проблеми, особливо в безпеці [17]. Багато виробників зосереджуються на швидкому виведенні на ринок недорогих рішень IoT, часто за рахунок відповідних стандартів безпеки. Це

призводить до серйозних вразливостей, які можна використати для несанкціонованого доступу та витоку даних, що порушує конфіденційність і безпеку користувачів. Незважаючи на ці загрози, у розвитку IoT є величезний потенціал. В епоху динамічного розвитку концепції розумних міст [18], розумної енергетики [19] та розумного сільського господарства [20] системи IoT стають ключовим інструментом, що дозволяє оптимізувати ресурси, зменшити витрати та підвищити якість життя.

Область систем моніторингу та управління на основі IoT є предметом інтенсивних досліджень, що відображено в численних наукових публікаціях. У літературі містяться описи різноманітних систем моніторингу та керування на основі IoT, наприклад, в роботі [21] розглядають системи моніторингу навколишнього середовища на основі IoT, основною метою яких є надання даних про навколишнє середовище з віддалених місць через Інтернет. Представлена система використовує малопотужні бездротові датчики, підключені до Інтернету, які передають свої вимірювання на центральний сервер, що дозволяє віддалену візуалізацію та аналіз даних з будь-якого пристрою, підключеного до Інтернету. Розроблене рішення є комплексною системою, що охоплює фізичний рівень (сенсори та протокол зв'язку), а також управління та зберігання даних на кібернетичному рівні, покращуючи інтеграцію та взаємодію систем та підвищуючи ефективність нагляду.

В дослідженні [22] систематично оцінено сучасну літературу про розумні будинки та проведено огляд досліджень та розробки в цій галузі. Також представлено детальну картину розвитку та особливостей сучасних систем розумного дому, пропонуючи глибоке розуміння найновішого апаратного забезпечення та тенденцій. У статті також обговорюються важливі послуги та їх переваги, поточні та майбутні тенденції досліджень, а також труднощі та перешкоди, пов'язані з їх впровадженням. В роботі [23] досліджувалось використання технологій IoT у різних галузях промисловості, зокрема в охороні здоров'я, де IoT перетворює традиційні системи в інтелектуальні системи охорони здоров'я. В цій роботі представлено огляд послуг, програмного забезпечення та

методів, розроблених у сфері охорони здоров'я на основі Інтернету речей, підкреслюючи переваги їх впровадження. Висновки цієї роботи слугують базою довідковою інформацією для медичних працівників, дослідників і тих, хто цікавиться інтелектуальною охороною здоров'я.

В роботі [24] відображено важливість IoT (як ключової технології для систем моніторингу, що є результатом швидкого розвитку сенсорних технологій та конвергенції інформаційних технологій. В роботі представлено основу для структурного моніторингу здоров'я (SHM) з використанням технологій IoT, розглядаючи технології, що використовуються для реалізації систем SHM та стратегій маршрутизації даних у середовищі IoT. Також описуються рішення для великих даних, необхідні для керування великими та складними даними, створеними встановленими датчиками на конструкціях.

Окрім того концепція IoT [25] показує важливість енергоменеджменту в контексті зростаючого попиту на енергію для додатків IoT та сприяння зеленій промисловості. Вони представляють структуру для резервування підключення та підвищення енергоефективності в інтелектуальних галузях на основі Інтернету речей, охоплюючи рішення для виробництва енергії та керування пристроями з низьким енергоспоживанням. Також розглядаються тематичні дослідження, зокрема енергетичне планування на розумних фабриках і бездротова передача енергії, а також обговорюються невирішені проблеми, пов'язані з системами управління енергією на основі Інтернету речей.

В роботі [26] висвітлено зростаючий попит на виробництво енергії через збільшення населення та споживачів енергії у великих містах, підкреслюючи важливість систем енергетичного моніторингу, як ефективного рішення для оптимізації зростаючого попиту та споживання енергії. Розглядаються і класифікуються існуючі підходи до моніторингу енергії та сприятливі технології, такі як датчики та портативні пристрої, підкреслюючи практичні аспекти та проблеми продуктивності. Також проаналізовано ключові фактори дизайну, параметри налаштування та випадки використання в реальному світі, одночасно представляючи спостережувані моделі та статистику в цій галузі.

В роботі [27] висвітлено важливість моніторингу якості води під час переходу до розумного сільського господарства та автоматизації моніторингу ключових компонентів щоденних потреб, таких як вода.

В роботі [28] дослідники висвітлюють зростаючу проблему дефіциту безпечної води, спричинену зростанням населення, забрудненням і зміною клімату, підкреслюючи важливість моніторингу якості води, особливо в домашніх умовах. В даному дослідженні розглянуто недоліки традиційних лабораторних методів перевірки якості води та слабкі сторони систем на основі бездротової сенсорної мережі (WSN) щодо управління енергією, безпеки даних і покриття зв'язку. В роботі представлено рекомендації щодо проектування більш ефективних систем, які можуть отримати користь від розробок технологій Інтернету речей для покращення якості води в домашніх умовах.

В роботі [29] обговорюється зростаючі виробничі проблеми в автомобільній промисловості та збільшення споживання енергії, що призводить до збільшення витрат і викидів парникових газів, наголошуючи на важливості ефективного використання енергії. У даній роботі представлені різні методи та механізми управління енергією в інтелектуальній мережі з використанням комунікаційних технологій і протоколів. Пропонується інтеграцію електромобілів із розумною електричною мережею для заряджання та розряджання енергії та ефективного обміну інформацією. Також в роботі обговорюються стратегії інтеграції, впровадження мультиагентних систем для моніторингу потоків енергії та підтримки стабільності мережі, оптимального планування зарядки електромобілів та порівняння стандартних протоколів зв'язку в різних сценаріях.

В роботі [30] наведено огляд моделей інтелектуальних транспортних систем (ITS) з передовими системами інформації для подорожуючих (ATIS) і передовими системами управління дорожнім рухом (ATMS) для інтелектуального транспорту. Також описано технології ГІС та Інтернету речей у ITS із використанням датчиків і алгоритмів виявлення по всьому світу. Модель ATIS надає подорожуючим інформацію про регіональні подорожі, дозволяючи їм приймати обґрунтовані рішення щодо найкращого виду транспорту, маршруту, часу та вартості. Водночас

ATMS включає в себе керування даними в реальному часі, виявлення та відстеження транспортних засобів, зв'язок, системи змінних повідомлень, прогнозування трафіку та динамічний контроль руху. У статті розглядаються апаратні компоненти, такі як камери відеоспостереження, індуктивні петлі, магнітометричні, інфрачервоні, акустичні, ультразвукові, радарні та GPS датчики в системах ATIS і ATMS.

Дослідники в дослідженні [31] обговорюють роль технологій у сільському господарстві, звертаючи увагу на зростаючий попит на продукти харчування та труднощі, пов'язані з традиційними методами сільського господарства, такими як залежність від людської праці, мінливість погоди та брак знань і технологій. В цій роботі представляється огляд рішень на основі Інтернету речей, які можуть підвищити продуктивність сільськогосподарських культур, охоплюючи моніторинг стану ґрунту та рослин, інтелектуальне зрошення та прогнозування погоди в реальному часі. В цьому дослідженні представлено архітектуру IoT моделі розумного сільського господарства та схематичну модель автоматизованої сільськогосподарської системи, яка інтегрує дані з різних підсистем за допомогою алгоритмів штучного інтелекту (ШІ), що дозволяє надсилати сповіщення на телефони фермерів та ефективно керувати добривами, пестицидами та водою, що сприяє збільшенню врожаю.

Дослідники зробили огляд застосування технології IoT в сільськогосподарській автоматизації [32], аналізуючи впровадження в різних сільськогосподарських системах, таких як системи управління, моніторингу, контролю та безпілотні машини. Також обговорювалися комунікаційні технології на основі IoT, що використовуються в сільському господарстві, такі як Wi-Fi, LoRaWAN, мобільний зв'язок (2G, 3G, 4G), ZigBee і Bluetooth. Дослідники визначили обмеження та перспективи розвитку IoT у сільському господарстві, підкреслюючи, що майбутній розвиток комунікаційних технологій, таких як 5G, дозволить ширше та швидше використовувати IoT у різноманітних сільськогосподарських процесах, що сприятиме автоматизації сільського

господарства, покращенню якості та ефективності врожаю та зменшенню трудомісткості.

Автори в [33] підкреслюють важливість інфраструктури, особливо мостів, які є ключовими національним ресурсом і важливими для соціально-економічного розвитку. Також звертається увага на необхідність регулярного та ефективного аудиту мостів, які особливо чутливі до пошкоджень і вимагають частих перевірок, щоб уникнути ризику серйозних пошкоджень, які можуть загрожувати життю та майну. У статті розглядаються різні методи, які використовуються для перевірки стану мостів.

Дослідники в статті [34] представлено систематичний огляд сучасних знань про системи моніторингу якості повітря в приміщеннях на основі технології IoT. Дослідження зосереджено на таких аспектах дизайну, як типи датчиків, мікроконтролерів, архітектури та підключення. Огляд показує, що 70% досліджень зосереджені на моніторингу параметрів теплового комфорту, 65% на рівнях CO₂ і 27,5% на рівнях PM, тоді як 37,5% і 35% систем засновані на контролерах Arduino і Raspberry Pi. Також зазначено, що лише 22,5% досліджень використовували калібрування перед впровадженням системи, тоді як 72,5% систем заявляють про енергоефективність.

В роботі [35] обговорюють проблему заторів, які є поширеними майже в кожному міському районі по всьому світу, і важливість управління дорожнім рухом за допомогою інформації в реальному часі. Ця робота зосереджена на системах керування дорожнім рухом, заснованих на технології IoT, описує їх апаратні та програмні системи та аналізує їх роботу в міському контексті. Дослідження висвітлює, як переваги, так і недоліки систем контролю трафіку на основі IoT, наголошуючи на тому, що, хоча вони мають свої переваги, вони також мають певні обмеження, які необхідно враховувати при їх проектуванні та впровадженні.

Розробники в роботі [36] представили комплексний огляд застосування технологій IoT в розумних містах, які дозволяють інтегрувати різні пристрої та технології, усуваючи потребу втручання людини. В даній роботі описано

компоненти ландшафту міст, заснованих на IoT, і технології, що забезпечують роботу цих систем, включаючи архітектури, мережеві технології та алгоритми штучного інтелекту, що використовуються в системах розумних міст. У документі також розглядаються найпоширеніші практики та програми в різних сферах розумного міста та обговорюються проблеми впровадження систем Інтернету речей у містах, а також пропонуються контрзаходи.

В роботі [37] розглядаються існуючі рішення на основі Інтернету речей у системах управління відходами в розумних містах, підкреслюючи їх роль у забезпеченні сталого розвитку. Результати показують, що, незважаючи на те, що існуючі рішення схожі щодо своїх інтеграційних платформ із технологіями IoT, вони відрізняються датчиками та технологіями зв'язку, які використовуються, у багатьох дослідженнях із використанням Arduino Uno. В роботі підтримуються подальші дослідження розробки нових технологій і вдосконалення існуючих систем.

Дослідження [38] представляє огляд впливу технологій IoT на моніторинг навколишнього середовища, охоплюючи якість повітря, забруднення води та управління відходами. Також ця стаття надає аналіз розвитку систем моніторингу навколишнього середовища, які еволюціонували від базових систем віддаленого моніторингу до розширених систем моніторингу навколишнього середовища з використанням технології Інтернету речей і вдосконалених модулів датчиків. Дослідження зосереджено на оцінці досліджень щодо моніторингу якості повітря, води, відходів і загального забруднення навколишнього середовища та їх впливу на навколишнє середовище, класифікації досліджень за цілями, методами та результатами, а також аналізу використання сенсорних технологій, Інтернету речей і машинного навчання.

В роботі [39] представлено дослідження використання технологій Інтернету речей для модернізації систем вуличного освітлення, щоб зменшити втрати енергії та підвищити ефективність управління міським освітленням. Замість традиційних ручних систем, які призводять до значних витрат енергії, у статті описано, як IoT може автоматизувати вуличне освітлення, що має вирішальне значення для

вирішення енергетичної кризи. У роботі обговорюються різні датчики та компоненти, що використовуються в інтелектуальних системах освітлення, які, хоча часто прості, ефективні у створенні сучасних інтелектуальних систем керування освітленням.

1.2 Огляд і аналіз існуючих рішень

У сучасному динамічному технологічному середовищі IoT встановлює нові стандарти управління та контролю за різними системами. Системи моніторингу та контролю на основі IoT є ключовими в розумних будівлях, промисловості та секторі охорони здоров'я.

Наприклад, в реалізації систем прогнозованого обслуговування на основі IoT обговорюється система прогнозування пошкоджень у виробничому секторі, яка призначена для мінімізації часу простою та раннього виявлення пошкоджень виробничих машин. Ця система об'єднує різні датчики на основі IoT і використовує штучний інтелект для аналізу та прийняття рішень щодо потенційних точок збою. Система складається з мережі датчиків, розміщених у стратегічних точках, які постійно контролюють різноманітні робочі параметри машин на виробничій лінії, такі як температура, вологість, тиск, струм, вібрація, якість повітря та газ. Додатково використовувалися спеціальні датчики, такі як ультразвукові та фотоіонізаційні. Ці датчики підключені до центральної системи керування, яка збирає дані з багатьох джерел і передає їх у хмару для подальшої обробки та аналізу. Хмара даних використовує розширені алгоритми машинного навчання, для визначення закономірностей, прогнозування тенденцій і створення звітів для кінцевих користувачів. Описана система моніторингу навколишнього середовища на основі IoT є інноваційним рішенням, яке дозволяє здійснювати ефективний моніторинг і раннє виявлення несправностей.

В [40] розробники представили інноваційну систему на основі технології IoT, спеціально розроблену для прогнозного обслуговування круглов'язальних машин. Машини цього типу мають вирішальне значення для текстильної

промисловості, і їх незаплановані простої можуть спричинити значні витрати та порушити виробничий процес. Система, заснована на IoT, забезпечує моніторинг машин у режимі реального часу, ідентифікацію потенційних несправностей та ефективне планування технічного обслуговування, сприяючи підвищенню операційної ефективності та зниженню витрат, пов'язаних з обслуговуванням машин. Система складається з вдосконаленої друкованої плати (PCB) з мікроконтролером ESP32 і різними датчиками, включаючи датчики швидкості та зупинки машини. Ці датчики дозволяють системі відстежувати різні зупинки машини, такі як зупинка лайкри, зупинка воріт, зупинка холостого ходу (спричинена взаємодією людини), зупинка подавача, зупинка голки та готовий рулон. Дані, зібрані системою, зберігаються в базі даних MongoDB, що дозволяє проводити подальший аналіз і використовувати їх для прогнозованого обслуговування машини. Система була розгорнута на двох різних круглов'язальних машинах PAILONG, що працюють при температурі навколишнього середовища 26 градусів Цельсія. Використовуючи дані, зібрані з датчиків, система забезпечує комплексний моніторинг машин у реальному часі, забезпечуючи швидке виявлення потенційних несправностей та ефективне планування технічного обслуговування. Головною перевагою системи є її здатність забезпечувати моніторинг машин у режимі реального часу та оперативне реагування на будь-які порушення в роботі. Визначивши причину зупинки машини, оператори можуть негайно вжити заходів щодо її усунення, мінімізуючи час простою та запобігаючи потенційним поломкам. Крім того, технологія IoT дозволяє збирати й аналізувати великі обсяги даних, створюючи дедалі точніші прогнозні моделі, які можуть передбачати майбутні несправності машин з більшою точністю. Описана система IoT для прогнозованого обслуговування круглов'язальних машин є значним кроком вперед у галузі промислового обслуговування текстильної промисловості. Його впровадження може значно підвищити ефективність роботи, знизити витрати на технічне обслуговування машин і забезпечити безперебійність виробничих процесів за рахунок мінімізації незапланованих простоїв. Це чудовий приклад використання сучасних технологій

для підвищення продуктивності та конкурентоспроможності промислового сектора

Дослідники в роботі [41] підкреслили, що сучасна цифровізація виробничих процесів і зміщення ринкового попиту на продукти та послуги з нижчою вартістю призвели до появи систем продукт-послуги (PSS). В дослідженні підкреслюється важливість технічного обслуговування, ремонту та навчання в PSS у поєднанні з доповненою реальністю (AR). Аналіз великих наборів даних, створених датчиками, дозволяє швидко приймати рішення.

В останні роки питання енергоефективності стали пріоритетними для виробництва. Енергоспоживання значно впливає на витрати та якість виробництва. «Холодовий ланцюг» охоплює обладнання та процеси, які забезпечують належні умови для швидкопсувних товарів від виробництва до споживання. Системи охолодження та холодильного зберігання (RCSS) відіграють вирішальну роль у збереженні свіжості товарів, але споживають понад 50% енергії супермаркетів. Тому дані моніторингу є важливими для оптимізації роботи RCSS і запобігання поломок. Існуюча література представляє різні підходи до виявлення аномалій в системах охолодження. Є підходи, засновані на енергетичному аналізі, методах машинного навчання та системах діагностики несправностей.

Також існують інструменти для моніторингу споживання енергії та виявлення витоків холодоагенту. В роботі запропоновано систему моніторингу та прогнозованого обслуговування, яка використовує технології IoT та штучний інтелект. Запропонована архітектура систем, заснована на сенсорних мережах і хмарних обчисленнях. Компоненти включають сенсорні вузли, мережеву програму та інтерфейс користувача. Використовуються такі протоколи зв'язку, як HTTP, WebSocket і TCP. Розгортання передбачає інтеграцію баз даних, серверів і програмного забезпечення для забезпечення потоку даних між ними. Сенсорні вузли мають різні датчики та комунікаційні модулі, такі як Particle Device OS і XBee Wi-Fi. Інтерфейс користувача дозволяє візуалізувати дані та отримувати сповіщення про збої. Програмне забезпечення розроблено з використанням технологій NodeJS, MongoDB і WebSocket. Реалізація включає в себе розгортання

системи керування сенсорною мережею, яка обслуговує зв'язок між сенсорними вузлами та мережевим додатком. Розроблено прототип пристрою для збору даних з використанням мікроконтролерів і комунікаційних модулів.

1.3 Системи моніторингу та контролю для розумних міст на основі IoT

З динамічним розвитком урбанізації та зростанням міського населення міста в усьому світі стикаються з численними проблемами, пов'язаними з інфраструктурою, транспортом, управлінням ресурсами та якістю життя їхніх мешканців. Передові технології, включно з Інтернетом речей, все частіше використовуються для вирішення завдань підвищення ефективності міст, а також для того, щоб вони стали більш дружніми до мешканців. Розумні міста використовують IoT для створення інтегрованих систем моніторингу та контролю, які дозволяють збирати та аналізувати дані в реальному часі, покращуючи управління міською інфраструктурою та послугами. Системи IoT широко використовуються в розумних містах, від моніторингу дорожнього руху та управління громадським транспортом через інтелектуальне вуличне освітлення до управління відходами та моніторингу якості повітря. Завдяки датчикам, комунікаційним мережам і передовим аналітичним алгоритмам міста можуть ефективніше реагувати на потреби мешканців, оптимізувати споживання енергії та ресурсів, підвищити безпеку та комфорт життя.

В роботі [42] представляють інноваційний підхід до управління та моніторингу вуличного освітлення на основі технології IoT, що представляє собою прорив у міському освітленні в інфраструктурі, пропонуючи інтелектуальні, більш індивідуальні рішення, які враховують динамічні потреби сучасних міст. Використовуючи такі технології, як мікроконтролери NodeMCU, релейні модулі, резистори (LDR), а також датчики руху та навколишнього середовища, ці системи дозволяють динамічно регулювати рівень освітлення залежно від руху, активності пішоходів та умов освітлення.

Інтеграція віддаленого моніторингу та управління дозволяє міській владі завчасно виявляти проблеми, оптимізувати продуктивність і зменшити експлуатаційні витрати. За допомогою причинно-наслідкового аналізу складні алгоритми керування враховують дорожній рух, активність пішоходів, час доби та погодні умови для оптимізації рівня освітлення. Варто зазначити, що використовувані мікроконтролери NodeMCU є центральними блоками керування, що забезпечує зв'язок між реле, LDR та світлодіодними датчиками. LDR, стратегічно розміщені в міських районах, збирають дані про інтенсивність світла, а релейні модулі контролюють вуличне освітлення.

Схема роботи системи передбачає використання датчиків LDR для збору даних про інтенсивність світла, які потім обробляються мікроконтролером. На основі цих даних система приймає рішення щодо керування вуличним освітленням, що дозволяє оптимізувати енергоспоживання та мінімізувати експлуатаційні витрати. Вищезазначені елементи та інтеграція даних від багатьох датчиків і алгоритмів керування складають комплексну систему, яка забезпечує ефективне, енергозберігаюче та інтелектуальне вуличне освітлення, що позитивно впливає на якість життя мешканців та міське середовище.

В роботі [43] автори описали важливість проблеми забруднення навколишнього середовища відходами та необхідність моніторингу та управління ними для захисту навколишнього середовища. Накопичення сміття навколо вулиць може бути розсадником щурів і бліх, які переносять шкідливі хвороби, що призводить до епідемій і смертей. Тому для зменшення забруднення необхідно мати систему моніторингу відходів. У статті обговорюються різні підходи до моніторингу відходів на основі IoT, такі як інтелектуальні сміттєві баки з використанням інфрачервоних датчиків і датчиків навантаження, системи сигналізації на основі GSM, а також системи на основі RFID і Zigbee. Крім того, в роботі було представлено аналіз управління побутовими відходами в різних регіонах світу та обговорено переваги використання інтелектуальних рішень для моніторингу відходів. У рамках експериментів у статті була запропонована система моніторингу відходів на базі платформи Thingspeak з використанням

ультразвукових датчиків і датчиків навантаження, які збирають дані про рівень і вагу сміття в баку. Ці дані в режимі реального часу передаються на платформу Thingspeak, де вони оновлюються та стають загальнодоступними, дозволяючи користувачам відстежувати стан свого смітника в режимі онлайн. Експериментальні результати показують, що запропонована система ефективно контролює рівень і вагу відходів у контейнері, що дозволяє ефективно збирати та утилізувати відходи. Крім того, використання платформи Twitter для обміну оновленнями про стан смітника допомагає підвищити обізнаність громадськості щодо управління відходами та захисту навколишнього середовища. Висновки статті свідчать про те, що інтелектуальні системи моніторингу відходів можуть покращити чистоту міста, екологічну гігієну та розумне управління відходами, що є важливим кроком до сталого розвитку та захисту навколишнього середовища.

Крім того новим трендом є системи інтелектуального паркування [44]. Датчики, встановлені на паркувальних майданчиках, збирають дані про наявність паркувальних місць, які потім обробляються та аналізуються для ефективного керування додатками для паркування. Традиційним системам керування паркуванням часто потрібна допомога для точного визначення вільних місць, особливо на великих безкоштовних паркувальних майданчиках. Щоб подолати їх, в роботі пропонується використовувати інфрачервоні датчики для точного виявлення паркувальних місць, усуваючи проблеми, пов'язані з точністю, умовами освітлення та погодою.

Архітектура системи в додатку А, включає кілька етапів, включаючи розробку Android-додатку для бронювання паркувальних місць, ідентифікацію вільних місць за допомогою інфрачервоних датчиків, аутентифікацію користувачів за допомогою RFID-міток, класифікацію паркувальних місць на основі розміру автомобіля та навігацію до призначених місць за допомогою GPS. Зокрема, в основі системи лежать мікроконтролери Arduino Uno, Крім того, система надає інструменти візуалізації для власників паркувальних центрів для відстеження деталей бронювання, наявності місць та платіжної інформації. Деталі реалізації висвітлюють використання мікроконтролерів Arduino, зчитувачів RFID та модулів

Wi-Fi для забезпечення зв'язку між датчиками, мобільними додатками та серверними системами. Стаття також містить експериментальні результати, які демонструють ефективність запропонованої системи для оптимізації використання місця паркування та покращення досвіду користувача.

Таким чином, система розумного паркування на основі IoT, пропонує перспективне рішення для ефективного управління паркуванням у міських районах.

1.4 Постановка задачі

Аналіз існуючих IoT-рішень показав, що сучасні технології Інтернету речей широко використовуються для моніторингу та керування в різних сферах, включаючи міську інфраструктуру, промисловість, медицину, транспорт та енергетику. Однак, з огляду на швидкий розвиток технологій та зростання вимог до ефективності та безпеки таких систем, виникає необхідність у розробці інтегрованого програмно-апаратного рішення для збору, обробки та аналізу даних, яке буде базуватися на сучасних мікроконтролерах та методах машинного навчання.

Основна мета дослідження – створення ефективної, масштабованої та безпечної архітектури, яка дозволить в режимі реального часу здійснювати збір, обробку та передачу даних для подальшого аналізу.

Система має забезпечити безперервний моніторинг даних у реальному часі, з використанням сенсора руху MPU6050, модуля microSD для локального збереження інформації, OLED-дисплея для виведення повідомлень, а також засобів бездротової передачі даних на базі Wi-Fi та MQTT. Усі компоненти мають бути інтегровані в єдину систему з підтримкою модульної архітектури та взаємодії з користувачем через вбудований веб-інтерфейс.

Для досягнення поставленої мети необхідно виконати наступні завдання:

1. Розробити архітектуру програмно-апаратного модуля збору та обробки даних з врахуванням підтримки локального і віддаленого доступу до інформації.

2. Розробити алгоритми функціонування ключових компонентів модуля.
3. Розробити інформаційне забезпечення програмно-апаратного модуля.
4. Провести вибір програмних і апаратних засобів.
5. Реалізувати ключові модулі.
6. Реалізувати інтерфейс з користувачем.
7. Протестувати працездатність програмно-апаратного модуля.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ ОБРОБКИ ДАНИХ ІЗ ПРИСТРОЇВ ІОТ

2.1 Розробка архітектури програмно-апаратного модуля

Система, базуватиметься на мікроконтролері ESP32, який має двоядерну архітектуру (рисунок 2.1).



Рисунок 2.1 – Загальна архітектура системи

Це означає, що один процесор виконує одні завдання, а інший займається обробкою даних та комунікацією. Такий підхід дозволить значно підвищити продуктивність та зменшити затримки у виконанні завдань, що є критично важливим для систем Інтернету речей. Основне завдання цієї системи – це збір та обробка даних з різних сенсорів, їх передача на сервер або хмарну платформу, а також надання користувачеві зручного способу керування та моніторингу пристрою. Використання мікроконтролера ESP32 є виправданим, оскільки цей пристрій підтримує бездротові технології Wi-Fi та Bluetooth, має достатньо ресурсів для локальної обробки даних і може працювати з багатьма сенсорами.

Одним з основних компонентів системи є модуль збору даних, який включає підключені сенсори, наприклад, акселерометр, гіроскоп та магнітометр. Ці пристрої дозволяють фіксувати фізичні параметри навколишнього середовища або руху пристрою. Наприклад, акселерометр може вимірювати прискорення пристрою, що корисно в розумних системах контролю руху або у відстеженні вібрацій. Гіроскоп дозволяє визначати кутове положення пристрою, а магнітометр може допомагати в орієнтації щодо магнітного поля Землі. Зібрані дані обробляються безпосередньо на мікроконтролері ESP32, що дозволяє мінімізувати кількість інформації, яка передається мережею. Такий підхід зменшує

навантаження на сервер і дозволяє працювати навіть у мережах із низькою пропускнуою здатністю.

Комунікаційний модуль системи відповідає за передачу оброблених даних на сервер або інші IoT-пристрої. Для цього можна використати протокол MQTT, який є одним з найбільш ефективних методів передачі даних у середовищі Інтернету речей. MQTT працює за принципом публікації та підписки, що дозволяє кільком пристроям взаємодіяти між собою через єдиний сервер або брокер. Ця технологія є дуже енергоефективною і дозволяє працювати навіть на пристроях з обмеженими ресурсами. Крім того, підтримка Wi-Fi дозволяє підключати ESP32 до локальних мереж або Інтернету, що забезпечує зручність у використанні.

Ще одним важливим компонентом системи є веб-інтерфейс, який дозволяє користувачам отримувати доступ до даних у режимі реального часу. Використання вбудованого веб-сервера в ESP32 дозволяє переглядати отримані дані прямо з браузера, без необхідності встановлення додаткового програмного забезпечення. Це дуже зручно для мобільних пристроїв, адже користувач може підключитися до пристрою навіть без доступу до Інтернету, використовуючи локальну Wi-Fi-мережу. Веб-інтерфейс також дозволяє змінювати налаштування системи, наприклад, встановлювати пороги чутливості сенсорів або вибирати спосіб збереження даних.

Щоб зберігати отриману інформацію, система може використовувати внутрішню пам'ять ESP32, SD-карту або передавати дані на сервер для подальшого аналізу. Це залежить від конкретних вимог користувача. Якщо важливо отримувати історію вимірювань і здійснювати складну аналітику, найкращим варіантом буде використання хмарного сховища або локального сервера, який працює з базами даних. Якщо ж основна мета – це миттєва обробка даних, можна зберігати інформацію безпосередньо на пристрої та перезаписувати її, коли пам'ять заповнюється.

Ще один важливий аспект цієї системи – це використання багатопотокової обробки на базі FreeRTOS. Це дозволяє ефективно розподіляти обчислювальні ресурси між різними задачами, наприклад, паралельно отримувати дані з сенсорів,

обробляти їх та відправляти в мережу. Завдяки цьому система працюватиме швидше та більш стабільно, навіть якщо підключено кілька пристроїв одночасно. Крім того, FreeRTOS дозволяє реалізовувати енергозберігаючі режими, що є важливим для пристроїв, які працюють на батареї.

Для обробки та подальшого аналізу отриманих даних система може використовувати додатковий сервер, наприклад, Raspberry Pi. Це дозволяє розширити можливості системи, використовуючи більш складні алгоритми аналізу та машинного навчання. Наприклад, якщо система використовується для моніторингу вібрацій обладнання, можна навчити модель визначати аномалії та повідомляти про можливі несправності. Raspberry Pi також може слугувати шлюзом між пристроями ESP32 та хмарними сервісами, що дозволяє інтегрувати систему в більш масштабну інфраструктуру.

Безпека в таких системах також є важливим аспектом, оскільки передача даних через Wi-Fi робить її вразливою до атак. Для захисту інформації використовується шифрування, аутентифікація користувачів та захист від несанкціонованого доступу. Наприклад, застосування WPA2-шифрування при підключенні до Wi-Fi, та перевірка за допомогою авторизаційних токенів при доступі до веб-інтерфейсу.

Таким чином, архітектура такої системи включає кілька основних модулів: модуль збору даних, модуль комунікації, веб-інтерфейс, систему зберігання інформації, аналітичний сервер та систему безпеки. Такий підхід дозволяє створити гнучку IoT-систему, яка може адаптуватися до різних сценаріїв використання. Вона може застосовуватися для промислового моніторингу, розумного дому, систем безпеки, контролю транспорту та інших завдань, що вимагають збору та аналізу даних у реальному часі. Завдяки використанню сучасних технологій система є економічною, енергоефективною та легкою в налаштуванні, що робить її ідеальним вибором для широкого спектра застосувань.

2.2 Алгоритми функціонування компонент модуля

Враховуючи архітектуру програмно-апаратного модуля, функціонування системи побудовано на модульному підході з врахуванням архітектури ESP32 з двома ядрами, підтримкою FreeRTOS та протоколом MQTT. Алгоритми функціонування кожного програмного мають забезпечувати високу надійність, гнучкої масштабованості та можливість підтримки режиму реального часу обробки даних від пристроїв Інтернету речей (IoT).

Першою компонентою даного модуля є конфігураційний модуль. Після ввімкнення пристрою або перезапуску системи, першою активується частина програмного забезпечення, яка відповідає за налаштування. Його головним завданням є зчитувати необхідні параметри роботи пристрою з файлів конфігурації, що зберігаються на SD-карті. Ці параметри дозволяють модулю підключитися до бездротової мережі Wi-Fi, встановити зв'язок з MQTT-брокером та підготувати інші важливі налаштування (рисунок 2.2).



Рисунок 2.2 – Діаграма станів конфігураційного модуля

Файли конфігурації мають формат JSON. Наприклад, файл `wifi.conf` зберігає назву мережі та пароль, а `mqtt.conf` містить IP-адресу брокера та порт, через який буде встановлено зв'язок. Коли система читає ці файли, вона перевіряє їхню цілісність та правильність вмісту.

Якщо все зчитано успішно, конфігураційний модуль передає ці параметри іншим частинам програми. Якщо ж виникає помилка (файл пошкоджений або не знайдено ключових полів) на OLED-дисплеї з'являється відповідне повідомлення. У такому випадку пристрій не переходить до подальшої роботи, що захищає систему від помилкової поведінки.

До другої компоненти відноситься комунікаційний модуль, який відповідає за взаємодію пристрою з іншими системами, зокрема з керуючим пристроєм або сервером. Його основна функція це передача та прийом даних за допомогою протоколу MQTT, який забезпечує модель обміну повідомленнями за принципом "публікація-підписка" (рисунок 2.3).

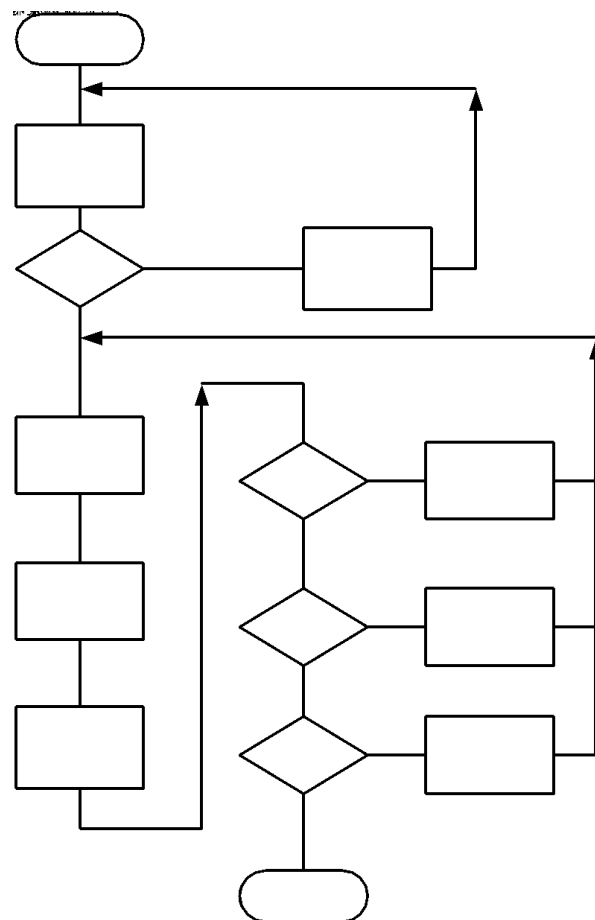


Рисунок 2.3 – Схема алгоритму роботи комунікаційного модуля

Після старту системи модуль автоматично ініціалізує підключення до MQTT-брокера. Коли з'єднання встановлено, модуль підписується на певні інформаційні канали (топіки), такі як: `esp/setting`, `esp/setting/real-time`, `esp/setting/frequency`. Через них керуюча система може надсилати команди до пристрою. Усі вхідні повідомлення обробляються спеціальною функцією `callback_mqtt`, яка визначає зміст повідомлення та виконує відповідну дію початку збору даних, перезапуск мікроконтролера або виведення діагностичної інформації на дисплей. Як тільки отримані дані з сенсорів, пристрій передає ці значення назад на брокер через топик `esp/mpu9250`. Для зменшення навантаження на систему повідомлення надсилаються в спрощеному текстовому форматі.

Третьою компонентною є модуль збору та обробки даних який відповідає збір та обробку даних від сенсорів. Робота модуля починається лише після того, як він отримує спеціальну команду `start` з MQTT-брокера. Ця команда запускає цикл збору даних. Частота вимірювання задається через окремий параметр (1000 вимірювань на секунду). Всі отримані значення спочатку тимчасово зберігаються у буфері оперативної пам'яті пристрою. Коли цикл завершено, дані автоматично записуються на SD-карту у форматі CSV (рисунок 2.4).

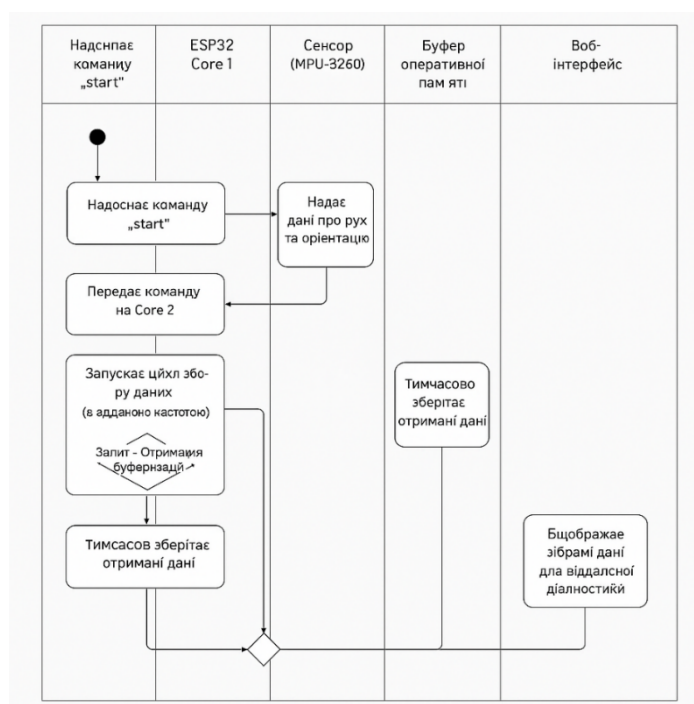


Рисунок 2.4 – Діаграма діяльності модуля збору та обробки даних

Четверта компонента дозволяє користувачу переглядати інформацію про стан пристрою через браузер. Після ввімкнення ESP32 автоматично запускається вбудований HTTP-сервер. Цей сервер працює локально і доступний за IP-адресою пристрою у Wi-Fi-мережі (рисунок 2.5).

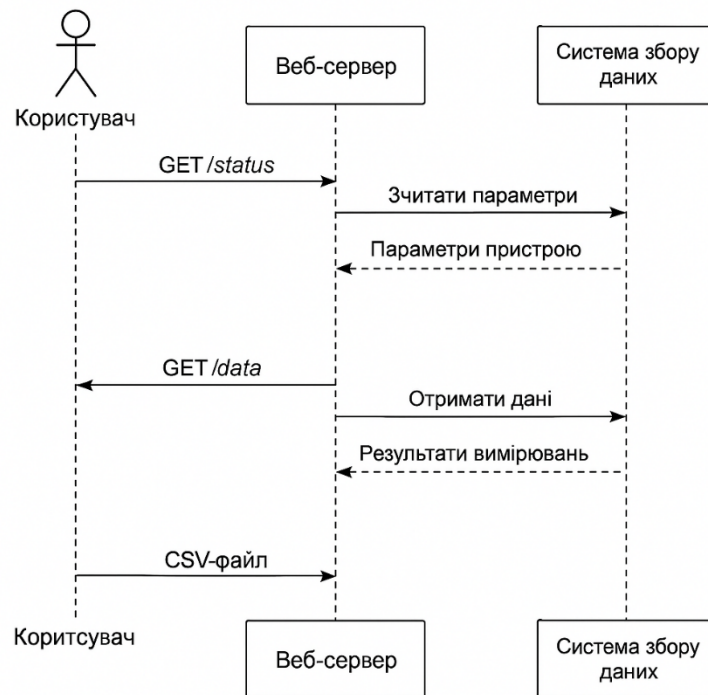


Рисунок 2.5 – Діаграма послідовності модуля модуль віддаленого доступу

На сервері налаштовано адреси (маршрути) і якщо користувач відкриває сторінку `/status`, він бачить поточні параметри температури, IP-адресу та кількість зібраних даних. Якщо перейти за адресою `/data`, можна завантажити файл з результатами вимірювань у форматі CSV.

Сторінки веб-інтерфейсу мають простий адаптивний дизайн і інформація оновлюється динамічно, що дозволяє бачити актуальні дані без потреби перезапустити пристрій. Завдяки цьому модулю користувач може переглядати та аналізувати дані без підключення до зовнішніх комп'ютерів або додаткового програмного забезпечення.

Останньою компонентною є модуль збереження даних (DataWriter), який відповідає за збереження виміряних даних на SD-карті. Основна задача модуля це

не допустити втрати важливої інформації у разі тимчасових збоїв у передачі через мережу або нестабільного з'єднання з брокером MQTT. Тому, незалежно від наявності зв'язку, всі результати вимірювань дублюються на локальний накопичувач — карту пам'яті формату SD. Також є діаграма послідовності програмно - апаратного модуля в додатку Б (рисунок 2.6).

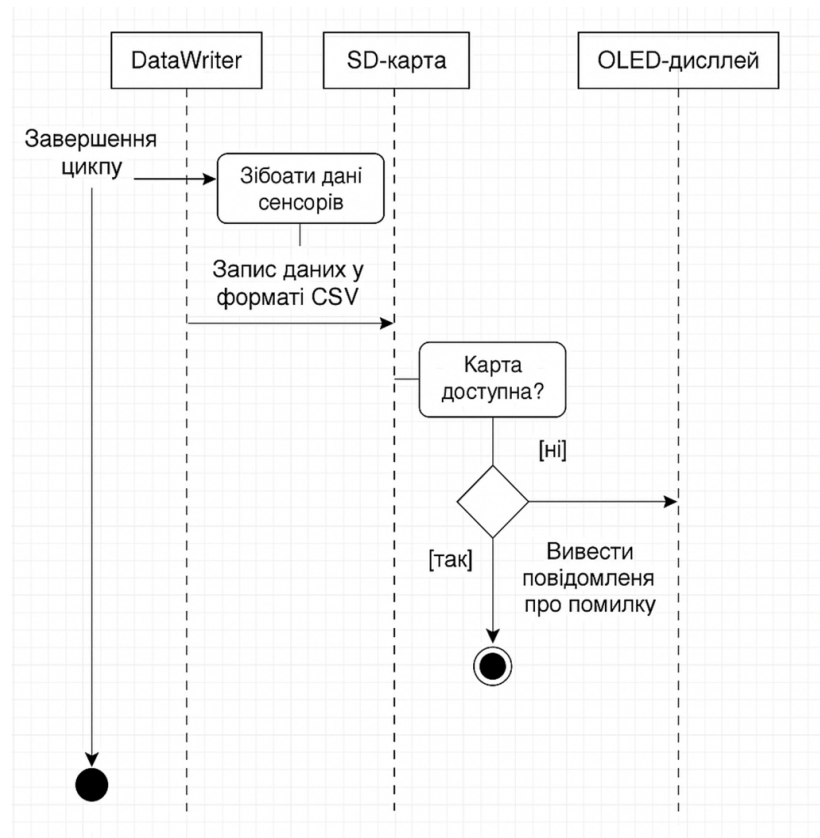


Рисунок 2.6 – Діаграма послідовності модуля збереження даних

Процес збереження активується після завершення кожного циклу збору даних. Дані, зібрані сенсорами формуються у вигляді звичайних текстових рядків. Кожен рядок містить часову мітку, ідентифікатор пристрою, а також значення, отримані з кожного сенсора. Дані записуються у форматі CSV. Крім того модуль перевіряє доступність SD-карти перед кожним записом. Якщо карта не виявлена або стала недоступною, модуль автоматично призупиняє збереження й виводить повідомлення про помилку на OLED-дисплеї пристрою. Це дозволяє користувачеві швидко виявити проблему та вжити необхідних дій Завдяки такій

логіці система гарантує збереження інформації навіть у непередбачуваних ситуаціях.

2.3 Інформаційне забезпечення програмно-апаратного модуля

До вхідної інформації належать дані, які надходять від сенсорів та керуючих команд. Основними джерелами є сенсор MPU-9250 та MQTT-повідомлення.

Сенсор MPU-9250 складається з трьох частин: акселерометра, гіроскопа і магнітометра. Акселерометр визначає прискорення пристрою у трьох напрямках: по осі X, Y та Z. Гіроскоп вимірює кутові швидкості обертання також у трьох напрямках. Магнітометр дає змогу виявити зміну магнітного поля, що корисно для орієнтації в просторі. Усі ці значення зчитуються з певною частотою, яку задає користувач.

Другим джерелом є MQTT-команди. Вони надходять через мережу Wi-Fi від керуючого пристрою (смартфон, комп'ютер, сервер). Основні команди:

- start починає вимірювання;
- stop завершує вимірювання;
- frequency встановлює частоту збору даних;
- real-time активує режим онлайн-передачі даних.

Отримавши ці команди, система реагує відповідно: починає зчитувати дані, змінює налаштування або припиняє роботу.

Також вихідна інформація зберігається у вигляді файлів формату CSV. Кожен запис у цьому файлі містить дату, час і всі виміряні параметри. Ці файли записуються на карту пам'яті (SD-карту), щоб їх можна було пізніше переглянути або проаналізувати.

Крім того, дані надсилаються через MQTT-брокер у вигляді повідомлень у топик `esp/mpu9250`. Це дає змогу віддалено спостерігати за параметрами в реальному часі.

Також на екрані OLED-дисплея відображається коротка інформація. Це може бути повідомлення про підключення до мережі, помилка читання файлу, активний режим роботи або кількість зібраних значень. Це додає додатковий

функціонал тому що стан пристрою можна моніторити без підключення до комп'ютера.

В таблиці 2.1 подано приклад основних параметрів, які використовуються в системі

Таблиця 2.1 – Основні вхідні та вихідні параметри системи

Параметр	Тип	Призначення
acc_x, acc_y, acc_z	Float	Прискорення по трьох осях
gyro_x, gyro_y, gyro_z	Float	Кутова швидкість
mag_x, mag_y, mag_z	Float	Магнітне поле
timestamp	Integer	Час зчитування
frequency	Integer	Частота вимірювань
start, stop	Команди	Керування роботою
mqtt_message	Текст	Дані, що передаються через MQTT
sample_data.csv	CSV-файл	Дані, збережені на SD-карті
OLED повідомлення	Текст	Статус і помилки для користувача

Така структура дозволить модулю працювати автономно або бути частиною більшої IoT-системи, передаючи інформацію на сервер або іншу систему збору та обробки даних.

На наступному етапі було визначено головні сутності системи, їхні зв'язки та структура інформації, яка обробляється. Програмно-апаратний модуль працює з різними джерелами даних. Це, зокрема, сенсорний модуль (MPU-9250), керуючі команди з MQTT-брокера, дані, що надходять у вигляді параметрів конфігурації, а також внутрішні події пристрою (час початку або завершення вимірювань). Усі ці дані логічно об'єднуються у структури, які можуть бути збережені або передані в інші підсистеми.

Було виділено наступні сутності:

– Сенсорне вимірювання містить поля timestamp, acc_x, acc_y, acc_z, gyro_x, gyro_y, gyro_z, mag_x, mag_y, mag_z.

- Команда користувача включає тип команди (start, stop, frequency), час надходження, параметри (наприклад, задана частота).
- Сеанс вимірювання містить дату початку, тривалість, кількість записів, ідентифікатор збереженого файлу.

Дані сутності взаємопов'язані, тобто один сеанс вимірювання може містити багато сенсорних записів. Команди користувача запускають або зупиняють сеанс, або змінюють параметри збору даних. Дані зв'язки представлені на рисунку 2.7.

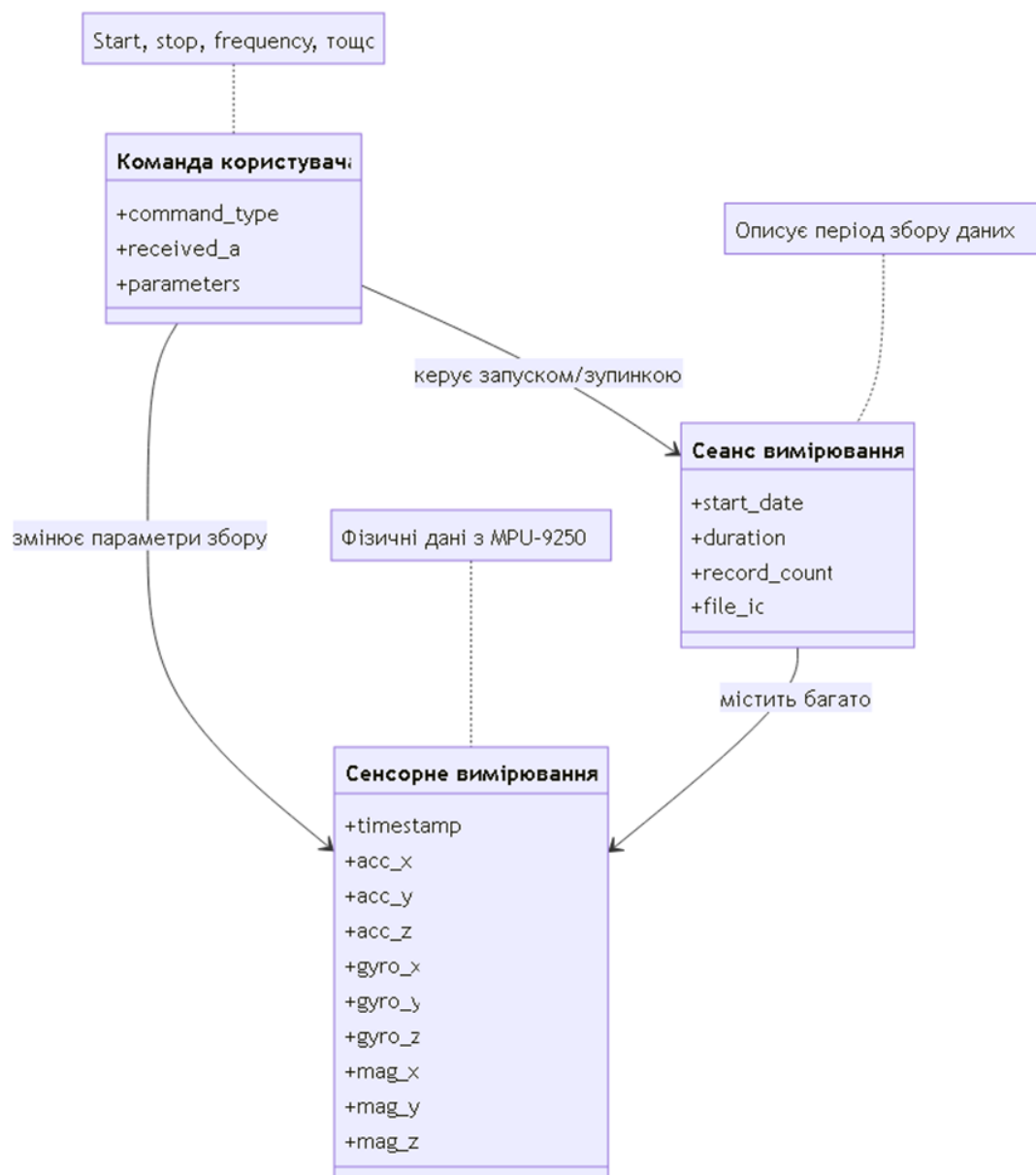


Рисунок 2.7 – Діаграма класів інфологічної моделі системи збору даних

При проектуванні було враховано структуру збереження інформації. Зібрані дані формуються у вигляді таблиць або структур CSV-файлів. Кожна сутність має свій унікальний ідентифікатор, що спрощує зв'язування між записами.

Для збереження результатів вимірювань використовується SD-карта. Тому фізична модель даних, визначає, в якому вигляді і де саме зберігаються файли з інформацією.

Кожен сеанс збору даних формує окремий файл. Ім'я файлу створюється автоматично на основі дати та часу початку вимірювання, що дозволяє легко впорядковувати та ідентифікувати файли без плутанини. Усі файли зберігаються в одній директорії на SD-карті.

Формат файлу має наступну структуру де кожен рядок містить одне вимірювання, а всі значення розділені комами, як показано на рисунку 2.8.

```
timestamp,acc_x,acc_y,acc_z,gyro_x,gyro_y,gyro_z,mag_x,mag_y,mag_z  
1683970812000,0.02,0.01,9.81,0.01,0.00,0.00,0.12,0.07,0.33
```

Рисунок 2.8 – Структура даних в файлі

Всі значення записуються в кодуванні UTF-8, що дозволяє відкривати файл у будь-якому текстовому редакторі або програмі типу Excel.

Якщо SD-карта не вставлена або стала недоступною під час запису, система виводить повідомлення про помилку на OLED-дисплей. Запис тимчасово зупиняється до усунення проблеми. Такий функціонал дозволяє користувачеві дізнатися, що щось пішло не так, і уникнути втрати даних.

Частота запису визначається користувачем через параметр frequency. При значенні 100 Гц в файл записується 100 рядків на секунду. Цей параметр впливає на обсяг файлу, і на точність даних. Система оптимізована для запису блоками, щоб зменшити знос SD-карти та пришвидшити роботу.

Така модель (рисунок 2.9) забезпечує надійне збереження даних і зручність у подальшій роботі з ними.

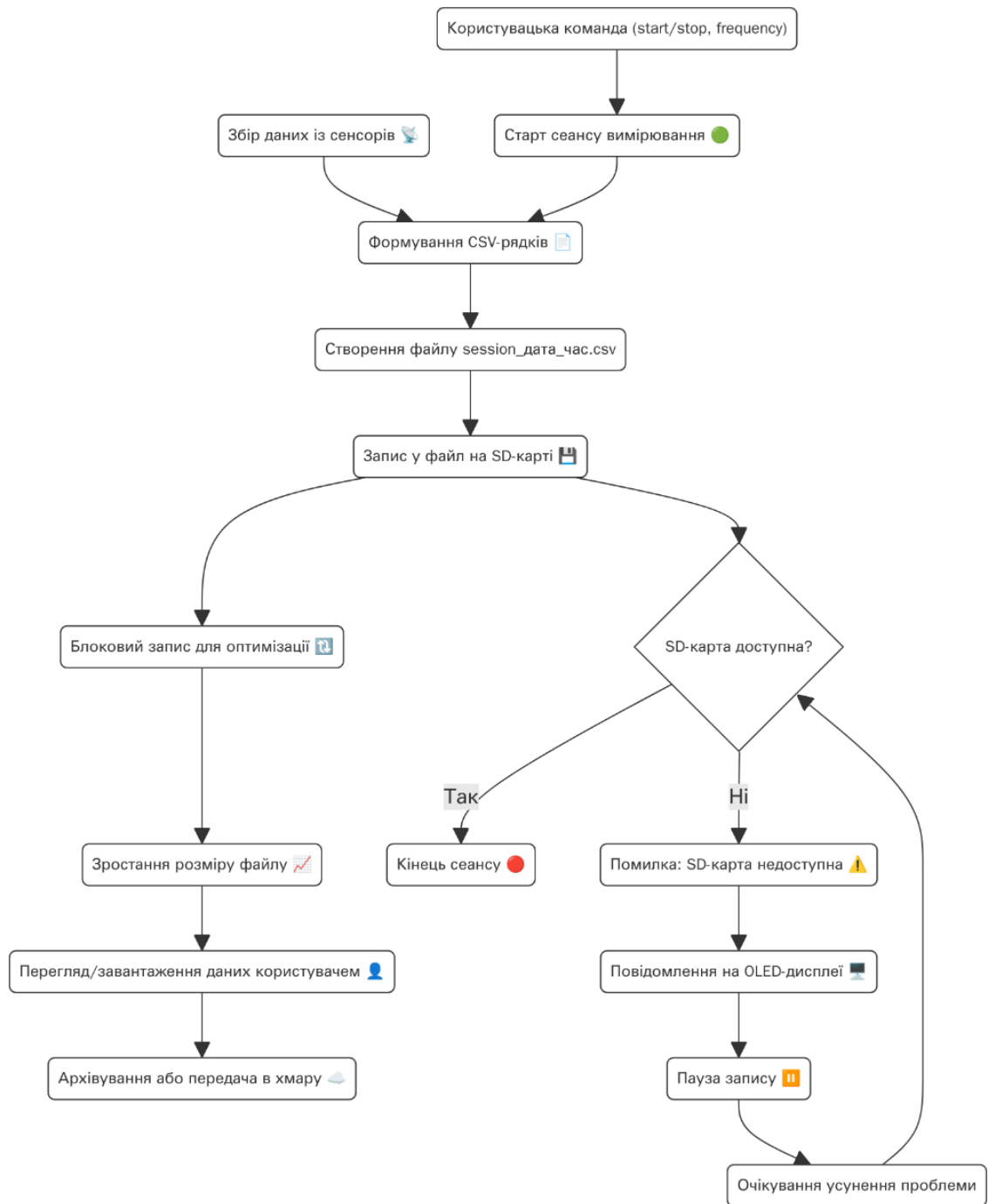


Рисунок 2.9 – Розширена діаграма фізичної моделі збереження даних з додатковими вузлами

Даний підхід дозволяє також розширювати структуру папок за сеансами або користувачами. Крім того можлива інтеграція із зовнішнім архівом або резервним копіюванням через мережу для підвищення надійності зберігання.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ ОБРОБКИ ДАНИХ ІЗ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ

3.1 Вибір програмних і апаратних засобів

На основі 2 розділу де була описана архітектура модуля та основні алгоритми функціонування було був проведений вибір відповідних апаратних і програмних компонентів. Основними критеріями, за якими відбувався цей вибір, були:

- сумісність між усіма елементами;
- надійність під час тривалого автономного функціонування;
- підтримка сучасних інтерфейсів для з'єднання;
- енергоефективність та доступність на ринку.

Враховуючи характер завдань, які має виконувати система, було вирішено розробити рішення на основі мікроконтролера ESP32, який поєднує у собі високу продуктивність, двоядерну архітектуру, бездротову комунікацію через Wi-Fi, а також доступ до широкого набору цифрових і аналогових входів/виходів (рисунок 3.1).

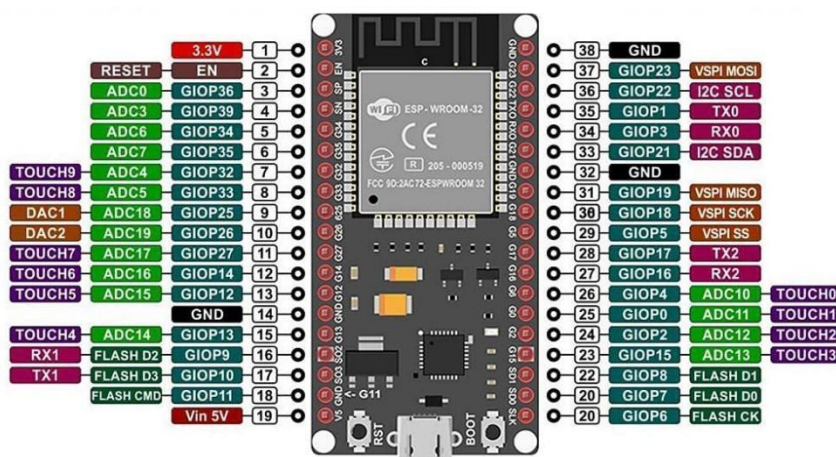


Рисунок 3.1 – Загальна конфігурація портів вводу/виводу модуля розробки

Таке поєднання дозволяє ефективно реалізувати одночасну роботу з сенсорами, передачею даних та локальним збереженням. Важливим фактором при

виборі була наявність великої кількості прикладів, документації та підтримка спільноти.

Основу апаратної частини складає мікроконтролер ESP32 DevKit v1. Він працює на частоті до 240 МГц, має два ядра та підтримує одразу кілька протоколів комунікації, таких як UART, I2C та SPI, що значно спрощує підключення зовнішніх пристроїв. Крім того даний контролер буде комунікувати з сенсором MPU-9250 (рисунок 3.2) , який поєднує в собі функції акселерометра, гіроскопа та магнітометра.

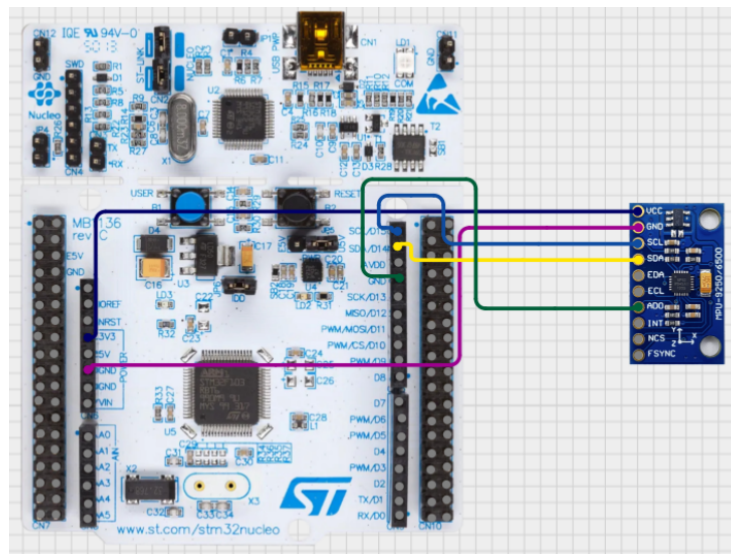


Рисунок 3.2 – Загальна конфігурація підключення MPU-9250

Це дозволяє системі збирати просторові дані, включаючи напрям руху, вібрацію, зміну орієнтації та положення. Підключення сенсора здійснюється через інтерфейс I2C. Для збереження зібраних даних використовується microSD-карта, яка працює через SPI-шину. Вона дозволяє записувати інформацію навіть у випадках втрати з'єднання з мережею або MQTT-брокером. Це забезпечує резервне збереження та дає змогу опрацювати результати згодом. Для відображення ключової інформації про стан пристрою використовується OLED-дисплей з роздільною здатністю 128x64 пікселі. Він підключається через I2C та дозволяє інформувати користувача про успішність з'єднання з мережею,

активний режим роботи, поточну частоту вимірювань, помилки під час збереження на SD-карту тощо.

При виборі програмного забезпечення було прийнято рішення використовувати середовище Arduino IDE (рисунк 3.3), оскільки воно є зручним, підтримує мікроконтролери ESP32, має розширену бібліотеку прикладів та дозволяє швидко компілювати і завантажувати прошивку.

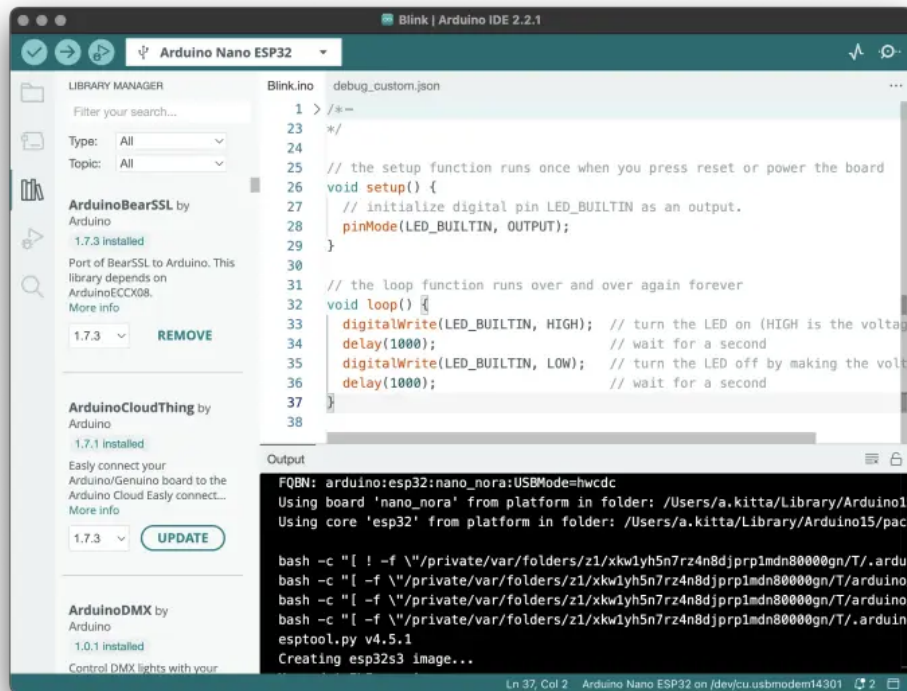


Рисунок 3.3 – Середовище розробки Arduino IDE v2

В даному середовищі були використані такі бібліотеки, як PubSubClient для роботи з протоколом MQTT, бібліотека SD для роботи з картою пам'яті, Wire для обміну даними через I2C та Adafruit_SSD1306 для керування OLED-дисплеєм. Ці бібліотеки є стабільними, підтримуються спільнотою та інтегруються без помилок в проєкт на основі ESP32. Реалізація коду буде враховувати багатопоточність FreeRTOS, яка вже вбудована в ESP32, що дозволяє рознести задачі збору та передачі даних по різних ядрах процесора. Таким чином, забезпечується незалежна робота окремих процесів без блокування системи.

Для обміну даними між пристроєм і сервером або користувачем було використано протокол MQTT, що є одним із найефективніших у сфері Інтернету речей. MQTT дозволяє мінімізувати навантаження на мережу та мікроконтролер, оскільки працює за принципом публікації і підписки. Для тестування використовувався публічний брокер HiveMQ (рисунок 3.4), але в разі потреби можливо встановити локальний брокер Mosquitto.

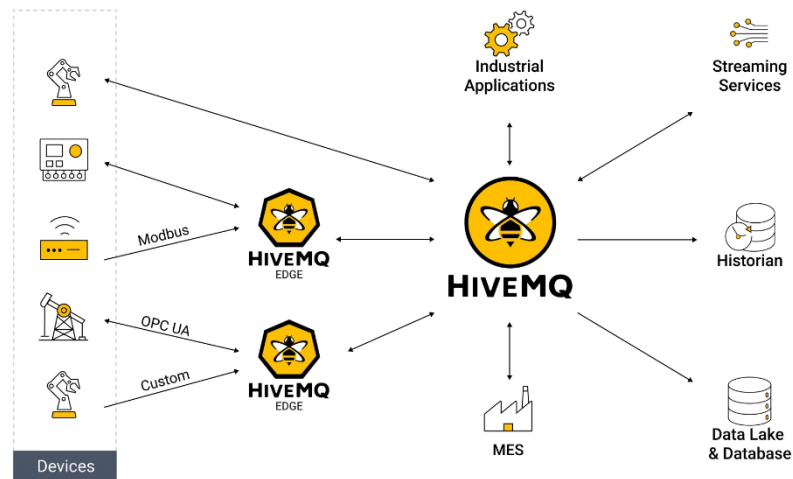


Рисунок 3.4 – HiveMQ

Передача відбувається у вигляді JSON або простих текстових повідомлень. На стороні клієнта передбачено використання веб-браузера. ESP32 має можливість працювати як веб-сервер, тому після підключення до локальної мережі користувач може зайти на сторінку пристрою та переглянути поточний стан, дані вимірювань або завантажити CSV-файл. Завдяки цьому не потрібно встановлювати додаткових програм. Такий підхід робить систему максимально універсальною та зручною для подальшого впровадження і використання.

3.2 Реалізація ключових модулів

Система побудована на основі кількох незалежних програмних модулів. Кожен з них виконує свою функцію — від зчитування температури до запису даних у файл і взаємодії з користувачем через кнопку.

Модуль ініціалізації та конфігурації виконується при запуску пристрою, який перевіряє наявність SD-карти, яка потрібна для збереження результатів. Якщо карта доступна, система повідомляє, що ініціалізація пройшла успішно. Якщо ні — виводиться повідомлення про помилку, і програма зупиняється. Приклад коду представлено на рисунку 3.5.

```
if (SD.begin(chipSelect)) {  
    Serial.println("SD card is initialized.");  
} else {  
    Serial.println("Error initializing SD card.");  
    return;  
}
```

Рисунок 3.5 – Перевірка підключення SD карти

Функція намагається ініціалізувати SD-карту на вказаному піні. Якщо ініціалізація вдалася, виводиться повідомлення в монітор порту. Якщо ні — пристрій не переходить до подальших дій, що запобігає втраті даних.

Після чого запускається конфігураційний модуль. Його основна функція полягає в зчитуванні параметрів мережі Wi-Fi та MQTT-брокера. Це дозволяє зробити систему гнучкою, оскільки параметри можна змінити без перепрошивки змінивши тільки записи файл на SD-карті (рисунок 3.6).

```
File wifiFile = SD.open("/wifi.conf");  
if (wifiFile) {  
    ssid = wifiFile.readStringUntil('\n');  
    password = wifiFile.readStringUntil('\n');  
} else {  
    ssid = "defaultSSID";  
    password = "defaultPASS";  
}  
WiFi.begin(ssid.c_str(), password.c_str());
```

Рисунок 3.6 – Зчитування конфігурації з SD карти

Функція намагається відкрити файл `wifi.conf`, зчитує з нього перший рядок як ім'я мережі (SSID), другий — як пароль. Якщо файл не знайдено або пошкоджено, використовуються стандартні значення.

Якщо все відпрацювало запускається модуль збору даних з MPU-9250 який відповідає за зчитування тривимірних даних з сенсора. Він підключений через шину I2C, що дозволяє ефективно передавати всі значення одразу (рисунок 3.7).

```
IMU.readSensor();  
float ax = IMU.getAccelX_mss();  
float gy = IMU.getGyroY_rads();  
float mz = IMU.getMagZ_uT();
```

Рисунок 3.7 – Зчитування вимірів з сенсора

Метод readSensor() зчитує усі показники сенсора. Значення прискорення (ax), кутової швидкості (gy) та магнітного поля (mz) зберігаються у змінних для подальшої обробки. Залежно від налаштувань бібліотеки, одиниці виміру можуть бути у м/с², рад/с, мікротеслах відповідно.

Після чого починає працювати модуль збереження, який відповідає за створення та запис даних у CSV-файл на карті пам'яті. Кожен сеанс вимірювання створює новий файл, що забезпечує збереження результатів (рисунок 3.8).

```
File dataFile = SD.open("/data/session_001.csv", FILE_WRITE);  
dataFile.println("timestamp,acc_x,gyro_y,mag_z");  
dataFile.println("1689250212,0.12,-0.02,0.35");  
dataFile.close();
```

Рисунок 3.8 – Запис даних

Файл відкривається у режимі дозапису. У перший рядок виводиться заголовок CSV-файлу, далі — значення із сенсора разом із часовою міткою. У разі відсутності карти SD виводиться помилка, а збереження не виконується.

Для забезпечення комунікації через мережу використовується модуль MQTT-брокера, він підписується на командні топіки та публікує зібрані дані в окремий канал (рисунок 3.9).

```
client.subscribe("esp/setting/start");  
client.publish("esp/mpu9250", "{ \"acc_x\": 0.12 }");
```

Рисунок 3.9 – Публікування даних

Підписка на топик `esp/setting/start` дозволяє пристрою реагувати на команди від сервера. У публікації використовується формат JSON для передачі даних у топик `esp/mru9250`, що забезпечує сумісність з більшістю IoT-платформ.

Крім того вбудований веб-сервер дозволяє користувачу отримати дані про стан пристрою безпосередньо через браузер (рисунок 3.10). Він працює у локальній мережі.

```
server.on("/status", HTTP_GET, []() {  
  server.send(200, "text/html", "<h2>Status OK</h2><p>Samples: 231</p>");  
});
```

Рисунок 3.10 – Публікування даних

Сервер створює сторінку `/status`, яка виводить поточний стан пристрою та кількість зібраних зразків. Ця інформація оновлюється при кожному запиті користувача.

Ще одним компонентом індикації роботи модуля є компонента для роботи з OLED-дисплеєм і використовується для локального виводу повідомлень про стан Wi-Fi, MQTT, наявність помилок. Така компоновка дозволяє індикацію в складних умовах або при відсутності комп'ютера.

3.3 Інтерфейс взаємодії з користувачем

Інтерфейс користувача є важливою частиною розробленого програмно-апаратного модуля. Він забезпечує зручний доступ до даних, які збирає пристрій, а також дозволяє моніторити поточний стан системи. Було реалізовано два основних типи інтерфейсів – це веб-інтерфейс через браузер та локальний OLED-дисплей. Разом вони створюють середовище як для віддаленої роботи з пристроєм, так і для контролю в реальному часі без додаткового обладнання.

Мікроконтролер ESP32 містить у собі функціонал для створення веб-сервера. Це дає змогу отримувати інформацію з пристрою через звичайний браузер. Було реалізовано два маршрути `/status` і `/data`.

`/status` — відображає базову інформацію про стан системи, такі як стан підключення до Wi-Fi, кількість зібраних зразків, статус SD-карти;

`/data` — дозволяє завантажити файл з даними у форматі CSV.

Код реалізації даного інтерфейсу представлено на рисунку 3.11.

```
server.on("/status", HTTP_GET, []() {  
    String html = "<!DOCTYPE html><html><head><meta charset='UTF-8'>";  
    html += "<title>Стан пристрою</title></head><body>";  
    html += "<h2>Пристрій активний</h2>";  
    html += "<p>Кількість зразків: " + String(sampleCount) + "</p>";  
    html += "<p>WiFi: Підключено</p>";  
    html += "<p>MQTT: Успішно</p>";  
    html += "</body></html>";  
    server.send(200, "text/html", html);  
});
```

Рисунок 3.11 – Блок генерації веб-інтерфейсу

Цей блок коду відповідає за генерацію HTML-сторінки. За запитом на `/status` сервер формує сторінку з інформацією про кількість зібраних даних, стан підключення до мережі, доступність MQTT. HTML-код генерується динамічно, тому значення `sampleCount`.

Для зручного доступу до зібраної інформації реалізовано окремий маршрут `/data`, який дозволяє завантажити CSV-файл (рисунок 3.12).

```
server.on("/data", HTTP_GET, []() {  
    File file = SD.open("/data/session_001.csv");  
    if (file) {  
        server.streamFile(file, "text/csv");  
        file.close();  
    } else {  
        server.send(404, "text/plain", "файл не знайдено");  
    }  
});
```

Рисунок 3.12 – Блок скачування файлу з даними

Коли користувач відкриває /data, пристрій відкриває CSV-файл на SD-карті та надсилає його у відповідь. Якщо файл відсутній або SD-карта недоступна, повертається повідомлення про помилку.

Крім веб-інтерфейсу, пристрій використовує OLED-дисплей для відображення коротких повідомлень. Це дозволяє побачити, чи підключено Wi-Fi або чи виникла помилка, як показано на рисунку 3.13.

```
display.clearDisplay();  
display.setCursor(0, 0);  
display.setTextSize(1);  
display.print("WiFi: OK");  
display.setCursor(0, 10);  
display.print("MQTT: Connected");  
display.setCursor(0, 20);  
display.print("Samples: ");  
display.print(sampleCount);  
display.display();
```

Рисунок 3.13 – Блок виводу повідомлень на OLED-дисплей

На дисплей виводиться інформація в три рядки де відображається статус Wi-Fi, статус MQTT та кількість зібраних зразків. Це корисно під час налаштування пристрою або для оперативного контролю в умовах обмеженого доступу до комп'ютера.

Така компоновка забезпечує два незалежні і доповнюючі один одного канали взаємодії з користувачем за допомогою веб-інтерфейсу для повного контролю та роботи через браузер, і OLED-дисплею для швидкої перевірки статусу. Код програмно апаратного модуля, додаток В.

3.4 Тестування функціональності

Крі опису усіх компонентів апаратно-програмного модуля, було також проведено тестування апаратної та програмної частини. На першому етапі перевірялось, як працюють всі апаратні компоненти, які були підключені до плати ESP32. Зокрема, сенсор руху MPU6050, дисплей OLED і карта пам'яті microSD.

Перевірка проводилась у два етапи: спочатку окремо кожного модуля, а потім — усіх разом у складі повноцінної системи.

Спочатку тестування проводилось в онлайн-середовищі Wokwi, де можна побачити, як працює схема без реального обладнання. Після цього ту саму схему було зібрано на макетній платі з реальною ESP32, сенсором, дисплеєм та SD-картою (рисунок 3.14).

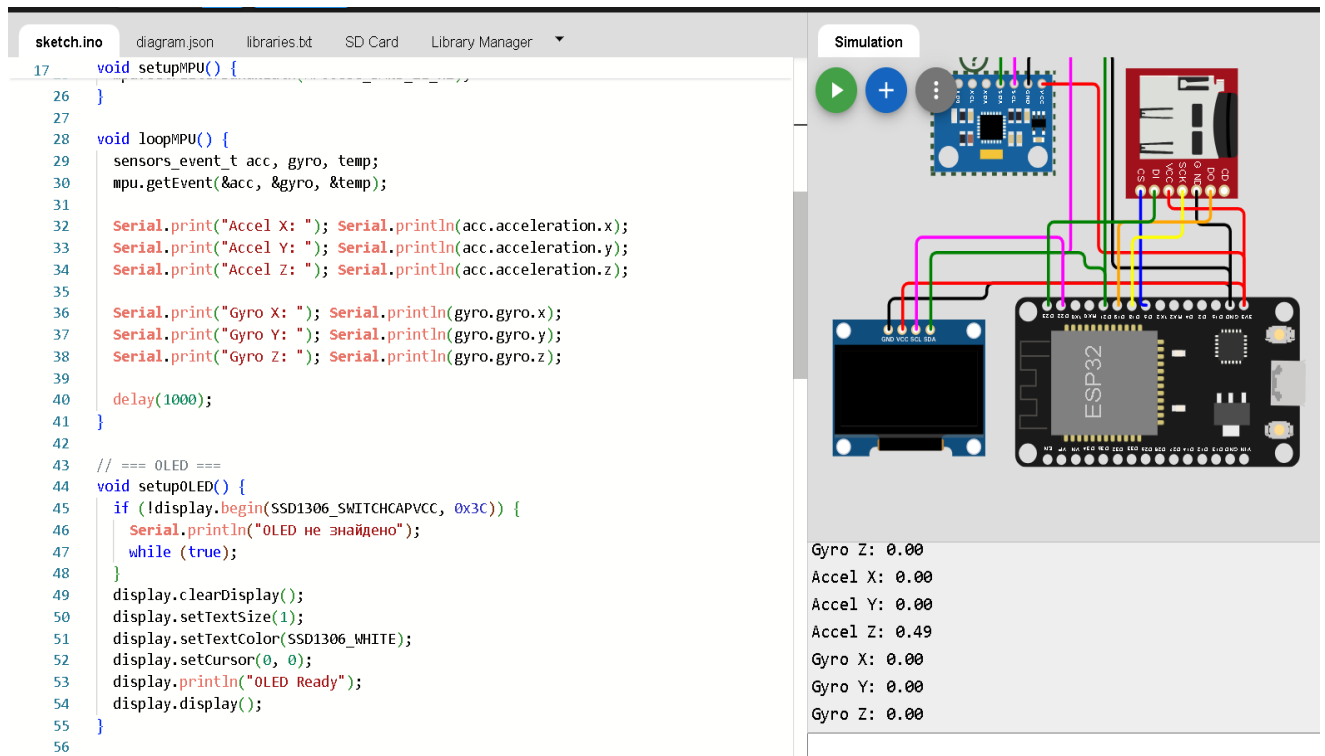


Рисунок 3.14 – Зібраний блок апаратної частини модуля

Для кожного модуля ми було перевірено базову функціональність. При роботі з MPU6050 аналізувалось, чи змінюються показники при русі. На OLED-дисплеї перевірялось, чи правильно з'являється текст, наприклад повідомлення про підключення. Для SD-карти перевірялось чи створюється файл, і чи записується туди текст.

Також було проведено тестування на можливість одночасної роботи усіх цих компонентів. Даний крок є обов'язковим, тому що MPU6050 і OLED використовують один і той самий інтерфейс I2C, а SD-карта — SPI (таблиця 3.1).

Таблиця 3.1 – Результати тестування апаратної частини

Компонент	Що перевіряли	Результат
MPU6050	Зчитування при русі пристрою	Значення змінюються
OLED-дисплей	Вивід тексту	Виводить повідомлення
SD-карта	Створення та запис файлу	Дані записані
Робота всіх модулів одночасно	Немає конфліктів I2C/SPI	Працює стабільно

В процесі було виявлено наступні проблеми, а саме сенсор іноді не відповідав, а OLED-дисплей не виводив повідомлення. Після налагодження затримок, перевірки адрес і пінів апаратна частина стала працювати стабільно. Для коду ми також оптимізували `setup()`, зробивши його модульним — це спростило логіку програми (таблиця 3.2).

Таблиця 3.2 – Проблеми та рішення під час тестування апаратної частини

Проблема	Рішення
MPU6050 не відповідав	Додано затримку перед ініціалізацією
OLED не вмикався	Уточнено адресу модуля (0x3C)
SD не ініціалізувалась	Перевірено підключення, CS = GPIO5
Конфлікти <code>setup()</code>	Створено модульні <code>setupMPU()</code> , <code>setupSD()</code>

Після завершення тестування апаратної частини, потрібно було провести подібні операції і з серверною частиною, яка відповідає за зв'язок через інтернет. Це включає веб-інтерфейс та роботу з протоколом MQTT. Через веб-інтерфейс користувач може переглянути дані, а за допомогою MQTT — надіслати команди пристрою, наприклад, щоб почати чи зупинити збір даних.

Спочатку перевірялось, чи ESP32 правильно підключається до Wi-Fi (рисунок 3.15). Якщо файл конфігурації був відсутній або пошкоджений, система мала показати помилку на OLED-дисплеї.

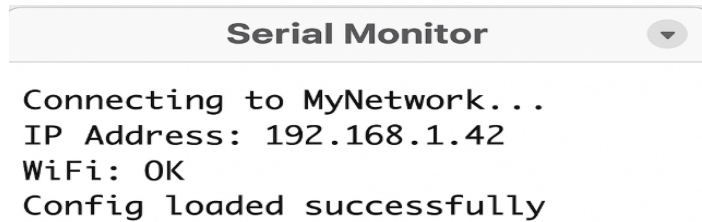


Рисунок 3.15 – Тест Wi-Fi

Потім перевірялась доступність сторінки /status (рисунок 3.16), яка повинна відображати поточний стан пристрою, зокрема IP-адресу, стан Wi-Fi та кількість зібраних зразків.

Після цього було проведено захід на сторінку /data, щоб завантажити CSV-файл з результатами вимірювань.

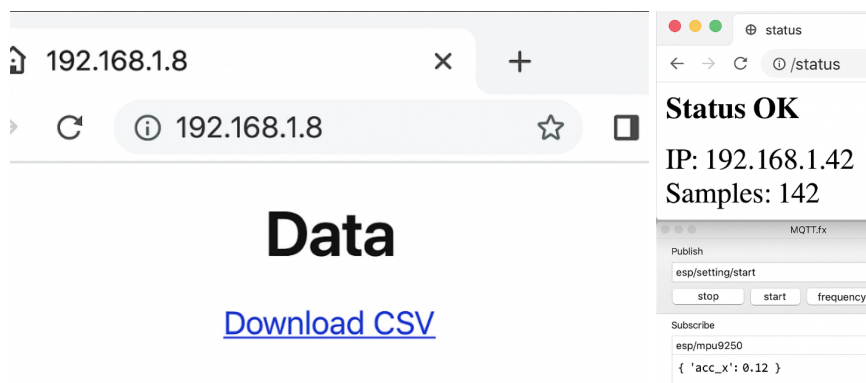


Рисунок 3.16 – Тест роботи генерації веб-сторінок

MQTT-зв'язок тестувався за допомогою клієнта MQTT.fx (рисунок 3.17).

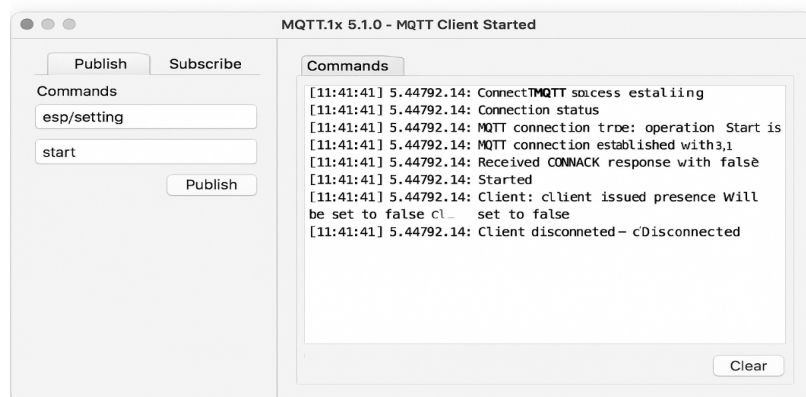


Рисунок 3.17 – Тестування MQTT

При тесті надсилались команди start, stop, frequency=500, щоб переконатися, що пристрій реагує на них. У відповідь пристрій починав збір даних, змінював частоту або зупиняв роботу. Також було протестовано, чи публікуються дані у відповідний MQTT-топік, і чи вони надходять у правильному форматі.

Під час тестування серверної частини було виявлено кілька типових проблем, пов'язаних із мережею та обробкою команд. Наприклад при відсутності файлу wifi.conf пристрій не міг підключитися до Wi-Fi, що блокувало подальшу роботу веб-інтерфейсу та MQTT-зв'язку. Це потребувало реалізації механізму резервних налаштувань та інформування користувача через дисплей. Також спостерігались періодичні втрати з'єднання з MQTT-брокером, особливо при нестабільній мережі і для цього було реалізовано функцію повторних автоматичних підключень.

Ще однією проблемою була обробка команди frequency, коли пристрій не реагував на некоректні значення. В результаті було впроваджено додаткову перевірку вхідних параметрів, щоб уникнути збоїв у роботі (таблиця 3.3).

Таблиця 3.3 - Проблеми та рішення під час тестування серверної частини

Проблема	Рішення
Відсутній wifi.conf	При відсутності конфігураційного файлу система автоматично використовує стандартні параметри і виводить повідомлення про помилку на дисплей.
Нестабільне з'єднання з MQTT	Додано функцію повторних підключень до брокера MQTT у разі втрати зв'язку.

Команда frequency не спрацьовувала	В коді реалізовано перевірку (валідацію) значення частоти, яке надходить через MQTT.
------------------------------------	--

В загальному система показала стабільну роботу, команди оброблялися правильно, з'єднання з Wi-Fi підтримувалося, веб-сторінки відкривались без помилок, а дані безпечно передавались і зберігались. Додатково перевірялось поведінку системи при втраті мережі. Пристрій повідомляв про розрив з'єднання та намагався перепідключитися.

ВИСНОВКИ

При виконанні кваліфікаційної роботи на тему «Програмно-апаратний модуль обробки даних із пристроїв Інтернету речей» було досягнуто наступних результатів:

1. Проведено аналіз сучасних технологій Інтернету речей (IoT), класифікацію їх архітектур, протоколів передачі даних та типових сценаріїв застосування в таких галузях, як охорона здоров'я, транспорт, енергетика, промисловість, сільське господарство, розумні будинки і міста.
2. Проаналізовано сучасні IoT-системи моніторингу та контролю для розумних міст, розглянуто приклади використання вуличного освітлення, смарт-паркування, управління відходами та моніторингу якості повітря з детальним описом їх компонентів та переваг.
3. Сформульовано постановку задачі, яка полягає у створенні ефективного та безпечного IoT-модуля на базі ESP32 з підтримкою локального збереження на microSD, віддаленої візуалізації даних та підтримкою комунікацій за допомогою MQTT-брокера.
4. Розроблено архітектуру програмно-апаратного модуля, яка включає сенсорний блок на базі MPU6050, систему локального збереження (microSD), OLED-дисплей для виводу повідомлень, веб-інтерфейс користувача та модуль бездротової передачі через Wi-Fi.
5. Реалізовано алгоритми опитування сенсора, збереження даних у CSV, обробку конфігурацій, формування MQTT-повідомлень, логіку керування індикацією на дисплеї та взаємодію з веб-сервером.
6. Розроблено інформаційне забезпечення, включаючи структури даних, формат журналів, систему параметризації конфігурацій через JSON, та механізми логування подій і помилок.
7. Проведено вибір апаратних і програмних засобів для реалізації запропонованого рішення з обґрунтуванням вибору.

8. Реалізовано ключові модулі системи, що забезпечують збирання, обробку, передачу та збереження даних, та реалізовано паралельну роботу задач за допомогою FreeRTOS.

9. Розроблено адаптивний веб-інтерфейс, який дозволяє переглядати зібрані дані, контролювати стан пристрою та вносити конфігураційні зміни через браузер.

10. Проведено тестування системи на стабільність роботи, відповідність функціональним вимогам, перевірку коректності збирання та передачі даних, а також симуляцію типових помилок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Y. Yang, M. Zhong, H. Yao, F. Yu, X. Fu and O. Postolach, “Internet of things for smart ports: Technologies and challenges,” *IEEE Instrum. Meas. Mag.*, vol. 21, no. 1, Feb. 2018
2. R. Ferrero, E. Beattie and J. Phoenixes, “Sensor city- a global innovation hub for sensor technology,” *IEEE Instrum. Meas. Mag.*, vol. 21, no. 1, Feb. 2018.
3. M. Hermann, T. Pentek, and B. Otto, “Design principles for Industrie 4.0 scenarios,” in Proc. 2016 49th Hawaii Int. Conf. System Sciences (HICSS), Jan. 2016.
4. Q. F. Hassan, *Internet of Things A to Z: Technologies and Applications*. Hoboken, NJ, USA: John Wiley and Sons, May 2018.
5. D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “Internet of things: vision, applications and research challenges,” *Ad Hoc Networks*, vol. 10, no. 7, 2012.
6. Borgia, E. The Internet of Things vision: Key features, applications and open issues. *Comput. Commun.* 2014, 54, 1–31.
7. Shinde, S.R.; Karode, A.; Suralkar, S. Review on-IoT based environment monitoring system. *Int. J. Electron. Commun. Eng. Technol.* 2017, 8, 103–108
8. Agarwal, V.; Sharma, S. IoT based smart transport management system. In *Proceedings of the International Conference on Advanced Informatics for Computing Research*, Gurugram, India, 26–27 December 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 207–216.
9. Azzawi, M.A.; Hassan, R.; Bakar, K.A.A. A review on Internet of Things (IoT) in healthcare. *Int. J. Appl. Eng. Res.* 2016, 11, 10216–10221.
10. Chen, W.; Jeong, S.; Jung, H. WiFi-Based home IoT communication system. *J. Inf. Commun. Converg. Eng.* 2020, 18, 8–15.
11. Kalanandhini, G.; Aravind, A.; Vijayalakshmi, G.; Gayathri, J.; Senthilkumar, K. Bluetooth technology on IoT using the architecture of Piconet and Scatternet. *AIP Conf. Proc.* 2022, 2393, 020121.

12. Tang, P.; Liang, Q.; Li, H.; Pang, Y. Application of Internet-of-Things Wireless Communication Technology in Agricultural Irrigation Management: A Review. *Sustainability* 2024, 16, 3575.
13. Jaloudi, S. microIoTSCADA: A Tool to Monitor and Control Renewable Energy-Based Systems Using MODBUS. In *Proceedings of the 2024 4th Interdisciplinary Conference on Electrics and Computer (INTCEC)*, Chicago, IL, USA, 11–13 June 2024; pp. 1–6.
14. Khan, M.H.; Javed, A.R.; Iqbal, Z.; Asim, M.; Awad, A.I. DivaCAN: Detecting in-vehicle intrusion attacks on a controller area network using ensemble learning. *Comput. Secur.* 2024, 139, 103712.
15. Santhosh, C.; Kumer, S.A.; Krishna, J.G.; Vaishnavi, M.; Sairam, P.; Kasulu, P. IoT based smart energy meter using GSM. *Mater. Today Proc.* 2021, 46, 4122–4124.
16. Shilpa, B.; Gupta, H.P.; Jha, R.K.; Hashmi, S.S. LoRa interference issues and solution approaches in dense IoT networks: A review. *Telecommun. Syst.* 2024, 87, 517–539.
17. Ammar, M.; Russello, G.; Crispo, B. Internet of Things: A survey on the security of IoT frameworks. *J. Inf. Secur. Appl.* 2018, 38, 8–27.
18. Harmon, R.R.; Castro-Leon, E.G.; Bhide, S. Smart cities and the Internet of Things. In *Proceedings of the 2015 Portland International Conference on Management of Engineering and Technology (PICMET)*, Portland, OR, USA, 2–6 August 2015; pp. 485–494.
19. Abir, S.M.A.A.; Anwar, A.; Choi, J.; Kayes, A.S.M. IoT-Enabled Smart Energy Grid: Applications and Challenges. *IEEE Access* 2021, 9, 50961–50981.
20. Khan, N.; Ray, R.L.; Sargani, G.R.; Ihtisham, M.; Khayyam, M.; Ismail, S. Current Progress and Future Prospects of Agriculture Technology: Gateway to Sustainable Agriculture. *Sustainability* 2021, 13, 4883.
21. Shinde, S.R.; Karode, A.; Suralkar, S. Review on-IoT based environment monitoring system. *Int. J. Electron. Commun. Eng. Technol.* 2017, 8, 103–108.

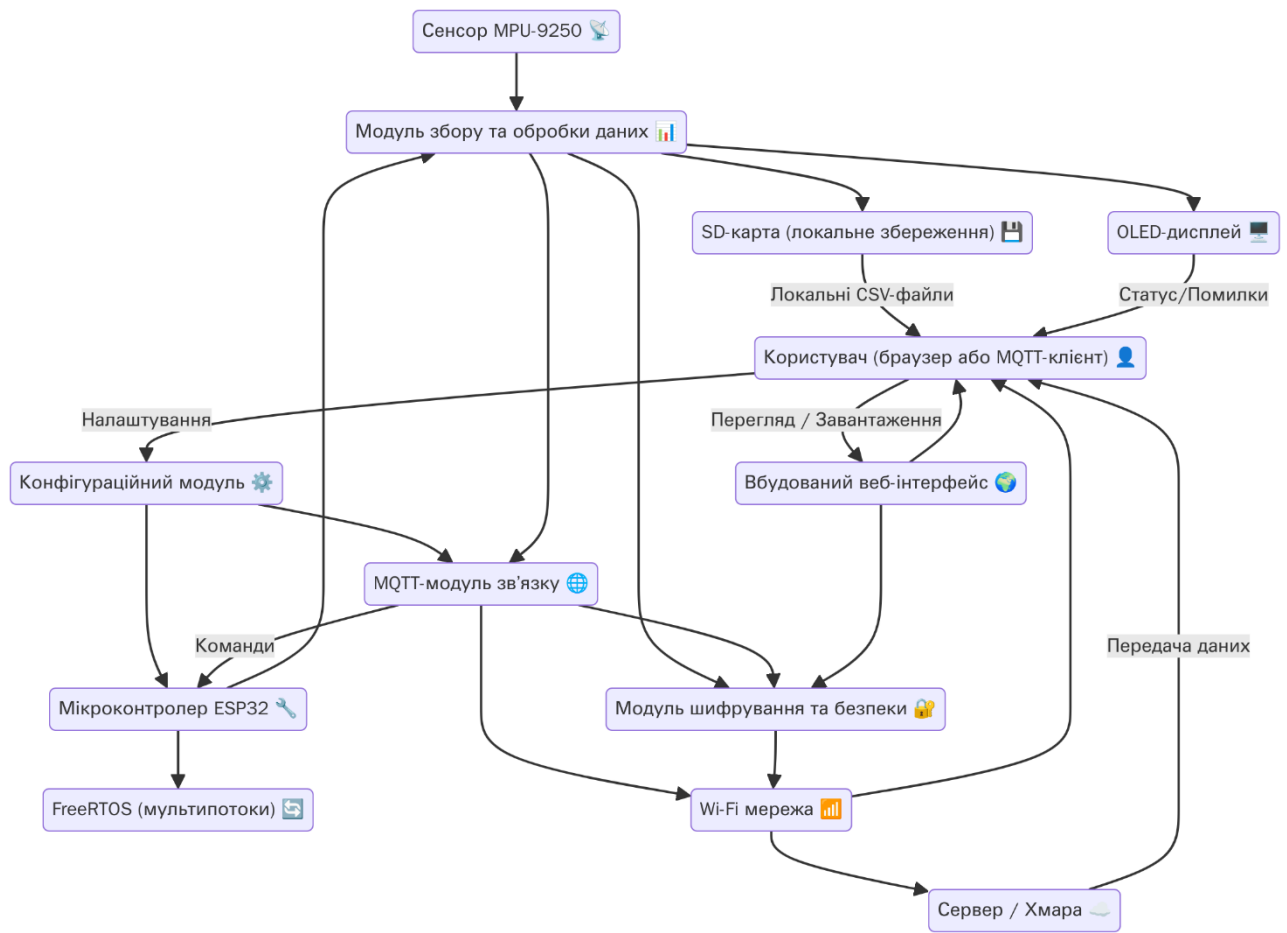
22. Chakraborty, A.; Islam, M.; Shahriyar, F.; Islam, S.; Zaman, H.U.; Hasan, M. Smart home system: A comprehensive review. *J. Electr. Comput. Eng.* 2023, 2023, 7616683.
23. Yassein, M.B.; Hmeidi, I.; Al-Harbi, M.; Mrayan, L.; Mardini, W.; Khamayseh, Y. IoT-based healthcare systems: A survey. In *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems*, Dubai, United Arab Emirates, 2–5 December 2019; pp. 1–9.
24. Tokognon, C.A.; Gao, B.; Tian, G.Y.; Yan, Y. Structural health monitoring framework based on Internet of Things: A survey. *IEEE Internet Things J.* 2017, 4, 619–635.
25. Bagdadee, A.H.; Zhang, L.; Saddam Hossain Remus, M. A brief review of the IoT-based energy management system in the smart industry. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*; Springer: Singapore, 2020; pp. 443–459.
26. Al-Turjman, F.; Altrjman, C.; Din, S.; Paul, A. Energy monitoring in IoT-based ad hoc networks: An overview. *Comput. Electr. Eng.* 2019, 76, 133–142.
27. Zulkifli, C.Z.; Garfan, S.; Talal, M.; Alamoodi, A.H.; Alamleh, A.; Ahmaro, I.Y.Y.; Sulaiman, S.; Ibrahim, A.B.; Zaidan, B.B.; Ismail, A.R.; et al. IoT-Based Water Monitoring Systems: A Systematic Review. *Water* 2022, 14, 3621.
28. Jan, F.; Min-Allah, N.; Düstegör, D. Iot based smart water quality monitoring: Recent techniques, trends and challenges for domestic applications. *Water* 2021, 13, 1729.
29. Pal, R.; Chavhan, S.; Gupta, D.; Khanna, A.; Padmanaban, S.; Khan, B.; Rodrigues, J.J. A comprehensive review on IoT-based infrastructure for smart grid applications. *IET Renew. Power Gener.* 2021, 15, 3761–3776.
30. Pal, R.; Chavhan, S.; Gupta, D.; Khanna, A.; Padmanaban, S.; Khan, B.; Rodrigues, J.J. A comprehensive review on IoT-based infrastructure for smart grid applications. *IET Renew. Power Gener.* 2021, 15, 3761–3776.
31. Pandey, A.K.; Mukherjee, A. Review on advances in IoT-based technologies for smart agricultural system. *Internet Things Anal. Agric.* 2022, 3, 29–44.

32. Kim, W.S.; Lee, W.S.; Kim, Y.J. A review of the applications of the internet of things (IoT) for agricultural automation. *J. Biosyst. Eng.* 2020, 45, 385–400.
33. Hirlekar, V.; Doshi, P.; Shetty, V.; Tungare, A. Survey on IoT-based bridge health monitoring systems. *Int. J. Res. Eng. Sci. Manag.* 2020, 3, 87–91.
34. Saini, J.; Dutta, M.; Marques, G. Indoor Air Quality Monitoring Systems Based on Internet of Things: A Systematic Review. *Int. J. Environ. Res. Public Health* 2020, 17, 4942.
35. Lavanya, N.; Panchami, S. A review on Iot based traffic management system. *Int. J. Eng. Appl. Sci. Technol.* 2020, 5, 612–615.
36. Syed, A.S.; Sierra-Sosa, D.; Kumar, A.; Elmaghraby, A. IoT in smart cities: A survey of technologies, practices and challenges. *Smart Cities* 2021, 4, 429–475.
37. Shah, A.; Fauzi, S.; Gining, R.; Razak, T.; Jamaluddin, M.; Maskat, R. A review of IoT-based smart waste level monitoring system for smart cities. *Indones. J. Electr. Eng. Comput. Sci.* 2021, 21, 450–456.
38. Laha, S.R.; Pattanayak, B.K.; Pattnaik, S. Advancement of Environmental Monitoring System Using IoT and Sensor: A Comprehensive Analysis. *AIMS Environ. Sci.* 2022, 9, 771–800.
39. Arjun, P.; Stephenraj, S.; Kumar, N.N.; Kumar, K.N. A Study on IoT based smart street light systems. In *Proceedings of the 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, Pondicherry, India, 29–30 March 2019; pp. 1–7.
40. Elkateb, S.; Métwalli, A.; Shendy, A.; Abu-Elanien, A.E. Machine learning and IoT—Based predictive maintenance approach for industrial applications. *Alex. Eng. J.* 2024, 88, 298–309.
41. Mourtzis, D.; Angelopoulos, J.; Panopoulos, N. Design and development of an IoT enabled platform for remote monitoring and predictive maintenance of industrial equipment. *Procedia Manuf.* 2021, 54, 166–171.
42. Sudha Rani, G.; Naveen Kumar, G.; Anusha, I.; Venkata Jyothi, M.; Niteesh Kumar, P. IoT Based Street Light Controller and Monitoring system. *Int. J. Mod. Trends Sci. Technol.* 2024, 10, 274–281.

43. Bhuvaneswari, T.; Hossen, J.; Asyiqinbt, N.; Hamzah, A.; Velraj Kumar, P.; Jack, O.H. Internet of things (IoT) based smart garbage monitoring system. *Indones. J. Electr. Eng. Comput. Sci.* 2020, 20, 736–743.
44. Cynthia, J.; Priya, C.B.; Gopinath, P. IOT based smart parking management system. *Int. J. Recent Technol. Eng. (IJRTE)* 2018, 7, 374–379.
45. Парубок Є., Осолінський О. Програмно-апаратний модуль обробки даних із пристроїв інтернету речей. Інтелектуальні інформаційні технології в прикладних дослідженнях : тези доп. студентської наук.-практ. конф. (м. Тернопіль, 27–29 травня 2025 р.). Тернопіль, 2025. С. 309-311.
46. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.
47. Островерхов В. М., Біловус Л. І., Восьний К. З., Луцишин О. О., Монастирський Г. Л., Надвиничний С. А., Питель С. В., Шандрук С. К. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

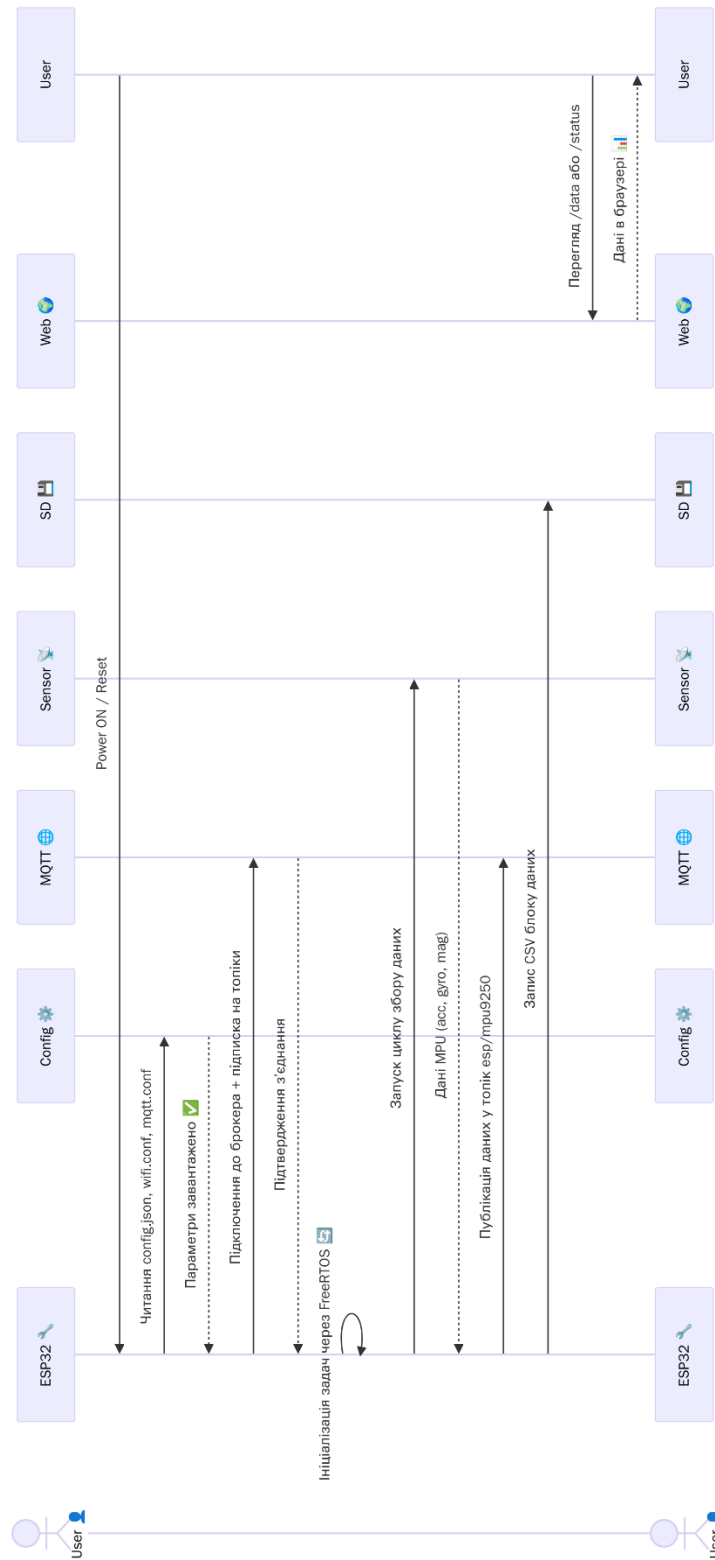
Додаток А

Розширена архітектура системи



Додаток Б

Діаграма послідовності програмно – апаратного модуля



Додаток В

Код програмно апаратного модуля

```
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <WebServer.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_SSD1306.h>

// === Wi-Fi + MQTT ===
const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";
const char* mqtt_server = "broker.hivemq.com";
WiFiClient espClient;
PubSubClient client(espClient);
WebServer server(80);

// === Ініціалізація об'єктів ===
Adafruit_MPU6050 mpu;
Adafruit_SSD1306 display(128, 64, &Wire, -1);
const int chipSelect = 5;

float accX, accY, accZ;
float gyroX, gyroY, gyroZ;

// === MPU6050 ===
void setupMPU() {
  if (!mpu.begin()) {
    Serial.println("Помилка: MPU6050 не знайдено");
    while (1) delay(10);
  }
  Serial.println("MPU6050 знайдено.");
  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
  mpu.setGyroRange(MPU6050_RANGE_500_DEG);
  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
}
```

```

void loopMPU() {
    sensors_event_t acc, gyro, temp;
    mpu.getEvent(&acc, &gyro, &temp);

    accX = acc.acceleration.x;
    accY = acc.acceleration.y;
    accZ = acc.acceleration.z;
    gyroX = gyro.gyro.x;
    gyroY = gyro.gyro.y;

    gyroZ = gyro.gyro.z;

    Serial.printf("Accel: %.2f %.2f %.2f\n", accX, accY, accZ);
    Serial.printf("Gyro: %.2f %.2f %.2f\n", gyroX, gyroY, gyroZ);
    delay(500);
}

// === OLED ===
void setupOLED() {
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        Serial.println("OLED не знайдено");
        while (true);
    }
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.println("OLED Ready");
    display.display();
}

void loopOLED() {
    display.clearDisplay();
    display.setCursor(0, 0);
    display.printf("AccX: %.2f\nAccY: %.2f\nAccZ: %.2f\nGyroX: %.2f\nGyroY: %.2f\nGyroZ: %.2f\n", accX, accY, accZ, gyroX, gyroY, gyroZ);
    display.display();
}

// === SD-карта ===
void setupSD() {
    if (!SD.begin(chipSelect)) {
        Serial.println("Помилка ініціалізації SD");
        return;
    }
    Serial.println("SD карта готова.");
}

```

```

void loopSD() {
  File dataFile = SD.open("/log.csv", FILE_APPEND);
  if (dataFile) {
    dataFile.printf("%.2f,%.2f,%.2f,%.2f,%.2f,%.2f\n", accX, accY, accZ,
gyroX, gyroY, gyroZ);
    dataFile.close();
  }
}

// === Wi-Fi та MQTT ===
void setupWiFi() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
  }
}

```

```

        Serial.println("Підключення до WiFi...");
    }
    Serial.println("WiFi з'єднано");
}

void reconnectMQTT() {
    while (!client.connected()) {
        if (client.connect("ESP32Client")) {
            Serial.println("MQTT з'єднано");
        } else {
            delay(2000);
        }
    }
}

void publishMQTT() {
    char msg[128];
    snprintf(msg, sizeof(msg),
        "{\\"accX\\":%.2f,\\"accY\\":%.2f,\\"accZ\\":%.2f,\\"gyroX\\":%.2f,\\"gyroY\\":%.2f,\\"gyroZ\\":%.2f}", accX, accY, accZ, gyroX, gyroY, gyroZ);
    client.publish("esp/mpu6050", msg);
}

// === Веб-сервер ===
void setupWeb() {
    server.on("/status", []() {
        String html = "<html><body><h1>ESP32 Status</h1><p>Accel: " +
            String(accX) + ", " + String(accY) + ", " + String(accZ) + "</p>";
        html += "<p>Gyro: " + String(gyroX) + ", " + String(gyroY) + ", " +
            String(gyroZ) + "</p></body></html>";
        server.send(200, "text/html", html);
    });
    server.begin();
    Serial.println("Веб-сервер запущено.");
}

// setup() + loop() ===
void setup() {
    Serial.begin(115200);
    Wire.begin();
    setupMPU();
    setupOLED();
    setupSD();
    setupWiFi();
    client.setServer(mqtt_server, 1883);
    setupWeb();
}

void loop() {
    loopMPU();
    loopOLED();
    loopSD();

    if (!client.connected()) reconnectMQTT();
    client.loop();
    publishMQTT();
    server.handleClient();
}

```

Додаток Г

Апробація отриманих результатів

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління



ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

Студентської науково-практичної конференції
ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРИКЛАДНИХ
ДОСЛІДЖЕННЯХ
(ПІТАР-2025)

27-29 травня 2025 року

Тернопіль
2025

Збірник тез доповідей студентської науково-практичної конференції «Інтелектуальні інформаційні технології в прикладних дослідженнях» (ІТІТАР-2025). (Тернопіль, 27-29 травня 2025 року). Тернопіль: ЗУНУ, 2025. 372 с.

До збірника увійшли тези доповідей учасників студентської науково-практичної конференції «Інтелектуальні інформаційні технології в прикладних дослідженнях» (ІТІТАР – 2025), що відбувалась у рамках AI Week на базі кафедри інформаційно-обчислювальних систем і управління Західноукраїнського національного університету. Мета конференції — об'єднати науковців, освітян, представників IT-бізнесу та здобувачів освіти для обміну досвідом, презентації сучасних досліджень і впровадження інноваційних рішень у сфері комп'ютерних наук, штучного інтелекту та суміжних галузей.

За зміст наукових праць та достовірність наведених фактологічних і статистичних матеріалів відповідальність несуть автори публікацій та їхні наукові керівники. У збірнику зберігається стилістика та орфографія авторів матеріалів.

Склад організаційного комітету конференції

Керівництво оргкомітету

Мирослав Комар, д.т.н., професор, професор кафедри інформаційно-обчислювальних систем і управління

Христина Ліп'яніна-Гончаренко, д.т.н., доцент, доцент кафедри інформаційно-обчислювальних систем і управління

Надія Васильків, к.т.н., в.о. завідувача кафедри інформаційно-обчислювальних систем і управління

Члени програмного комітету

Василь Коваль, к.т.н., доцент, доцент кафедри інформаційно-обчислювальних систем і управління, заступник декана факультету комп'ютерних інформаційних технологій

Олександр Осолінський, к.т.н., доцент, доцент кафедри інформаційно-обчислювальних систем і управління

Павло Биковий, к.т.н., доцент, доцент кафедри інформаційно-обчислювальних систем і управління

Діана Загородня, к.т.н., доцент, доцент кафедри інформаційно-обчислювальних систем і управління

Ігор Майків, к.т.н., доцент кафедри інформаційно-обчислювальних систем і управління

Члени оргкомітету

Андрій Івасечко, викладач кафедри інформаційно-обчислювальних систем і управління

Дмитро Дюг, викладач кафедри інформаційно-обчислювальних систем і управління

Христина Юрків, студентка ФКІТ ЗУНУ, технік лабораторії з проблем інформаційних технологій

Мар'яна Соя, студентка ФКІТ ЗУНУ, лаборант кафедри інформаційно-обчислювальних систем і управління

Микола Телька, студент ФКІТ ЗУНУ, лаборант кафедри інформаційно-обчислювальних систем і управління, студентський декан факультету комп'ютерних інформаційних технологій

© Кафедра інформаційно-обчислювальних систем і управління ЗУНУ

Микола Гасендич	275
ВБУДОВАНА ІНТЕЛЕКТУАЛЬНА СИСТЕМА МОНІТОРИНГУ ВИРОБНИЦТВА НА БАЗІ ІНТЕРНЕТУ РЕЧЕЙ	275
Гнатів Святослав, Олександр Осолінський	278
ПРОГРАМНИЙ МОДУЛЬ ОБЛІКУ РОБОЧОГО ЧАСУ З ВИКОРИСТАННЯМ ІoT ТЕХНОЛОГІЙ	278
Козак Володимир, Дорош Віталій	281
МОНІТОРИНГ ТРАНСПОРТНИХ ВИКИДІВ НА ОСНОВІ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ ТА ІНТЕРНЕТУ РЕЧЕЙ	281
Кочан Іванна, Кочан Володимир, Кочан Орест	284
ОПРАЦЮВАННЯ РЕЗУЛЬТАТІВ ВИМІРЮВАНЬ ШЛЯХОМ ДИСКРЕТНОГО УСЕРЕДНЕННЯ	284
Крупський Владислав, Осолінський Олександр	287
ПРОГРАМНИЙ МОДУЛЬ ДЛЯ АВТОМАТИЧНОГО СТВОРЕННЯ ІНТЕРФЕЙСІВ КОРИСТУВАЧА З ВИКОРИСТАННЯМ МЕТАДАНИХ ІoT	287
Круглий Олександр, Коваль Василь	291
КОНЦЕПЦІЯ МОДУЛЯ ДЕТЕКТУВАННЯ ДРОНІВ НА ОСНОВІ АКУСТИЧНОЇ ІДЕНТИФІКАЦІЇ	291
Леус Віталій, Осолінський Олександр	295
СИСТЕМА КЕРУВАННЯ ЖЕСТАМИ НА ОСНОВІ КОНТРОЛЕРА LEAP MOTION ТА МІКРОКОНТРОЛЕРА ARDUINO LEONARDO	295
Ліщинський Ігор	299
АЛГОРИТМИ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ РОЗПОДІЛЕНИХ DDOS-АТАК У МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ	299
Макулевич Наталя, Дорош Віталій	302
МОДУЛЬ АНАЛІЗУ ПОЗИ ЛЮДИНИ НА ОСНОВІ MEDIAPIPE	302
Мельник Назар, Осолінський Олександр	306
АРХІТЕКТУРА СИСТЕМИ ДОМАШНЬОЇ АВТОМАТИЗАЦІЇ НА БАЗІ ANDROID З ВИКОРИСТАННЯМ ІНТЕРНЕТУ РЕЧЕЙ	306
Парубок Євген, Осолінський Олександр	309
ПРОГРАМНО-АПАРАТНИЙ МОДУЛЬ ОБРОБКИ ДАНИХ ІЗ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ	309
Ружицький Дмитро, Осолінський Олександр	312
КОНЦЕПЦІЯ ІНТЕРФЕЙСІВ ПРИКЛАДНОГО ПРОГРАМУВАННЯ ДЛЯ ХМАРИ ІНТЕРНЕТУ РЕЧЕЙ	312

Парубок Євген
студент групи КН-43
evgenparubok@gmail.com

Осолінський Олександр
к.т.н., доцент
oso@wunu.edu.ua

Західноукраїнський національний університет
Тернопіль, Україна

ПРОГРАМНО-АПАРАТНИЙ МОДУЛЬ ОБРОБКИ ДАНИХ ІЗ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ

Вступ. Стрімкий розвиток технологій Інтернету речей (IoT) спричинив зростання попиту на ефективні, автономні та масштабовані рішення для збору, обробки та передачі даних у реальному часі. Такі рішення є критично важливими для розумних міст, промислових систем, сільського господарства, а також для побутових пристроїв, що потребують постійного моніторингу. Забезпечення надійного зберігання даних, візуалізації результатів та віддаленого доступу до пристроїв у динамічному цифровому середовищі створює актуальний запит на розробку програмно-апаратних модулів нового покоління.

Архітектура модуля

В даній роботі представлено розробку програмно-апаратного модуля на базі мікроконтролера ESP32, який інтегрує сенсор MPU6050, модуль microSD для локального зберігання, OLED-дисплей та бездротове з'єднання через Wi-Fi з підтримкою MQTT-протоколу.

Запропонована архітектура реалізує багаторівневу обробку даних із використанням FreeRTOS, що забезпечує паралельну обробку задач, високий рівень стабільності та адаптивності до змінних умов експлуатації.

На рисунку 1 представлено загальну архітектуру системи, що охоплює модулі збору, обробки, передачі, збереження даних та веб-інтерфейс для віддаленого моніторингу.

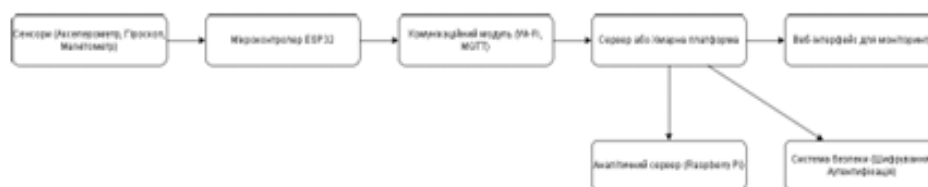


Рисунок 1 – Архітектура програмно-апаратного модуля IoT-системи

Реалізація модуля

Система використовує файл конфігурації для динамічного налаштування параметрів, таких як мережа Wi-Fi та MQTT-з'єднання.

Реалізовано веб-інтерфейс, який забезпечує доступ до поточних параметрів пристрою та завантаження зібраних даних у форматі CSV. У разі втрати зв'язку передбачено дублювання інформації на SD-картку.

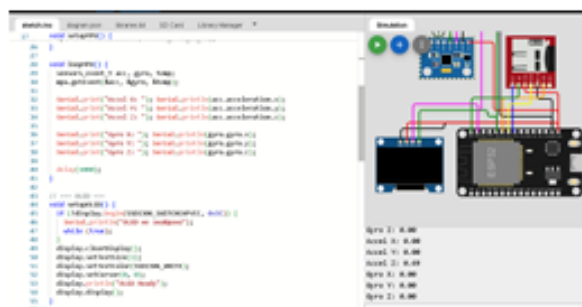


Рисунок 2— Зібраний блок апаратної частини модуля

Алгоритми обробки даних враховують частоту вибірки та зберігають точність показників.

Результати тестування продемонстрували стабільність функціонування пристрою, швидкий відгук на запити користувача та адаптацію до зміни налаштувань у режимі реального часу.

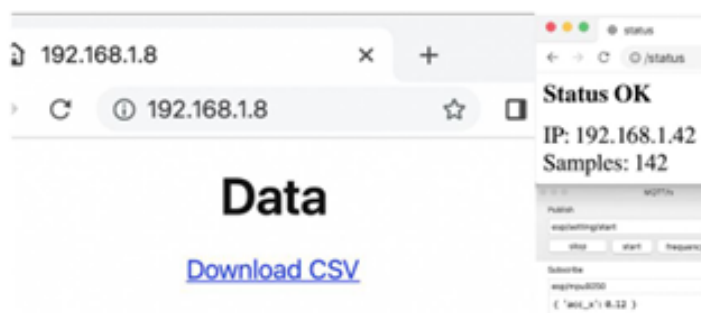


Рисунок 3— Тест роботи генерації веб-сторінок

Висновок

Було розроблено повнофункціональний програмно-апаратний модуль для обробки даних з пристроїв Інтернету речей. Рішення побудовано на базі мікроконтролера ESP32 з підтримкою сенсорів, локального збереження даних, бездротової передачі інформації та веб-інтерфейсу для взаємодії з користувачем.

Модульна архітектура дозволяє легко масштабуватись і легко адаптувати для інших сенсорів або задач.

Даний модуль можна використовувати як базу для створення автономних IoT-систем моніторингу в різних галузях.

Список використаних джерел

1. Liu, Z. (2025). Evaluating digitalized visualization interfaces: Integrating visual design elements and analytic hierarchy process. *International Journal of Human-Computer Interaction*, 41(9), 5731-5760.
2. Ruano, I., Estévez, E., Gámez, J., & Gómez, J. (2023). Standards for the integration of online laboratories with learning management systems. *IEEE Access*, 11, 125267-125290.
3. Ala-Laurinaho, R., Autiosalo, J., & Tammi, K. (2020). Open sensor manager for IIoT. *Journal of Sensor and Actuator Networks*, 9(2), 30.
4. Q. F. Hassan, *Internet of Things A to Z: Technologies and Applications*. Hoboken, NJ, USA: John Wiley and Sons, May 2018.
5. D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, 2012.
6. Borgia, E. "The Internet of Things vision: Key features, applications and open issues," *Comput. Commun.*, vol. 54, 2014, pp. 1–31.
7. Elkateb, S., Métwalli, A., Shendy, A., Abu-Elanien, A.E. "Machine learning and IoT—Based predictive maintenance approach for industrial applications," *Alex. Eng. J.*, vol. 88, 2024, pp. 298–309.
8. Mourtzis, D., Angelopoulos, J., Panopoulos, N. "Design and development of an IoT enabled platform for remote monitoring and predictive maintenance of industrial equipment," *Procedia Manuf.*, vol. 54, 2021, pp. 166–171.
9. Chen, W.; Jeong, S.; Jung, H. "WiFi-Based home IoT communication system," *J. Inf. Commun. Converg. Eng.*, vol. 18, 2020, pp. 8–15.