

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформаційних систем та технологій**

«На правах рукопису»

УДК \_\_\_\_004.043 \_\_\_\_

До захисту допущено:

В. о. завідувача кафедри

\_\_\_\_\_ Олександр РОЛІК

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**за освітньо-професійною програмою «Інформаційне забезпечення  
робототехнічних систем»**

**зі спеціальності 126 «Інформаційні системи та технології»**

**на тему: «Система автоматизованого збору та обробки даних з датчиків  
робототехнічного модуля в системі розумний будинок»**

Виконав:

Студент VI курсу, групи ІК-01мп

Устинов Олексій Вікторович \_\_\_\_\_

Керівник:

Доцент, к.т.н.

Крилов Євген Володимирович \_\_\_\_\_

Рецензент:

Доцент, к.т.н.

Ліхоузова Тетяна Анатоліївна \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних посилань.  
Студент \_\_\_\_\_

Київ – 2021 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри

\_\_\_\_\_ Олександр РОЛІК

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**  
**Устинову Олексію Вікторовичу**

1. Тема дисертації «Система автоматизованого збору та обробки даних з датчиків робототехнічного модуля в системі розумний будинок», науковий керівник дисертації Крилов Євген, к.т.н., доц., затверджені наказом по університету від «27» 10 2021 р. № 3587-с
2. Термін подання студентом дисертації \_\_\_\_\_
3. Об'єкт дослідження Програмне забезпечення для формування процесу ефективного збору, обробки та швидкої передачі даних користувачу з віддалених датчиків в системі розумний будинок
4. Вихідні дані Технології, методи та засоби для створення апаратної та програмної частин системи розумний будинок
5. Перелік завдань, які потрібно розробити Аналіз та опис предметної області; вибір та опис компонентів й технологій для розробки системи; розроблення програмного забезпечення, маркетинговий аналіз стартап-проєкту
6. Орієнтовний перелік графічного (ілюстративного) матеріалу Архітектура автоматизованої системи; структурна схема взаємодії компонентів системи; алгоритм роботи серверної частини системи; схема доставки повідомлень до користувача; діаграма прецедентів системи; діаграма послідовностей системи; структура бази даних

7. Орієнтовний перелік публікацій: \_\_\_\_\_

8. Дата видачі завдання 01.09.2020

Календарний план

| № з/п | Назва етапів виконання магістерської дисертації             | Термін виконання етапів магістерської дисертації | Примітка |
|-------|-------------------------------------------------------------|--------------------------------------------------|----------|
| 1     | Аналіз предметної області та постановка завдання            | 03.09.2021р.                                     |          |
| 2     | Вибір та опис компонентів й технологій для розробки системи | 14.09.2021р.                                     |          |
| 3     | Проектування апаратного модуля                              | 23.09.2021р.                                     |          |
| 4     | Опис архітектури системи                                    | 06.10.2021р.                                     |          |
| 5     | Розроблення програмної та апаратної частин                  | 25.10.2021р.                                     |          |
| 6     | Маркетинговий аналіз стартап-проекту                        | 02.11.2021р.                                     |          |
| 7     | Оформлення пояснювальної записки                            | 19.11.2021р.                                     |          |
| 8     | Висновки                                                    | 24.11.2021р.                                     |          |

Студент

Олексій УСТИНОВ

Науковий керівник

Євген КРИЛОВ

## АНОТАЦІЯ

У даній роботі розглянуто спосіб ефективного збору даних з датчиків, обробки та швидкої передачі даних користувачу в системі розумний будинок. Предметом даного дослідження були методи, технології та засоби для побудови апаратної та програмної частин з яких складається система розумний будинок.

У даній роботі проведено аналіз по вибору компонентної бази та програмного забезпечення системи. Апаратна частина системи була розроблена на основі мікроконтролера ESP32. До апаратної частини також входять датчики освітленості, температури й вологості, датчик газів та датчик руху. Дана система має можливість віддалено інформувати користувача про стан приміщення, а також віддаленого налаштування системи. Унікальною ознакою автоматизованої системи є функціонал, що надає можливість користувачу віддалено конфігурувати кредити підключення до Wi-Fi мережі, якою потім і буде виконано відправлення даних користувачу, а також можливість зберігати конфіденційні дані до флешпам'яті пристрою. Далі усі зібрані дані система надсилає на віддалений вебсервер Telegram, де згодом користувач зможе дізнаватись про стан приміщення шляхом відправки метрик ботом в телеграм канал.

Ключові слова: мікроконтролер, сервер, телеграм, користувач, автоматизована система, флеш-пам'ять, база даних, датчики, IoT, SPIFFS, Telegram Bot API, ESP32.

Розмір пояснювальної записки – 99 аркушів, містить 60 ілюстрацій, 8 додатків.

## ABSTRACT

This paper we considers a method of efficient data collection from sensors, processing and fast data transmission to the user in a smart home system. The subject of this study were methods, technologies and tools for building hardware and software parts of which consists of a smart home system.

In this paper, an analysis of the choice of component base and system software was performed. The hardware of the system was developed on the basis of the ESP32 microcontroller. The hardware also includes light, temperature and humidity sensors, a gas sensor and a motion sensor. This system has the ability to remotely inform the user about the status of the room, as well as remote system settings. A unique feature of the automated system is the functionality that allows the user to remotely configure credits to connect to a Wi-Fi network, which will then send data to the user, as well as the ability to store confidential data to the device's flash memory. Then the system sends all the collected data to the remote web server Telegram, where the user can later find out about the state of the room by sending metrics by a bot to the telegram channel.

Keywords: microcontroller, server, telegram, user, automated system, flash memory, database, sensors, IoT, SPIFFS, Telegram Bot API, ESP32.

The size of the explanatory note is 99 sheets, contains 60 illustrations, 8 appendices.

# ЗМІСТ

|                                                                                                                 |    |
|-----------------------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....                                                                                  | 8  |
| Вступ.....                                                                                                      | 9  |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ СИСТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ.....                                                   | 10 |
| 1.1 Об'єкт та предмет дослідження .....                                                                         | 10 |
| 1.2 Огляд існуючих функціональних можливостей збору даних з віддалених датчиків в системі розумний будинок..... | 10 |
| 1.3 Аналіз існуючих систем збору даних з віддалених датчиків .....                                              | 14 |
| 1.4 Постановка задачі.....                                                                                      | 17 |
| Висновок до розділу.....                                                                                        | 18 |
| 2 ВИБІР АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ .....                                                                         | 20 |
| 2.1 Аналіз компонентів апаратної частини.....                                                                   | 20 |
| 2.1.1 ESP32 .....                                                                                               | 20 |
| 2.1.2 BH1750.....                                                                                               | 24 |
| 2.1.3 AM-312 .....                                                                                              | 26 |
| 2.1.4 DHT-11.....                                                                                               | 27 |
| 2.1.5 MQ-2 .....                                                                                                | 28 |
| 2.2 Протоколи зв'язку між компонентами.....                                                                     | 31 |
| Висновок до розділу.....                                                                                        | 36 |
| 3 ВИБІР ТЕХНОЛОГІЙ ДЛЯ ПРОЦЕСУ АВТОМАТИЗАЦІЇ ЗБОРУ ДАНИХ З ВІДДАЛЕНИХ ДАТЧИКІВ В СИСТЕМІ РОЗУМНИЙ БУДИНОК ..... | 38 |
| 3.1 C++.....                                                                                                    | 38 |
| 3.2 Telegram Bot API .....                                                                                      | 40 |
| 3.3 HTML, CSS та Javascript.....                                                                                | 42 |
| 3.4 Вибір бази даних .....                                                                                      | 46 |
| Висновок до розділу.....                                                                                        | 49 |
| 4 РОЗРОБЛЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ РОЗУМНИЙ БУДИНОК ІЗ КЕРУВАННЯМ ЧЕРЕЗ ТЕЛЕГРАМ БОТА.....                   | 51 |
| 4.1 Архітектура системи .....                                                                                   | 51 |

|                                                              |     |
|--------------------------------------------------------------|-----|
| 4.2 Загальна схема взаємодії компонентів системи.....        | 52  |
| 4.3 Розроблення серверної частини системи.....               | 53  |
| 4.3 Розроблення клієнтської частини системи .....            | 69  |
| Висновок до розділу.....                                     | 73  |
| 5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ .....                          | 75  |
| 5.1 Опис ідеї проєкту .....                                  | 75  |
| 5.2 Технічний аудит проєкту.....                             | 77  |
| 5.3 Аналіз ринкових можливостей запуску стартап-проєкту..... | 78  |
| 5.4 Розроблення ринкової стратегії стартап-проєкту .....     | 87  |
| 5.5 Розроблення маркетингової програми стартап-проєкту.....  | 91  |
| Висновок до розділу.....                                     | 95  |
| ВИСНОВКИ.....                                                | 96  |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                            | 99  |
| ДОДАТОК А.....                                               | 100 |
| ДОДАТОК Б.....                                               | 101 |
| ДОДАТОК В .....                                              | 102 |
| ДОДАТОК Г.....                                               | 103 |
| ДОДАТОК Д.....                                               | 104 |
| ДОДАТОК Е .....                                              | 105 |
| ДОДАТОК Ж .....                                              | 106 |
| ДОДАТОК И.....                                               | 107 |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IoT – internet of thing

SPI – serial peripheral interface

SDIO – secure digital input output

SRAM – static random access memory

I2C – inter-integrated circuit

JSON – javascript object notation

SSL – secure sockets layer

RNG – random number generation

CSS – cascading style sheets

AJAX – asynchronous javascript and xml

SQL – structured query language

IP – internet protocol

IDE – integrated development environment

API – application programming interface

URL – uniform resource locator

WS – web socket

## ВСТУП

Нині, однією з головних частин побутового життя є датчики. Їх використовують як в системі розумного будинку в якості сигналізації й датчиків руху, так навіть в медичному обладнанні.

Зокрема, системи автоматизації вже отримали широке визнання у кількох галузях промисловості, відіграючи життєво важливу роль у домінуванні багатьох операцій, пов'язаних із процесами. Ми живемо у світі автоматизації, де більшість систем стали машинними, наприклад, промислова автоматизація, автоматизація власних будинків та альтернативні сфери бізнесу. Системи домашньої автоматизації просуваються до процесів механізації, завдяки чому машинне обладнання вимагає менших зусиль для управління численними системами в будинках. Такі системи передбачають автоматичне керування побутовою технікою з використанням абсолютно різних технологій та контролерів за настільними комп'ютерами, ноутбуками, смартфонами або планшетами. Системи домашньої автоматизації також класифікуються на три типи: домашня автоматизація на основі лінії електропередачі або кабельної шини та бездротова автоматизація будинку.

Промислові системи використовують датчики для моніторингу як самих промислових процесів якості продукції, так і стану обладнання. Розумні будівлі використовують датчики для визначення місцезнаходження людей та навіть стану будівлі. Ці дані можна використовувати для управління системами опалення, вентиляції, кондиціонування повітря та системами освітлення для зменшення експлуатаційних витрат, а також для захисту будівлі від небажаного втручання.

Метою даної роботи є розроблення пристрою збору даних, який буде підходити для багатьох задач, легко масштабуватися та розширюватися, мати можливість перепрограмування, володіти можливістю віддаленої передачі інформації за допомогою мережі Wi-Fi.

Тому в даній дисертації буде представлено проєкт Інтернету речей, який має мету розв'язати проблему зі збором інформації з датчиків робототехнічного модуля.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ СИСТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Об'єкт та предмет дослідження

Процес автоматизації дозволяє виключити людський фактор. Часто трапляється так, що людина за якихось причин забуває про увімкнену газову плиту або воду. Це не рідко приводить до поганих наслідків. Автоматизація не дає розвитку таким ситуаціям і вчасно виявляє дані дії. Сучасні системи розробляються на основі стандартів, що дозволяють забезпечити як високу ефективність, так й можливість їх взаємодії між собою.

Дана робота присвячена дослідженню, що проводиться в області збору даних з віддалених датчиків та передачі їх на віддалений сервер, де користувач отримуватиме в месенджері Telegram сповіщення про стан системи, а також матиме можливість керувати датчиками шляхом посилення команди на сервер.

Об'єктом дослідження є процес ефективного збору, обробки та швидкої передачі даних користувачу з віддалених датчиків в системі розумний будинок.

Предметом дослідження є методи, технології та засоби для побудови апаратної та програмної частин з яких складається система розумний будинок.

## 1.2 Огляд існуючих функціональних можливостей збору даних з віддалених датчиків в системі розумний будинок

В основі кожної системи IoT визначають підключені пристрої, які відповідають за забезпечення сутності Інтернету речей, особливо даних. Щоб отримати фізичні параметри з зовнішнього світу або з об'єкта необхідні датчики. Вони бувають або вбудованими в пристрої, або реалізованими як окремі об'єкти для вимірювання чи збору даних телеметрії. Ще одним обов'язковим елементом є виконавчі механізми. Будучи в тісній зв'язці з датчиками, вони здатні перетворювати дані, які генеруються інтелектуальними об'єктами, у фізичні дії. Це важливо, адже підключені об'єкти повинні не лише тримати зв'язок із необхідними шлюзами чи системами збору даних, а також вміти розпізнавати і спілкуватися один з одним для збору та обміну

інформацією, спільної роботи в режимі реального часу. Далі системи на основі концепції Інтернету речей можна вважати збір даних IoT і використання шлюзів, які отримують величезну масу необроблених даних, задля подальшого їх перетворення в цифрові потоки з відповідною обробкою та аналізом. Шлюзи полегшують зв'язок між датчиками й іншою частиною системи, перетворюючи дані датчиків у формати, які легко переносяться і можуть використовуватися іншими компонентами системи. Більш того, вони можуть контролювати, фільтрувати і вибирати дані, щоб мінімізувати обсяг інформації, яка повинна бути передана в «хмару», що позитивно впливає на вартість передачі мережею і час відгуку. В результаті шлюзи надають можливість для обробки даних датчиків локально, які стискаються в пакети і є готовими для подальшої обробки.

На сьогодні виділяють 4 головні структури підключення пристроїв IoT в мережі:

- безпосереднього зв'язку між пристроями;
- зв'язку між пристроєм та хмарою;
- зв'язку пристрою зі шлюзом;
- внутрішнього обміну даними.

Першу структуру зображено на рис. 1.1. Вона дозволяє пристроям з одним і тим самим протоколом зв'язку взаємодіяти між собою. Найчастіше використовується в системах домашньої автоматизації для передачі невеликих порцій даних між пристроями IoT та іншими інтелектуальними пристроями з достатньо не великою швидкістю передачі необхідних даних. Такими пристроями зазвичай є лампи, датчик температури, датчик вологості, інші датчики що надсилають інформацію один одному. Цей підхід дозволяє підключати розумні пристрої, які можна носити, наприклад, розумний годинник, де дані не обов'язково мають передаватися у глобальну мережу, а іноді достатньо лише зв'язатися із смартфоном через додаток тощо.

Друга структура, яку зображено на рис. 1.2 дає можливість IoT-пристроєм обмінюватись даними через інтернет мережу та хмарні служби. Перевагами існуючих систем зв'язку є наявність: Ethernet, або Wi-Fi, щоб можна було установити зв'язок

між пристроєм інтернет речей та IP-мережею, яка в кінці з'єднується з хмарним сервером.



Рисунок 1.1 – Безпосередній зв'язок між пристроями

Це підключення допомагає користувачам отримувати доступ на відстані до пристрою IoT. Наприклад, система розумний будинок використовує цей підхід. Тобто клієнт цієї системи, маючи телефон, або персональний комп'ютер, може керувати та слідкувати за станами датчиків.

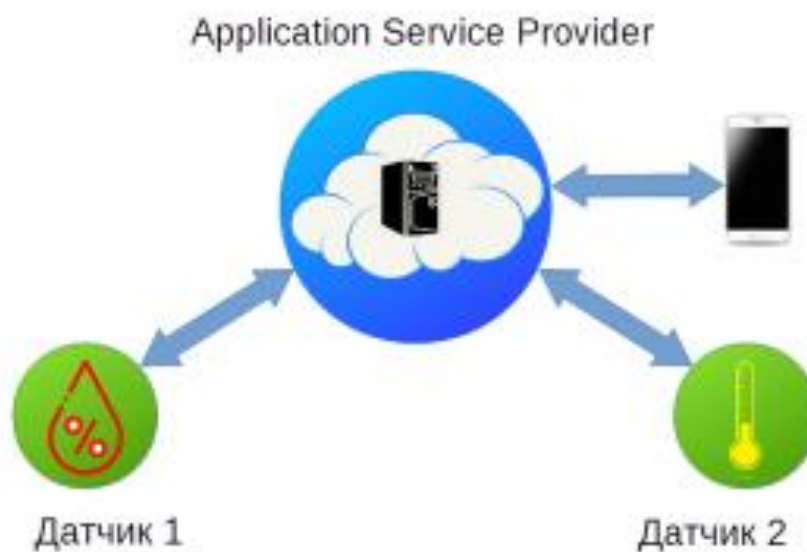


Рисунок 1.2 – Зв'язок між пристроєм IoT та хмарою

Третю структуру, що зображено на рис. 1.3 дозволяє пристроям IoT підключатись до пристрою для доступу до хмарного середовища. На даному пристрої встановлена програма, яка знаходиться між пристроєм та хмарним середовищем й виконує такі функції: безпечний обмін даними між пристроями IoT та хмарним середовищем; авторизацію, узгодження протоколів. В якості шлюзу може бути девайс з додатковим програмним забезпеченням, наприклад, для передачі інформації з розумного годинника, або спеціальне обладнання на базі мікроконтролера для управління сенсорною мережею і забезпечення доступу до хмарного середовища.

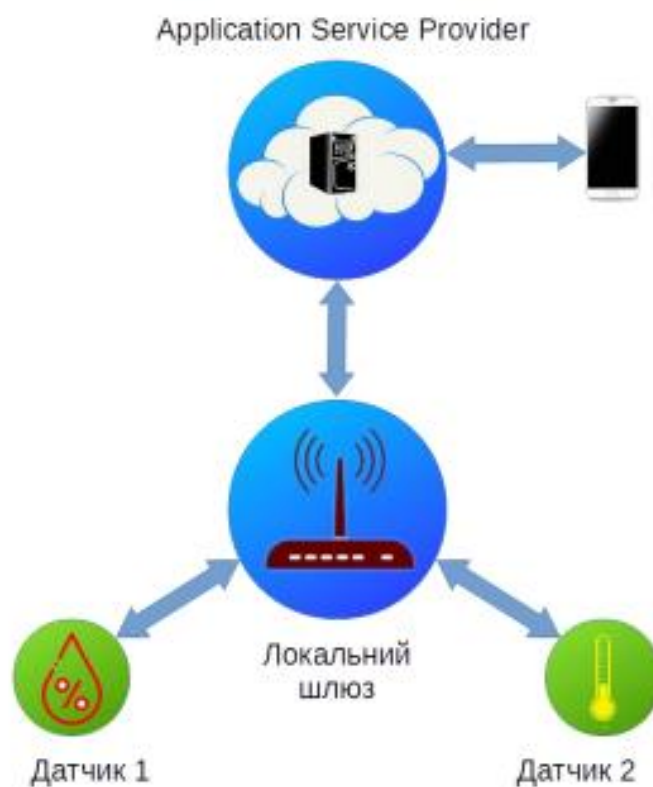


Рисунок 1.3 – Зв'язок між IoT-пристроєм та шлюзом

Четверта структура на рис. 1.4 по суті розширює структуру комунікації між пристроями IoT та хмарними середовищами. Структури під номером два та три забезпечують пристроям IoT загрузку даних тільки до одного хмарного середовища. Однак не рідко необхідна можливість експорту та аналізу даних в поєднанні з даними отриманих зі сторонніх джерел. За приклад такого використання можна взяти систему

збору, обробки та зберігання електроенергії та устаткування приватного будинку, вулиці, міста.

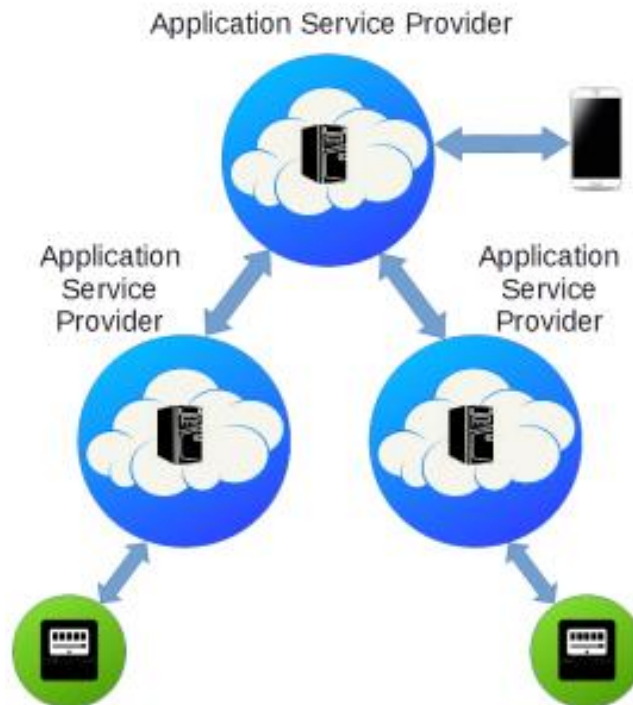


Рисунок 1.4 – Зв'язок між хмарними службами

### 1.3 Аналіз існуючих систем збору даних з віддалених датчиків

На даний момент в світі набули широкого поширення кілька технологій організації домашніх бездротових мереж, що складаються з датчиків і центрального контролера. Серед них: ZigBee, Bluetooth Low Energy. Також досить часто використовується підключення датчиків до домашньої мережі Wi-Fi.

Розумні будинки дозволяють забути про багато технічних моментів побутового життя і зосередитися на інших завданнях, надавши людям вільний час. На ринку представлені готові рішення, але не завжди такі системи підходять для реалізації тих завдань саме які б хотілося бачити. Але, є більш гнучка альтернатива, що дозволяє створити розумний будинок своїми руками на базі контролера Arduino.

Arduino – це платформа для додавання і програмування електронних пристроїв, з типами управління: ручний, напівавтоматичний і автоматичний. Платформа являє

собою конструктор, з прописаними правилами взаємодії елементів між собою. Система є відкритою, тому кожен зацікавлений виробник може вносити власні допрацювання в розвиток Arduino. На рис. 1.5 зображено приклад типового проєкту домашньої автоматизації на базі Ардуіно.

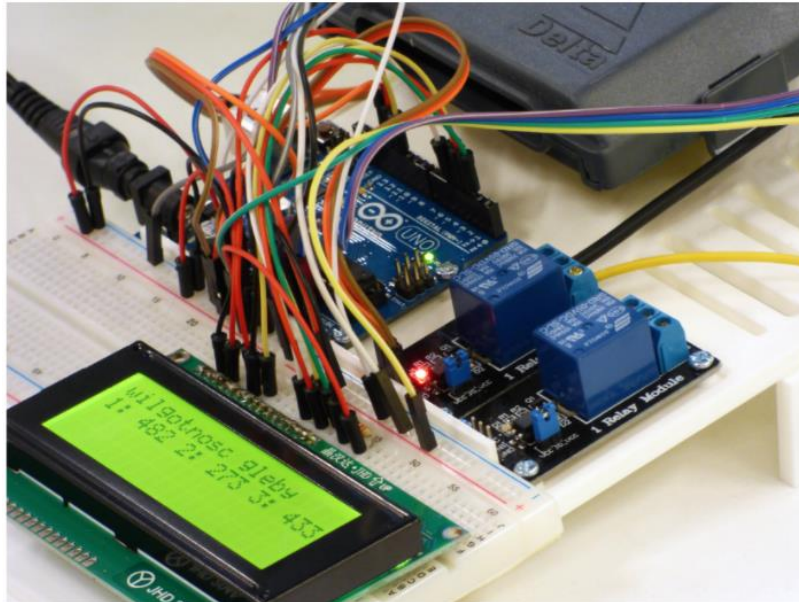


Рисунок 1.5 – Домашня автоматизація на базі Arduino Uno[1]

Функції стандартного розумного будинку:

- збір інформації за допомогою датчиків;
- аналіз даних і прийняття рішення, за допомогою програмованого мікроконтролера;
- реалізація прийнятих рішень за допомогою поданих команд, на різні підключені до системи пристрою.

Конструктор Arduino хороший тим, що в його системі можна використовувати будь-які елементи розумного будинку, від різних виробників. Ця можливість дозволяє платформі же не бути обмеженою лише однієї екосистемою розумного будинку, а підбирати будь-які компоненти електроніки, для реалізації вирішення власних завдань [1].

Переваги та недоліки Arduino Uno (табл. 1.1).

Таблиця 1.1 – Переваги та недоліки Arduino Uno

| Переваги                                                                                                            | Недоліки                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Використання компонентів інших виробників з контролером Arduino                                                     | По зрівнянню з сучасними рішеннями на 2020 рік Arduino IDE – найгірша середа розробки. Навіть коли ви перейдете в іншу середу, IDE знадобиться залишити для прошивки контролера |
| Досить проста мова програмування, є багато документації в мережі                                                    | Невеликий об'єм флешпам'яті                                                                                                                                                     |
| Велика кількість вбудованих бібліотек для роботи з датчиками                                                        | Відсутня можливість роботи Wi-Fi модуля одночасно працювати як точка доступу та як клієнт                                                                                       |
| Посилання команд та сповіщень, програмування, або перенесення готових програмних рішень можна за рахунок USB кабелю | Низька частота процесора                                                                                                                                                        |

Наступна система побудована на базі модуля Raspberry Pi, що зображено на рис. 1.6. Одноплатний обчислювальний модуль Raspberry Pi є чудовою платформою для створення вбудованих комерційних і промислових Linux-додатків. Конструктивно Raspberry Pi версії CM3 + являє собою компактний модуль з чотирьохядерним процесором ARM (BCM2837), з 1 ГБ оперативної пам'яті, eMMC 8/16/32 ГБ і великим числом портів введення та виведення.

Перелік кращих проєктів Raspberry Pi:

- метеостанція;
- відеонагляд;
- портативна бібліотека;
- інтернет-радіо;
- голосовий помічник Amazon Echo;
- ретроконсоль.

Унікальні проєкти на базі Raspberry Pi перетворюють пристрій у що завгодно. Цілком можливо організувати робоче місце на базі Raspberry Pi для роботи з текстом, Web-розробкою, програмуванням та інше.

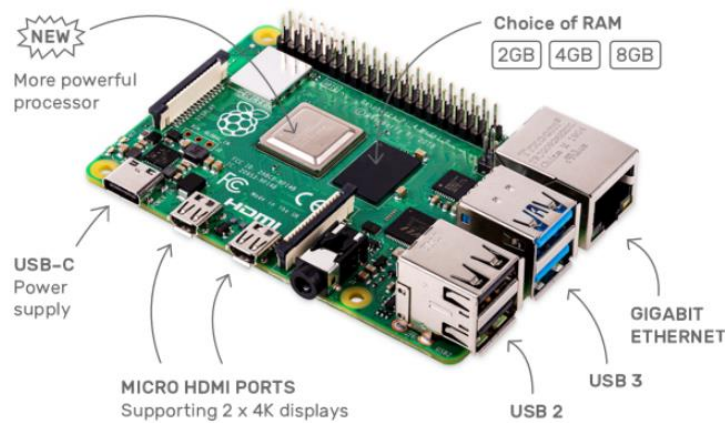


Рисунок 1.6 – Автоматизація на базі Raspberry Pi [2]

Переваги та недоліки використання Raspberry Pi (табл. 1.2).

Таблиця 1.2 – Переваги та недоліки Raspberry Pi

| Переваги                                                    | Недоліки                                                                                      |
|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Доступність(продовжується виробництво та підтримка модулів) | Відсутність Wi-Fi та Bluetooth(необхідні додаткові модулі розширення, такі як SPI, SDIO, USB) |
| Флешпам'ять (8/16/32 ГБ)                                    | Вартість модуля(до 50 доларів)                                                                |
| Температурний діапазон роботи процесора (-25...85C)         | Вартість відладочної плати(до 130 доларів за екземпляр)                                       |
|                                                             | СМЗ + потрібно відразу шість рівнів напруги живлення)                                         |

#### 1.4 Постановка задачі

Метою магістерської дисертації є розроблення автоматизованої системи збору даних з віддалених датчиків в системі розумний будинок. Дана система має

можливість віддалено інформувати користувача про стан приміщення, а також віддаленого налаштування системи. Унікальною ознакою автоматизованої системи є функціонал, що надає можливість користувачу віддалено конфігурувати кредити підключення до Wi-Fi мережі, якою потім і буде виконано відправлення даних користувачу, а також можливість зберігати конфіденційні дані до флешпам'яті пристрою. Далі усі зібрані дані система надсилає на віддалений вебсервер Telegram, де згодом користувач зможе дізнаватись про стан приміщення шляхом відправки метрик ботом в телеграм канал.

Для виконання поставленої мети в даній роботі було визначено наступні завдання:

- проаналізувати існуючі функціональні можливості та аналоги систем збору даних з віддалених датчиків;
- на основі аналізу можливостей, переваг та недоліків сучасних систем, обрати найкращий тип для подальшої обробки та збору даних;
- вибір технологій для процесу автоматизації;
- розробити програмну реалізацію автоматизованої системи розумний будинок зі збором, обробкою, передачею даних на сервер та керуванням стану приміщення віддалено через телеграм бота;
- провести тестування на рівні устрою системи сірого та чорного ящика, також навантажене тестування для виявлення максимальної навантаженості системи.

## Висновок до розділу

В даному розділі було розглянуто структури підключення IoT-пристроїв у мережі та з'ясовано основні функціональні можливості і аналоги систем збору даних з віддалених датчиків. Спочатку було проаналізовано систему на базі контролера Arduino Uno. Недоліком першої системи є досить низька частота роботи процесора, замалий об'єм пам'яті для збереження програм, а також відсутня можливість модуля Wi-Fi працювати одночасно в режимах як точка доступу та як клієнт. Взагалом даний контролер дуже влучно використовувати на початкових етапах ознайомлення з

інтернетом речей, оскільки дуже багато документації та існуючих рішень знаходяться у відкритому доступі. Система є відкритою, а отже кожен зацікавлений виробник може вносити власні допрацювання в розвиток Arduino.

Інша розглянута система побудована на основі одноплатного мікрокомп'ютера Raspberry Pi. Мікрокомп'ютери можуть виконувати програми в рамках операційної системи, найчастіше це Linux. На відміну від попередньої системи, Raspberry Pi легко справляється з багатозадачністю, оскільки керується операційною системою. Можна програмувати на будь якій мові, яку підтримує операційна система, а це надає набагато більше можливостей ніж у випадку з попередньою системою. До недоліків такої системи можна віднести споживання струму сотні тисяч міліампер, через це заряду потужного акумулятору вистачить не більше ніж декілька десятків годин. Також з недоліків – це відсутність модулів Wi-Fi та Bluetooth (необхідні додаткові модулі розширення, такі як SPI, SDIO, USB), незважаючи на досить високу вартість такого пристрою.

Врахувавши переваги та недоліки переглянутих систем було визначено мету, об'єкт та предмет дослідження. Також визначено основні завдання, які необхідно виконати для досягнення поставлених цілей.

## 2 ВИБІР АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ

### 2.1 Аналіз компонентів апаратної частини

Для вибору апаратних компонентів проводився аналіз з точки зору ефективності системи та якомога економічної вигоди виготовлення одиниці такої системи із підтримкою поставлених вимог до навантаження, а також можливості масштабування.

Було визначено такі функціональні модулі даної системи:

- головний модуль «Home center»;
- модуль «Light sensor»;
- модуль «Motion sensor»;
- модуль «Smoke sensor»;
- модуль «Climate sensor».

В реалізації даної системи передбачається використання єдиного керуючого модуля «Home center» для обробки даних отриманих з датчиків та управління самими датчиками. Призначенням датчиків інших модулів є збір даних та своєчасна реакція на перевищенні рівні безпечних показів, які регулюються користувачем.

В якості платформи на якій буде базуватись головний модуль системи було визначено що найкраще підходить мікроконтролер ESP32. Це досить сучасний мікроконтролер в якому наявні периферійні інтерфейси та протоколи безпроводного зв'язку, а також досить низька ціна за одиницю такого модуля. Тому саме через підходящі характеристики контролера та його достатню обчислювальну потужність під поставлену задачу було вирішено обрати мікроконтролер ESP32.

#### 2.1.1 ESP32

Мікроконтролер ESP32 вийшов на ринок восени 2015 року, що представляє собою видатний пристрій, але не тільки через низьку ціну. ESP32 – високоінтегрований модуль, що вміщує в собі протоколи технологій Bluetooth та Wi-Fi, виконаний для рішень де необхідні мінімальні енергозатрати.

Даний мікроконтролер отримав значний приріст в продуктивності по зрівнянню із попередником ESP8266. Його обчислювальна потужність виросла у 4 рази. Модуль оснащений двома ядрами, кожне з яких працює на частоті 160МГц. За рахунок другого ядра немає необхідності розподіляти час між виконанням задач, так наприклад одне ядро може відповідати за виконання та обробку задач в реальному часі, а друге ядро може відповідати за протоколи зв'язку тим самим спрощується навантаження [3].

Сьогодні ESP32 одна з найдоступніших потужних платформ для створення IoT. Контролер представляє собою систему побудовану на кристалі з вмонтованими Wi-Fi та Bluetooth модулями. Контролер в собі має двох ядерний процесор, що працює на частотах 80, 160 й 240 МГц. Характеристика контролера (табл. 2.1).

Таблиця 2.1 – Характеристики контролера ESP32

| Параметр                                       | Характеристика             |
|------------------------------------------------|----------------------------|
| Мікроконтролер                                 | ESP32                      |
| Робоча напруга                                 | 2.2 - 3.6 В                |
| Вхідна напруга (рекомендований)                | 5 В                        |
| Цифрові Входи / Виходи                         | 28                         |
| Флешпам'ять                                    | 4 MB                       |
| Тактова частота                                | 240 МГц, 2 ядра            |
| Підтримка протоколів віддаленої передачі даних | Wi-Fi<br>Bluetooth<br>Lora |

Також до системи входять антенні комутатори, радіочастотні компоненти, фільтри, підсилювачі, модулі управління живленням. Для виконання запису та відтворення прошивки ESP32 з'єднується з комп'ютером через USB кабель, або

можлива бездротове з'єднання через протокол Bluetooth. Загальний вигляд контролера зображено на рис. 2.1.

Окремо модуль ESP32 використовують рідко, найчастіше його поєднують із посадковою платою, наприклад DevKit, що зображено на рис. 2.2.



Рисунок 2.1 – Вигляд мікроконтролера ESP32 [3]



Рисунок 2.2 – Вигляд мікроконтролера ESP32 на платі DevKit [3]

Наступні елементи визначають функціональні можливості і параметри плати:

- процесорна частина:
  - а) ядро Tensilica Xtensa LX6;
    - 1. два ядра;

2. 32 розряду;

3. продуктивність до 600 MIPS (мільйонів інструкцій в секунду);

б) співпроцесор з ультранизьким енергоспоживанням.

– пам'ять:

а) 448 KB ROM для завантажувача і функцій ядра;

б) 520 KB SRAM для даних і команд;

в) 16 KB SRAM в RTC (незалежна пам'ять);

г) 4 MB FLASH – пам'ять програми.

– бездротові інтерфейси:

а) Wi-Fi: 802,11 b/g/N;

б) Bluetooth;

в) Lora.

– годинники і таймери:

а) внутрішній генератор 8 мГц;

б) внутрішній RC-генератор;

в) зовнішній генератор 40 мГц;

г) зовнішній генератор 32 кГц для RTC;

д) RTC таймер (годинник реального часу);

е) RTC сторожовий таймер.

– периферійні інтерфейси:

а) 34 програмованих портів введення та виведення;

б) 12ти розрядний АЦП, до 18 каналів;

в) 2 x 8 бітових ЦАП;

г) 10 портів для підключення ємнісних датчиків;

д) 4 x SPI;

е) 2 x I<sup>2</sup> S;

ж) 2 x I<sup>2</sup> C;

з) 3 x UART;

и) 1 хост контролер SD / eMMC / SDIO;

к) 1 слейв контролер SDIO / SPI;

- л) Ethernet;
- м) CAN 2.0;
- н) ІК порт;
- о) ШІМ для двигунів;
- п) ШІМ для світлодіодів до 16ти каналів;
- р) датчик холу.
- безпека:
  - а) безпечне завантаження;
  - б) шифрування FLASH;
  - в) 1024-бітний ключ;
  - г) Криптографічне прискорення:
    - 1. хешування SHA-2;
    - 2. RSA;
    - 3. ETC;
    - 4. генератор випадкових чисел (RNG) [3].

Згідно з розглянутими характеристиками, можна зробити висновок що модуль ESP32 володіє потужними апаратними можливостями. Технології Wi-Fi, Bluetooth Classic і Bluetooth Low Energy доповнюють модуль і завдяки цьому ESP32 є чудовим вибором для будь-яких підключених пристроїв. Вбудований апаратний прискорювач покриває безпечне зберігання коду та безпечне підключення до інтернет мережі за допомогою протоколу TLS.

### 2.1.2 BH1750

Розглянемо наступний модуль системи – «Light Sensor». Цифровий датчик освітленості BH1750, що зображений на (рис. 2.3) використовується для виміру освітлення. BH1750 – 16-бітний датчик освітленості. Даний датчик чудово підходить для отримання даних освітлення навколишнього середовища. Фотодіод, що знаходиться в датчику BH1750 фіксує рівень світла і перетворює його в сигнал, який передається по протоколу I2C до мікроконтролера. Потім мікроконтролер ESP32

приймає рішення відповідно до отриманого цифрового значення, чи є необхідність у ввімкненні світлодіода для підсвічування, або показники освітлення влаштовують. Принцип роботи датчика BH1750 полягає в перетворенні сигналу в електричний, що далі проходить через аналогово-цифровий перетворювач та в результаті модуль інтерпретує дані в люксах, тобто немає необхідності в складних обчисленнях.



Рисунок 2.3 – Цифровий датчик освітленості BH1750 [4]

За даними, що було отримано в специфікації пристрою, цифровий датчик BH1750 є достатньо чутливим до світла та практично не піддається впливу інфрачервоного випромінювання, а це значить, що датчик реагує на такий самий діапазон спектру, що й людське око. Розглянемо характеристики даного датчика (табл. 2.2).

Таблиця 2.2 – Характеристики датчика освітленості BH1750

| Параметр                             | Характеристика   |
|--------------------------------------|------------------|
| Вхідна напруга                       | 2.4-3.6 В        |
| Струм живлення                       | 120 мкА          |
| Струм споживання в режимі очікування | 0,01 мкА         |
| Цифровий інтерфейс                   | I <sup>2</sup> C |
| Довжина хвилі                        | 560 нм           |
| Фільтрація світлових шумів           | 50-60 Гц         |

Даний датчик було обрано для розробки автоматизованої системи оскільки він має властивість максимально точно передавати рівень світла.

### 2.1.3 AM-312

Піроелектричний датчик AM-312, є доступним і універсальним для використання у різних пристроях та системах у поєднанні з мікроконтролером та самостійно, як автономний пристрій. Характеристики даного датчика (табл. 2.3).

Таблиця 2.3 – Характеристики піроелектричного датчика

| Параметр            | Характеристика         |
|---------------------|------------------------|
| Вхідна напруга      | 4.8-20 В               |
| Струм живлення      | 50 мА                  |
| Час затримки        | 0.5-200с (регульована) |
| Кут огляду          | Від 100 градусів       |
| Робоча температура  | 15С...+70С             |
| Визначення об'єктів | 23мм                   |

Принцип роботи датчика AM-312 полягає у тому, що при його вмиканні задається значення інфрачервоного випромінювання. Датчик AM-312 використовує лінзу Френеля за допомогою якої фіксується інфрачервоне випромінювання на пристрій. Потім датчик реагує на перешкоди у просторі за наявності об'єкта-ініціатора. Таким чином датчик генерує електричний струм, коли він отримує інфрачервоне випромінювання. Так як сигнал з датчика не випромінюється, а лише сприймається, то його називають пасивним датчиком. Під час виявлення перешкоди в просторі, датчик змінює свій сигнал на виході. Зовнішній вигляд датчика зображений на рис. 2.4.



Рисунок 2.4 – Датчик руху АМ-312

#### 2.1.4 DHT-11

DHT11 – це цифровий датчик вологості і температури, що зображено на рис. 2.5, він складається з термістора і ємнісного датчика вологості. Для того, щоб обробити та визначити температуру та вологість в приміщенні використовують датчик з аналогово-цифровим перетворювачем (АЦП). Датчик DHT11 не володіють високою швидкістю і точністю, але зате простий, недорогий і відмінно підходять для навчання і контролю вологості в приміщенні [5].

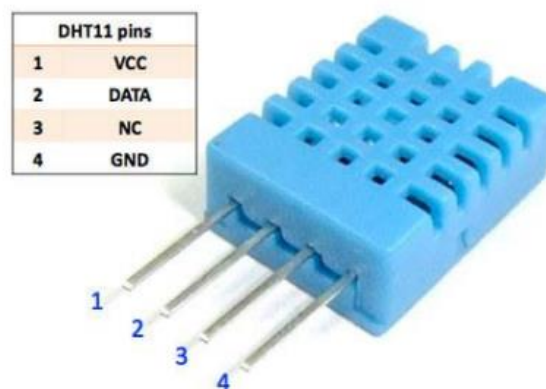


Рисунок 2.5 – Датчик температури та вологості DHT-11 [5]

Датчик має 4 виводи:

- VCC (живлення 3-5 В);
- DATA (вихід даних);
- не використовується;
- GND (земля).

Характеристики датчика DHT-11 (табл. 2.4).

Таблиця 2.4 – Характеристики датчика DHT-11

| Параметр                            | Характеристика                                      |
|-------------------------------------|-----------------------------------------------------|
| Вхідна напруга                      | 3,5-5,5 В                                           |
| Струм живлення в режимі вимірювання | 0.3mA                                               |
| Параметр                            | Характеристика                                      |
| Струм живлення в режимі очікування  | 60μA                                                |
| Визначення вологості                | 20-80% з точністю 5%                                |
| Визначення температури              | 0-50 ° С з точністю 2%                              |
| Частота опитування                  | не більше 1 Гц (не більше одного разу на 1 секунду) |

#### 2.1.5 MQ-2

Датчик MQ-2, що зображено на рис. 2.7 належить до напівпровідникових приладів. Принцип дії датчика, що зображено на рис. 2.6, заснований на зміні опору тонкоплівкового шару діоксиду олова  $\text{SnO}_2$  при контакті з молекулами газу. Чутливий елемент датчика складається з покритою оксидом алюмінію ( $\text{Al}_2\text{O}_3$ ) керамічної трубки і нанесеного на неї чутливого шару діоксиду олова. Всередині трубки проходить нагрівальний елемент, який нагріває чутливий шар до температури, при якій він починає реагувати на визначений газ. Чутливий елемент напівпровідникових газоаналізаторів виготовляється з високоактивного оксиду металу, зазвичай  $\text{SnO}_2$ . При нагріванні поверхні чутливого елемента до певної температури починається поглинання атомів кисню повітря, в результаті якого вони іонізуються, що призводить до збільшення опору чутливого елемента. При наявності

в повітрі газу рівень іонізації атомів кисню зменшується, що призводить до зменшення опору напівпровідника. Таким чином, вимірюючи опір чутливого елемента, можна визначати концентрацію великої кількості газів [6].

У даного датчика виділяють такий діапазон вимірювання газів:

- пропан від (200 – 5000 ppm);
- бутан від (300 – 5000 ppm);
- метан від (500 – 20000 ppm);
- водень від (300 – 5000 ppm) [6].

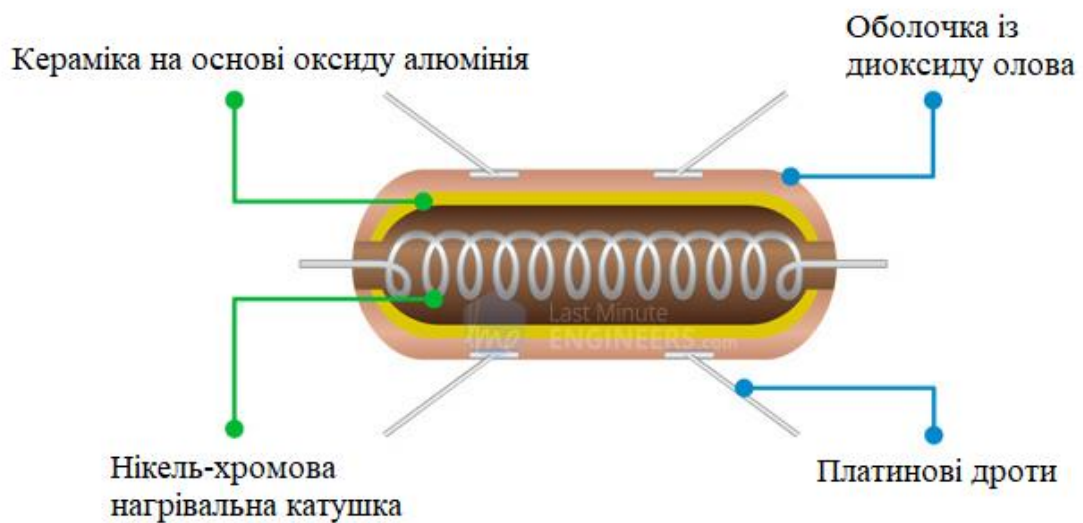


Рисунок 2.6 – Внутрішня структура чутливого елемента датчика газу MQ-2 [7]



Рисунок 2.7 – Датчик газів та диму MQ-2 [7]

Характеристика датчика газу MQ-2 (табл. 2.5).

Таблиця 2.5. – Характеристики датчика MQ-2

| Параметр                 | Характеристика  |
|--------------------------|-----------------|
| Вхідна напруга           | 5 В             |
| Опір навантаження        | 20 кОм          |
| Опір нагрівача           | 33 Ом + 5%      |
| Споживана потужність     | < 800 мВт       |
| Опір чутливості          | 10 кОм – 60 кОм |
| Вимірювання концентрації | 200-10000 ppm   |

Датчик складається з двох шарів тонкої сітки з нержавіючої сталі, яка називається «антивибухова сітка». Даний шар гарантує, що нагрівальний елемент всередині датчика не вибухне. Зовнішні компоненти датчика зображено на рис. 2.8.



Рисунок 2.8 – Зовнішні компоненти датчика MQ-2 [7]

Для того, щоб відкалібрувати датчик газу необхідно повертати потенціометр за годинниковою стрілкою щоб збільшити чутливість, або проти щоб зменшити чутливість, поки на модулі не почне світити червоний світлодіод.

Розглянемо виходи датчика MQ-2, що зображено на рис. 2.9.



Рисунок 2.9 – Терморегулятори датчика газу MQ-2 [7]

Контакти датчика MQ-2:

- VCC забезпечує живлення для модуля, підключити необхідно до виходу 5V на платі DevKit;
- GND земля, необхідно підключити до виходу GND на платі DevKit;
- D0 забезпечує цифрове відтворення проявності небезпечних газів;
- A0 забезпечує аналогову вихідну напругу, що пропорційне концентрації диму та газу.

Компаратор на пристрої постійно перевіряє, чи досяг вихідний сигнал A0 граничне значення, встановленого потенціометром. Якщо A0 перетне граничне значення, тоді цифровий вихід D0 видасть високий логічний рівень і загориться світлодіодний індикатор.

## 2.2 Протоколи зв'язку між компонентами

Розглянемо протоколи та їх характеристику, що використовуються при взаємодії модулів системи.

Протоколи LoRa та LoRaWAN. На фізичному рівні в основі технології LoRa лежить однойменний метод модуляції, який був запатентований компанією Semtech. Цей метод базується на принципі розширення спектру і лінійної частотної модуляції.

В процесі передачі дані кодуються широкополосними імпульсами із зменшенням, або збільшенням частоти. На відміну від інших існуючих мереж, що використовують архітектуру mesh, де вузли мережі для розширення покриття передають інформацію від одного до іншого пристрою, LoRa – мережа використовує топологію «зірка». За рахунок цього вдається зменшити енергетичні споживання пристроїв (через відсутність необхідності пересилання пакетів від інших пристроїв), і спростити таким чином архітектуру мережі. У мережі LoRaWAN вузол пов'язується не з визначеним шлюзом, а передає дані на кілька шлюзів. Кожен шлюз пересилає отриманий пакет від кінцевого вузла через транспорт на хмарний сервер. Сервер керує мережею, відкидає зайві пакети, виконує перевірки безпеки, планує оптимальний маршрут передачі, підтверджує отримання повідомлення та управляє швидкістю передачі даних. Вузли в мережі працюють асинхронно і передають дані по мірі накопичення або по перериванню. Відмова від постійної синхронізації пристроїв (як в mesh, або стільникових мережах) так само дозволяє економити заряд батареї. У мережі з «зоряної» топологією складно організувати велику ємність мережі одночасно з великою площею покриття. Для реалізації такої можливості в LoRaWAN застосовують адаптивну швидкість передачі даних і використовують багатоканальні трансивери в шлюзах, для того щоб повідомлення передавалися відразу по декількох каналах. Так як, LoRa є модуляцією на основі розширення спектра, сигнали практично ортогональні один одному, коли використовуються різні коефіцієнти розширення. При зміні коефіцієнта розширення ефективна швидкість передачі даних теж змінюється. Шлюз використовує цю властивість, маючи можливість отримувати кілька різних швидкостей передачі даних на одному каналі одночасно. Якщо вузол має добре сполучення і знаходиться близько до шлюзу – це дає змогу використовувати більш високу швидкість передачі даних, скорочуючи час в ефірі та відкривати «вікно» для передачі від інших вузлів. Мережа може будуватися виходячи з необхідної ємності, наприклад можна збільшити потужність в мережі, встановити більшу кількість шлюзів, але зменшити кількість шлюзів, яке будуть прослуховувати клієнти і збільшити пропускну здатність каналу в 6-8 разів. Система LoRa може добре підходити для екосистеми розумних міст, так місто – це чітка географічна зона. Хоча

вибір конкретної реалізації часто залежить від топології, щільності забудови та платоспроможності населення. Мережі LoRa за рахунок роботи в неліцензованому діапазоні можуть запропонувати більшу гнучкість і меншу вартість розгортання і володіння в порівнянні з стільниковими мережами. Відмінними ознаками рішень для розумного міста є те, що кінцеві додатки не вимагають постійного широкосмугового підключення і великого часу в ефірі. Такий рівень потреб добре узгоджується з можливостями LoRa. Так, наприклад застосувати таку мережу можна для обміну даними між двома платами ESP32, що мають чіп Loga, які при цьому знаходяться відносно далеко один від одного без покриття Wi-Fi.

Приклад взаємодії декількох пристроїв через протокол Loga зображено на рис. 2.10.

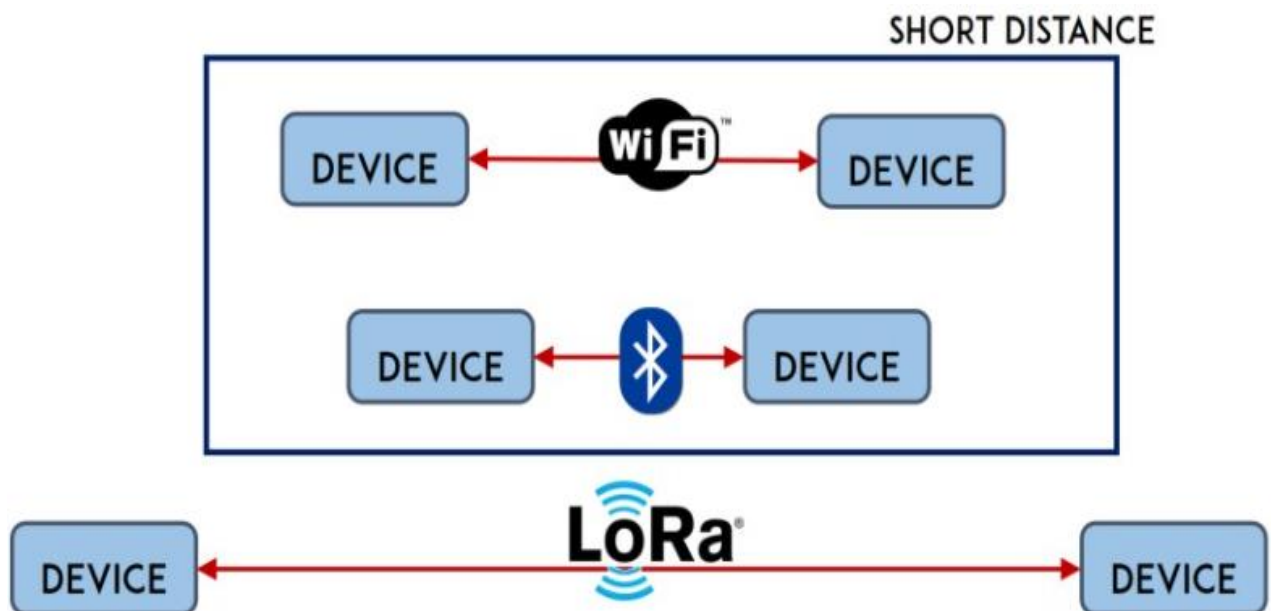


Рисунок 2.10 – Зв’язок двох контролерів ESP32 через протокол Lora

Протокол Wi-Fi. Wi-Fi є найвідомішим бездротовим стандартом у галузі, тому виробник Espressif інтегрував цю технологію в ESP32. Даний протокол використовується практично в кожному розумному будинку. Дуже популярне рішення для пристроїв постійно підключених до електричного живлення. Вузли в мережі Wi-Fi утворюють топологію “зірка”, що означає наявність центрального хаба та інші вузли приєднані до центрального хаба для взаємодії всередині мережі, або з

виходом назовні в глобальну мережу, якщо хаб приєднаний до маршрутизатора, або є сам маршрутизатором.

Тому виділяють два різні режими:

- режим роботи станції як STA (Station mode);
- режим точки доступу AP (Access Point mode).

Наявна можливість налаштувати ESP32 працювати в обох режимах. В режимі STA ESP32 може підключатись до точки доступу та приєднуватись до мережі Wi-Fi як вузол. Коли він перебуває в режимі точки доступу, інші пристрої з підтримкою Wi-Fi можуть підключатись до мережі, яку ESP32 розповсюджує як точку доступу.

На рис. 2.11 зображено обидва режими роботи Wi-Fi, де ESP32 спочатку розповсюджує власну точку доступу для того, щоб клієнт за допомогою неї звернувся до вебсервера для налаштувань параметрів тощо. Після успішних налаштувань параметрів WiFi контролер переходить в стан роботи STA й має змогу надсилати дані в глобальній мережі інтернет.

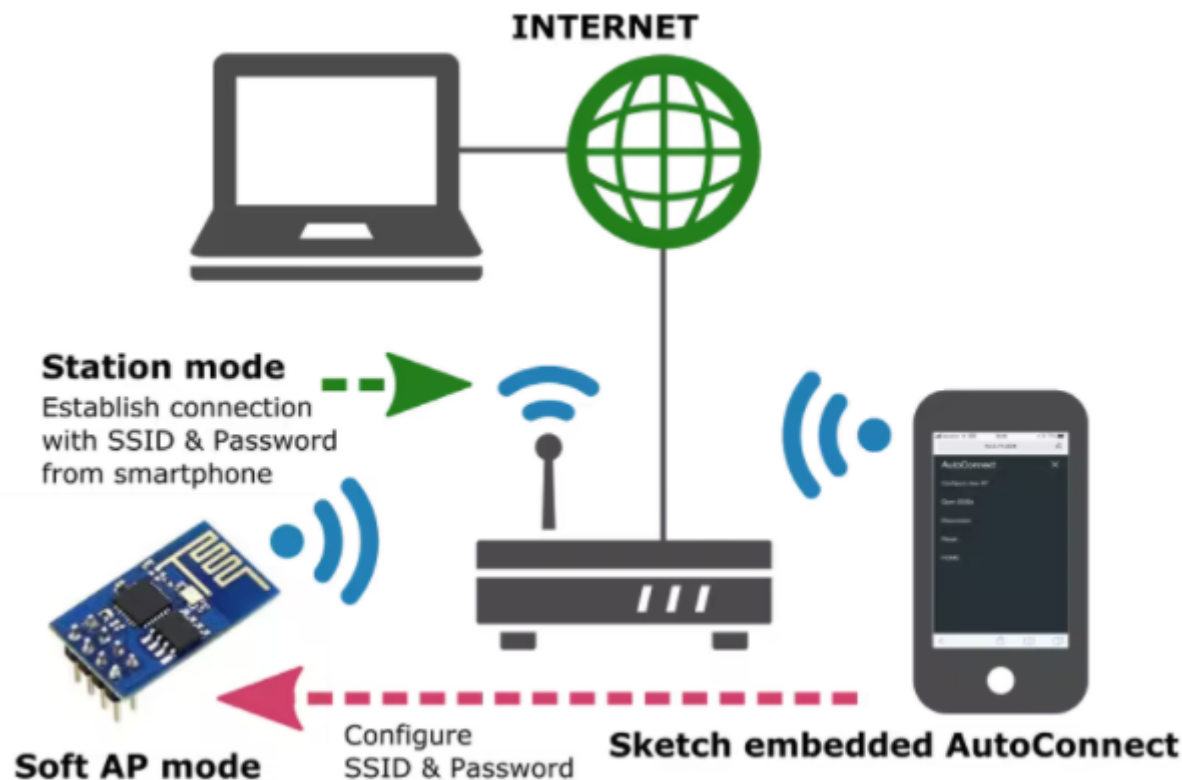


Рисунок 2.11 – ESP32 в режимах STA та AP [8]

Протокол зв'язку I<sup>2</sup>C. ESP32 має два інтерфейси шини I<sup>2</sup>C, які дозволяють працювати йому як головний (Master), або підлеглий (Slave) на шині I<sup>2</sup>C. I<sup>2</sup>C означає Inter Integrated Circuit і є протоколом синхронного зв'язку з декількома провідними і декількома підлеглими пристроями. Ви можете підключити декілька провідних пристроїв до одного майстра: наприклад, ваш ESP32 зчитує дані з датчика, використовуючи I<sup>2</sup>C, і записує покази датчика на OLED-дисплеї. Протокол I<sup>2</sup>C схематично зображено на рис. 2.12.

Відповідно до документації на ESP32, інтерфейси I<sup>2</sup>C ESP32 підтримують:

- стандартний режим (100 кбіт/с);
- швидкий режим (400 Кбіт/с);
- до 5 МГц, але обмежений підтягуванням лінії SDA;
- 7-бітний/10-бітовий режим адресації;
- режим подвійної адресації. Користувачі можуть програмувати регістри команд для керування інтерфейсами I<sup>2</sup>C, щоб вони були універсальними.

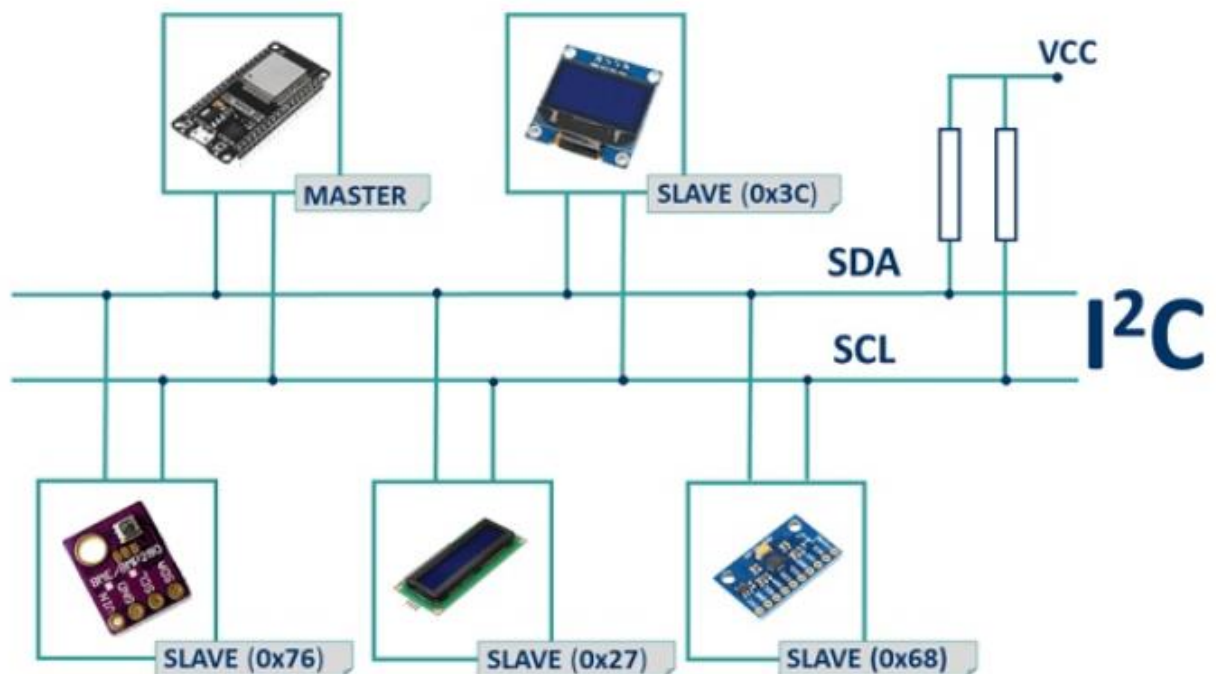


Рисунок 2.12 – Протокол I<sup>2</sup>C [9]

Протокол зв'язку I<sup>2</sup>C використовує два дроти для обміну інформацією. Один використовується для тактового сигналу SCL, а інший використовується для надсилання та отримання даних – SDA. Приклад SDA та SCL інтерфейсів зображено на рис. 2.13.

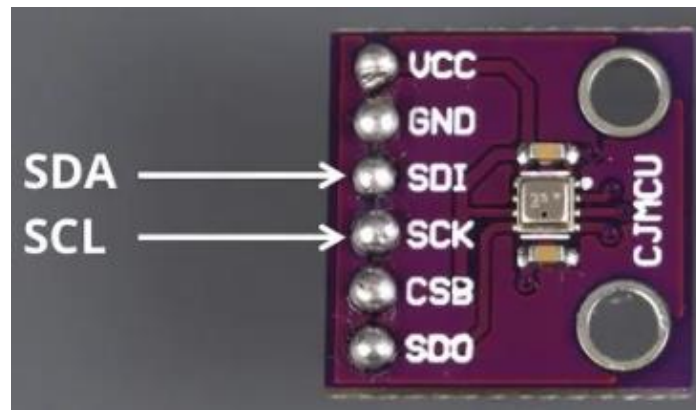


Рисунок 2.13 – SDA(SDI) та SCL(SCK) інтерфейси протоколу I<sup>2</sup>C [9]

Підключення пристрою I2C до ESP32 зазвичай таке ж просте, як підключення GND до GND, SDA до SDA, SCL до SCL і позитивного джерела живлення до периферійного пристрою, зазвичай 3,3 В.

### Висновок до розділу

В даному розділі розглянуто апаратні компоненти модулів системи. Було визначено такі функціональні модулі даної системи: Головний модуль “Home center”, модуль “Light sensor”, модуль “Motion sensor”, модуль “Smoke sensor” та модуль “Climate sensor”.

Платформа, на якій буде базуватись головний модуль системи було визначено, що найкраще підходить мікроконтролер ESP32. Це досить сучасний мікроконтролер в якому наявні периферійні інтерфейси та протоколи безпроводного зв'язку, а також досить низька ціна за одиницю такого модуля. Тому саме через підходящі характеристики контролера та його достатню обчислювальну потужність під поставлену задачу було вирішено обрати даний мікроконтролер.

Для наступного модуля “Light sensor” було обрано цифровий датчик освітленості BH1750. За даними, що було отримано в специфікації пристрою, цифровий датчик BH1750 є достатньо чутливим до світла та практично не піддається впливу інфрачервоного випромінювання, а це значить, що датчик реагує на такий самий діапазон спектру, що й людське око.

Модуль “Motion sensor” базується на основі піроелектричного датчика AM-312. Піроелектричний датчик AM-312, є доступним і універсальним для використання у різних пристроях та системах у поєднанні з мікроконтролером та самостійно, як автономний пристрій. Детектор визначає інфрачервоне випромінювання за допомогою піроелектричного датчика. Датчик генерує електричний струм, коли до нього надходить інфрачервоне випромінювання. Оскільки сигнал з датчика не випромінюється, а тільки сприймає, то його називають «пасивним». При виявленні порушення цілісності простору, датчик змінює сигнал на виході.

Наступний модуль – “Smoke sensor”. За основу взятий датчик газу та диму MQ-2. Принцип дії датчика заснований на зміні опору тонкоплівкового шару діоксиду олова при контакті з молекулами газу. Чутливий елемент датчика складається з покритою оксидом алюмінію керамічної трубки і нанесеного на неї чутливого шару діоксиду олова. Всередині трубки проходить нагрівальний елемент, який нагріває чутливий шар до температури, при якій він починає реагувати на визначений газ. При нагріванні поверхні чутливого елемента до певної температури починається поглинання атомів кисню повітря, в результаті якого вони іонізуються, що призводить до збільшення опору чутливого елемента.

Останній модуль системи – “Climate sensor”, що складається з датчика температури та вологості DHT11. Це цифровий датчик вологості і температури що складається з термістора і ємнісного датчика вологості. Для того, щоб обробити та визначити температуру та вологість в приміщенні використовують датчик з аналогово-цифровим перетворювачем (АЦП).

Також було визначено інтерфейси зв'язку компонентів між собою. Серед них: Lora, Wi-Fi, I<sup>2</sup>C.

Мережі LoRa за рахунок роботи в неліцензованому діапазоні можуть запропонувати більшу гнучкість і меншу вартість розгортання і володіння в порівнянні з стільниковими мережами. Так, наприклад застосувати таку мережу можна для обміну даними між двома платами ESP32, що мають чіп Lora, які при цьому знаходяться відносно далеко один від одного без покриття Wi-Fi.

Wi-Fi є найвідомішим бездротовим стандартом у галузі, тому виробник Espressif інтегрував цю технологію й в ESP32. Даний протокол використовується практично в кожному розумному будинку. Дуже популярне рішення для пристроїв, постійно підключених до електричного живлення.

I<sup>2</sup>C означає Inter Integrated Circuit і є протоколом синхронного зв'язку з декількома провідними і декількома підлеглими пристроями. ESP32 має два інтерфейси шини I<sup>2</sup>C, які дозволяють працювати йому як головний (Master), або підлеглий (Slave) на шині I<sup>2</sup>C.

## 3 ВИБІР ТЕХНОЛОГІЙ ДЛЯ ПРОЦЕСУ АВТОМАТИЗАЦІЇ ЗБОРУ ДАНИХ З ВІДДАЛЕНИХ ДАТЧИКІВ В СИСТЕМІ РОЗУМНИЙ БУДИНОК

### 3.1 C++

Мова програмування C++ представляє собою об'єкто-орієнтовану низькорівневу мову програмування, що відповідає стандартам ANSI і міжнародній організації стандартів ISO(International standards organization). Об'єктно-орієнтованість C++ дає можливість дотримуватись єдиного стилю програмування при великомасштабному кодуванні та забезпечує їх розширювання. Мова C++ початково походить від мови C та допрацьована під сучасні технічні потреби. Сьогоднішні можливості C++ мають оголошення функцій у вигляді виразів, перетворення типів у вигляді функцій, поява операторів new і delete, тип даних bool, посилання, константи, аргументи за змовчуванням, перевизначення, простори імен, класи (такі можливості як успадкування, віртуальні функції та класи), перевизначення операторів, шаблони, динамічна ідентифікацію та інше. C++ має деякі важкі для розуміння концепції, які характеризували C. Зокрема, вказівники є складною концепцією і їх неправильне використання може призвести до збоїв системи та небезпечного використання

пам'яті. Відсутність збирача сміття також ускладнює фільтрацію непотрібних даних. Іншим обмеженням для C++ є наявність проблем безпеки, пов'язаних з доступністю показників, глобальних змінних [10].

На мові C++ ми будемо програмувати апаратне забезпечення в середі розробки Arduino IDE. IDE розшифровується як інтегрована середа розробки в якому пишуть програмний код, а препроцесор та компілятор перевіряють код, який потім завантажують на пристрій. Перед завантаженням кода програми необхідно налаштувати порт до якого підключена плата і саму платформу, наприклад ESP32.

Система вкладок Arduino IDE працює досить незвичайним чином і дуже відрізняється від поняття вкладок в інших програмах:

- вкладки відносяться до одного і того ж проєкту, до файлів, що знаходяться з ним в одній папці;
- вкладки просто ділять загальний код на частини, тобто в одній вкладці фігурна дужка на початку функції може відкритись, а в наступній вкладці може закритись. При компіляції коду всі вкладки поєднуються в один текстовий файл по порядку зліва направо (з лівої вкладки до правої). Також це означає, що вкладки повинні містити код, що стосується конкретного одного проєкту та виконувати функцію `setup()` та цикл `loop()` тільки в одній вкладці так як вони можуть бути тільки в одному екземплярі;
- вкладки автоматично розташовуються в алфавітному порядку, тому вкладка, що створюється, може виявитися між вже існуючими. Це означає, що розбивати блоки коду за різними вкладками (як в другому пункті, де початок на одній вкладці та кінець функції на іншій вкладці) не рекомендується;
- також змінна повинна бути оголошена до свого виклику, тобто вкладка з оголошенням змінної повинна бути ліворуч від вкладки, де змінна викликається. Створюючи нову вкладку треба одразу розуміти де вона з'явиться з таким ім'ям і чи не буде через це проблем. Також назву вкладок можна починати з цифр і таким чином точно контролювати їхній порядок.
- вкладки зберігаються в папці з проєктом і мають розширення `“.ino”` та при запуску будь-якої вкладки відкриється весь проєкт з усіма вкладками;

– крім файлів “.ino” Arduino IDE також вивантажує файли з розширеннями “.h” (файл бібліотека), “.cpp” (файл реалізації) та “.pde” (старий формат файлів Arduino IDE). Ці файли так само з'являються у вигляді вкладок, але наприклад, заголовний файл “.h” не бере участь у компіляції поки не буде підключений до проєкту за допомогою команди `include`. Тобто він буде виглядати як вкладка і його можна редагувати, але без підключення так і залишиться просто окремим текстом [11].

### 3.2 Telegram Bot API

Bot API являє собою HTTP-інтерфейс для роботи з ботами в месенджері Telegram. Бот – це спеціальний обліковий запис для автоматичного оброблення та відправлення повідомлень.

Документація Telegram Bot API виділяє два способи отримання оновлень:

- запити з певною періодичністю;
- встановлення веб-хуків.

Оновлення не надходять до того моменту як сервер не опрацює їх. Це займає до 24-х годин. Як результат ми маємо об'єкт “Update”, який подано у вигляді JSON.

Одним з найпростіших варіантів опитування – є запити до серверів Telegram про актуальність інформації. Відбувається конект на короткий час і зміни надсилаються Telegram боту і проходить через протокол “Long Polling”. Насправді це елементарний спосіб, проте він не є дуже надійний.

Веб-хуки це зовсім інше. Якщо в чат надходить повідомлення, то Telegram сам сповіщує, в цьому і полягає головна ознака вебхука. Тепер необхідності в регулярному опитуванні сервера немає, саме через це зникають помилки пошукових роботів. Однак необхідно доплачувати за встановлення повноцінного веб-сервера на пристрій, де й запускають пошукових ботів.

Для зв'язку через вебкуки потрібно мати власний SSL сертифікат (Secure Sockets Layer), адже веб-хуки в Telegram працюють лише по протоколу HTTPS. Схематично зображено обидва способи взаємодії з Telegram Bot API на рис. 3.1.

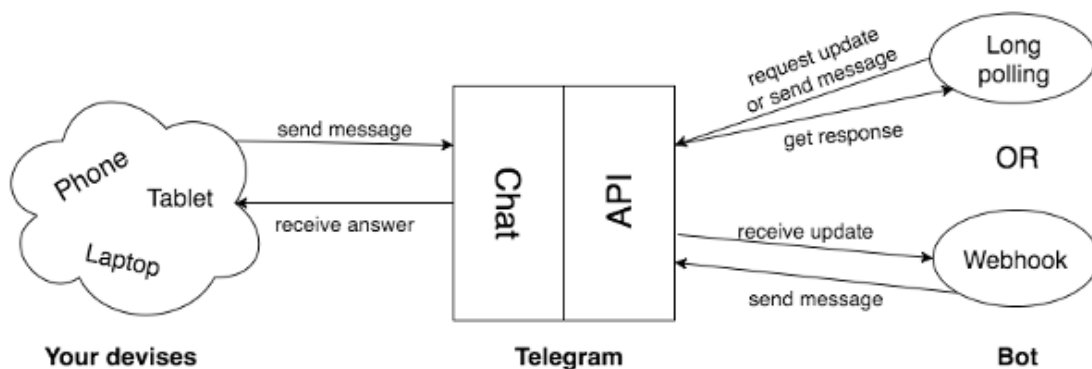


Рисунок 3.1 – Способи отримання оновлень через Telegram Bot API [12]

Для отримання токена необхідно написати спеціальному боту “@BotFather”.

Наведемо приклади доступних методів API:

- `getUpdates` – це метод, що використовує протокол “Long Polling” для отримання оновлень;
- `setWebhook` – це метод, що з'єднує бота з файлом-обробником;
- `sendMessage` – це метод, який відправляє повідомлення користувачу в Telegram;
- `sendLocation` – це метод, який відправляє дані з координатами місцезнаходження пристрою;
- `getFile` – це метод, який повертає файли по їх назві.

Використовуються запити з методами POST і GET. Існують 4 варіанти передачі параметрів в Bot API:

- запит через пошукову строку URL;
- використовуючи формат `application / x-www-form-urlencoded`;
- також формат `application / json`;
- `multipart/form-data` (підходить для завантаження файлів).

Отже, в процесі опису чат-ботів було здійснено вибір месенджеру Telegram. Це зумовлено наявністю гарно задокументованого API, а саме: методи передачі та кодування даних, аутентифікація на серверах Telegram, методи API для отримання даних та передачі команд.

### 3.3 HTML, CSS та Javascript

Вперше розроблений Тімом Бернерсом-Лі в 1990 році, HTML є скороченням від Hypertext Markup Language. HTML використовується для створення веб-сторінок, що відображаються в мережі інтернет. Кожна сторінка містить певну кількість підключень до інших сторінок, які називаються гіперпосиланнями. Всі існуючі веб-сторінки, були написані з використанням мови розмітки HTML.

HTML-код забезпечує правильне форматування тексту та зображень для браузера. Сам браузер не розуміє й не знає як зображувати текст та елементи дизайну. Завдяки HTML є можливість накладати каскадні таблиці стилів на базову структуру сторінки щоб змінювати зовнішній вигляд. Можна вважати, що HTML це правила про структуру веб-сторінки, а CSS як про її зовнішній вигляд. На рис. 3.2 зображено HTML теги та CSS атрибути, що зазвичай має кожен елемент розмітки.

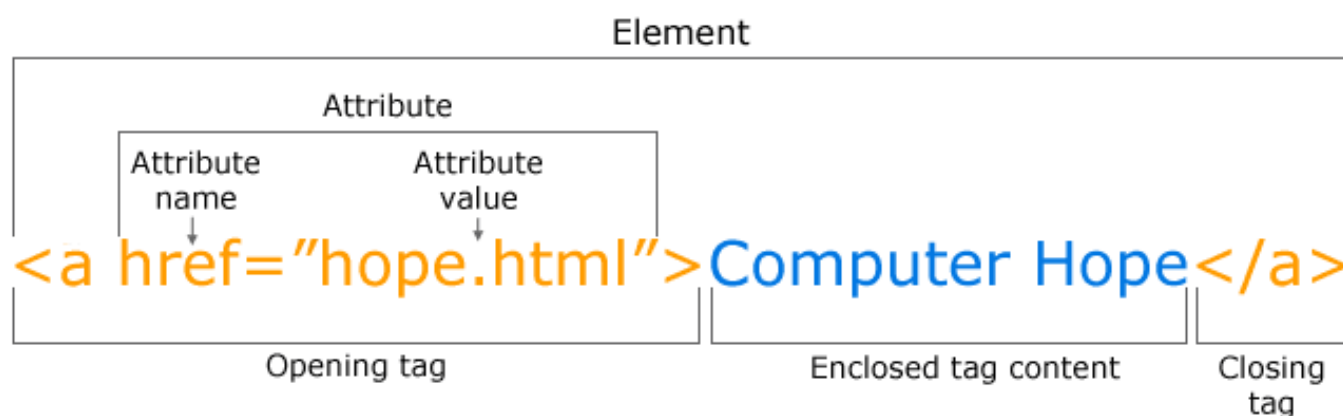


Рисунок 3.2 – HTML тег та CSS атрибут елемента розмітки [13]

Практично всі HTML теги мають відкриваючий тег, що містить назву тегу, атрибути тегу, закриваючий тег, що містить пряму косу риску, і назву тегу, що закривається. Для тегів, які не мають закривального тега, наприклад <img>, необхідно закривати тег тільки косою рискою без назви тегу.

Майже всі теги знаходяться в кутових дужках і все наповнення, що знаходиться між відкритим і закритим тегом, відображається або впливає на теги. На рис. 3.2 тег <a> створює посилання під назвою «Computer Hope», яке вказує на файл "hope.html".

Це означає, що при переході на веб-сторінці по даному посиланню браузер повинен перенаправити на вказану сторінку.

Зазвичай базова структура файлу html складається з наступних тегів:

- найвищий рядок DOCTYPE описує, якою версією HTML була написана сторінка, щоб браузер міг інтерпретувати текст, що передається;
- тег <HTML> дозволяє браузеру розуміти, що він читає HTML код;
- розділ <head> містить інформацію про сторінку, таку як її назва, мета-теги та місце розташування файлу CSS;
- розділ <body> містить все, що можна переглянути у браузері. Наприклад, весь текст, який тут бачите, міститься в тегах body;
- тег <H1> є видимим заголовком сторінки;
- тег <P> абзацу тексту. Більшість веб-сторінок (наприклад, ця) мають кілька тегів абзацу;
- в абзаці міститься тег <b>, який виділяє жирним шрифтом слово приклад в абзаці;
- закриваючими тегами обгортають кожен із зазначених вище тегів.

Файли HTML використовують розширення .htm або .html. Старіші версії Windows (Windows 3.x) дозволяють лише трибуквені розширення файлів, тому вони використовували .htm замість .html. Однак обидва розширення файлів мають однакове значення, і кожне з них може використовуватися сьогодні. З огляду на це, ми рекомендуємо дотримуватися однієї конвенції щодо імен, оскільки певні веб-сервери можуть віддавати перевагу одному розширенню перед іншим [13].

CSS допомагає веб-розробникам створити уніфікований вид багатьох сторінок веб-сайту. Для того, щоб не визначати, наприклад, для кожної таблиці одні й ті самі налаштування позицій часто використані стилі необхідно визначити тільки один раз у документі CSS. Далі стиль визначено в каскадній таблиці стилів, його може використовувати кожна сторінка, в якій є посилання на файл CSS. Також CSS дає можливість змінювати стилі на декількох сторінках в один і той час. До прикладу, можна навести ситуацію, коли розробнику необхідно збільшити розмір тексту з 10 до

14 для десятків сторінок веб-сайту. Коли всі сторінки посилаються на один CSS файл, тоді розмір тексту необхідно змінити тільки на таблиці стилів.

CSS дозволив кілька нововведень у макеті веб-сторінки, наприклад, такі можливості:

- вказувати шрифти, відмінні від стандартних для браузера;
- вказувати колір й розмір тексту та посилань;
- застосовувати кольори до фону;
- містити елементи веб-сторінки в полях і переміщувати ці поля на певні позиції на сторінці.

Хоча CSS чудово підходить для створення стилів тексту, він також корисний для форматування інших аспектів макета веб-сторінки. Наприклад, CSS можна використовувати для визначення заповнення клітинок таблиці, стилю, товщини та кольору межі таблиці, а також заповнення навколо зображень чи інших об'єктів. CSS дає веб-розробникам більш точний контроль над тим, як виглядатимуть веб-сторінки, ніж HTML. Ось чому більшість веб-сторінок сьогодні включають каскадні таблиці стилів [14].

Сучасний JavaScript є «безпечною» мовою програмування. Він не надає низькорівневого доступу до пам'яті чи ЦП, оскільки спочатку був створений для браузерів, яким це не потрібно. Функції JavaScript залежать від середовища, де він працює. Як приклад, фреймворк Node.js дає можливість JavaScript зчитувати/записувати файли, виконувати запити по мережі тощо. JavaScript в браузері може маніпулювати веб-сторінками, взаємодіяти з користувачем і веб-сервером.

Наприклад, JavaScript в браузері здатний виконувати такий функціонал:

- додавати новий HTML на сторінку, змінювати наявний вміст та стиль;
- реагувати на дії користувача, клацання миші й натискання клавіш;
- відправляти запити по мережі на віддалені сервери, завантажувати та вивантажувати файли (технології AJAX і COMET);
- отримувати та встановлювати файли cookie, задавати запитання відвідувачеві, показувати повідомлення;
- пам'ятати клієнтські дані [15].

Можливості JavaScript в браузері обмежені через безпеку користувача.

Як приклад, можуть бути такі обмеження:

– мова Javascript на веб-сторінці не може зчитувати та записувати файли на жорсткому диску, копіювати їх або виконувати програми на комп'ютері. Він не має прямого доступу до функцій операційної системи. Сучасні браузери дозволяють працювати з файлами, але доступ обмежений і надається лише в тому випадку, якщо користувач виконує певні дії, наприклад вибирає файл за допомогою тегу `<input>`. Існують способи взаємодіяти з камерою, або з іншими пристроями, які потребують дозволу користувача;

– різні вкладки та вікна зазвичай не знають про існування інших. Інколи відбувається так, що одна вкладка використовує JavaScript для відкриття іншої. Зазвичай, JavaScript не має доступу до іншого вікна, якщо він надходить з другого ресурса (інший домен, протокол або порт). Щоб обійти це обмеження, обидві сторінки мають погодитися на обмін даними та містити спеціальний код JavaScript, який це обробляє;

– мова JavaScript може зв'язуватись з сервером через мережу саме з якого прийшла поточна сторінка. Але здатність отримувати дані з інших сайтів обмежена, оскільки це вимагає явної угоди, що вказується в заголовках HTTP з віддаленої сторони. Але все ж таки це обмеження безпеки [16]. Приклад зображено на рис. 3.3.

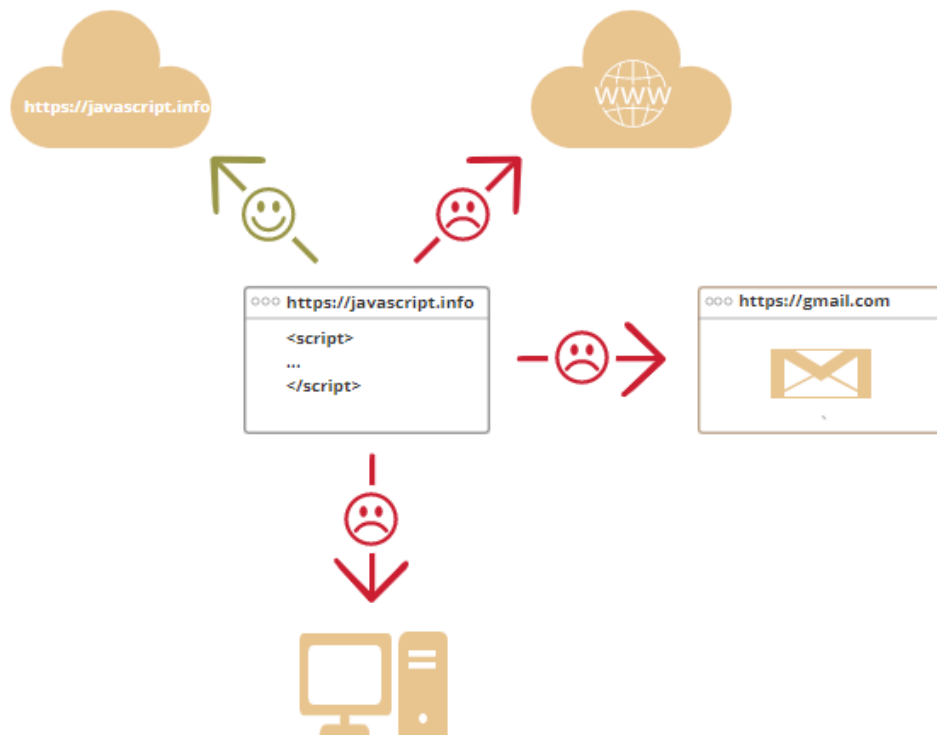


Рисунок 3.3 – Обмеження безпеки отримування даних зі сторонніх ресурсів [17]

### 3.4 Вибір бази даних

На сьогодні багато технологічних підприємств виготовляють додатки Інтернету речей. Наразі пристрої IoT обробляють такі об'єми даних, які раніше не уявлялись можливим. Наприклад, компанія Bosch використовує дані зчитані з датчиків в цілому ряді промислових додатків Інтернету речей, що включає автомобільне виробництво, енергетика та роздрібна торгівля. Відповідно настає необхідність в збереженні великої інформації даних в сховищі. Для цього потрібно використовувати бази даних.

Розглянемо типи систем управління базами даних, які потенційно можуть підходити для нашої системи. Одною з таких є MySQL. MySQL це система управління реляційними базами даних (СУБД) з відкритим вихідним кодом з моделлю клієнт-сервер. СУБД – це спеціальне програмне забезпечення, що розроблене для управління базами даних. Існує багато СУБД, наприклад: PostgreSQL, SQLite, Oracle, MsSQL. Реляційна база даних – це набір структурованих даних, що організовані у вигляді таблиць. Ми можемо за допомогою реляційної бази даних

досить просто відображати зв'язки між таблицями. Нормалізувавши дані ми можемо заставляти змінювати дані атомарно в одному місці. Відкритий код означає, що будь хто може вільно використовувати вихідний код, а також змінювати його. Тобто, при необхідності можна вивчити та налагодити вихідний код таким чином, щоб він краще відповідав вимогам клієнта. По функціональному оснащенні та надійності MySQL конкурує з іншими відомими продуктами. Найчастіше використовують при розробці веб-рішень, що пояснюється високою інтеграцією з популярними мовами програмування, високими показниками швидкості та тим, що дане рішення є безкоштовним.

#### Переваги MySQL:

- відкритий вихідний код. Є можливість безкоштовного використання;
- середовище бази даних MySQL встановлюється з легкістю, має зрозумілий інтерфейс та досить велику кількість плагінів, що зпрощує роботу з базою даних;
- функціонал включає в собі майже весь необхідний набір інструментів, який може знадобитись при розробці проєкту;
- безпека. Більшість систем безпеки вбудовано та працюють по замовчуванню;
- масштабованість. Може використовувати великий обсяг даних;
- швидкість. Вважається одною із найшвидких серед існуючих СУБД.

Таким чином, на MySQL звертають увагу як невеликі компанії, так і великі корпорації. Ця СУБД приваблює своєю надійністю та вільним простором. Якщо необхідно розробити веб-функціонал, то MySQL практично завжди буде найкращим варіантом при виборі СУБД.

Іншим типом баз даних є нереляційні бази даних. Розглянемо найпопулярнішу нереляційну базу даних. MongoDB має сучасний підхід до створення баз даних, де відсутні звичайні нам таблиці, схеми, SQL запити, первинні та зовнішні ключі, що притаманне для реляційних баз даних. На відміну від реляційних баз даних MongoDB використовує документо-орієнтовану модель даних, зарахунок чого MongoDB працює швидше, має кращу масштабованість. Також MongoDB покращить продуктивність додатку, наприклад, якщо необхідно зберігати важкі структури даних. Вся система MongoDB може представляти не тільки одну базу даних, що

знаходиться на одному фізичному сервері, також функціональність MongoDB дозволяє розташувати декілька баз даних на декількох фізичних серверах. В Mongo обмін даними відбувається по JSON (Javascript Object Notation) формату. З точки зору MongoDB це являється перевагою, що використовується гнучкий формат передачі документів. Також це зручніше, оскільки немає необхідності в написанні трудомістких SQL-запитів. З точки зору продуктивності в основному проблеми виникають коли виконуються SQL-запити із оператором JOIN з багатьма таблицями. Якщо говорити про додатки, де використовують MongoDB – це зазвичай стартапи чи проєкти з швидкою розробкою. Це відбувається тому, що виникає необхідність постійно вносити зміни і не треба дотримуватись суворого формату документа як це зазвичай в реляційних базах даних [18].

На рис. 3.4 зображено синтаксичні відмінності при написанні запиту на вставку нового рядка для реляційної та нереляційної баз даних.

## SQL vs CRUD - Insert

### • SQL

```
INSERT INTO book (  
  'ISBN', 'title', 'author'  
)  
VALUES (  
  '9780992461256',  
  'Full Stack JavaScript',  
  'Colin Ihrig & Adam Bretz'  
);
```

### • CRUD

```
db.book.insert({  
  ISBN: "9780992461256",  
  title: "Full Stack JavaScript",  
  author: "Colin Ihrig & Adam Bretz"  
});
```

Рисунок 3.4 – Синтаксис запиту Insert для SQL та NoSQL баз даних [18]

На рис. 3.4 можна побачити, що такі прості операції як вставка, оновлення, видалення працюють досить добре та інтерпретація цих запитів в нереляційних базах виглядає більш читабельною. Але, коли виконуються трохи складні операції вибірки, тоді SQL запит виглядає більш читабельним, що зображено на рис. 3.5.

# SQL vs CRUD - Search

---

```
SELECT title FROM book  
WHERE price > 10;
```

```
db.book.find(  
  { price: { $gt: 10 } },  
  { _id: 0, title: 1 }  
);
```

Рисунок 3.5 – Синтаксис запиту Select для SQL та NoSQL баз даних [18]

В якості бази даних найкраще підійде MongoDB, оскільки наша система може масштабуватись, а отже ймовірно постане необхідність в зміні структури бази даних.

## Висновок до розділу

В даному розділі розглянуто технології згідно з якими буде розроблено систему автоматизованого збору та обробки даних з датчиків робототехнічного модуля в системі розумний будинок. На мові C++ ми будемо програмувати апаратне забезпечення в середі розробки Arduino IDE. Було з'ясовано, що перед завантаженням кода програми в мікроконтролер необхідно в середовищі Arduino IDE налаштувати порт до якого буде підключена плата і саму платформу ESP32, яку було розглянуто в попередньому розділі. Було проаналізовано, що досить зручно отримувати повідомлення про стан датчиків системи та керувати датчиками через чат-ботів. В процесі опису чат-ботів було здійснено вибір месенджера Telegram. Це зумовлено наявністю гарно задокументованого API, а саме: методи передачі та кодування даних, аутентифікація на серверах Telegram, методи API для отримання даних та передачі команд. Також до функціоналу системи входить конфігураційна HTML сторінка користувача. Керувати чат-ботом можна тільки по заздалегідь підключеній сиситемі до Wi-Fi. Тому необхідно розробити таку сторінку з можливістю налаштовувати

параметри підключення до мережі Wi-Fi. Для розробки такого функціоналу будуть використовуватись технології HTML, CSS та Javascript. Порівняли реляційні та нереляційні бази даних. З'ясували переваги та недоліки відповідно MySQL та MongoDB баз даних та як результат було обрано нереляційну базу даних MongoDB, оскільки наша система може масштабуватись, а отже ймовірно постане необхідність в зміні структури бази даних.

## 4 РОЗРОБЛЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ РОЗУМНИЙ БУДИНОК ІЗ КЕРУВАННЯМ ЧЕРЕЗ ТЕЛЕГРАМ БОТА

### 4.1 Архітектура системи

Архітектура автоматизованої системи наведено в додатку А та формується з наступних компонентів:

- модуль «Датчик освітлення» відповідає за зняття показів освітлення приміщення;
- модуль «Датчик руху» орієнтований на сприйняття будь-якого руху у просторі;
- модуль «Датчик газу та диму» відповідає за рівень вміщення шкідливих газів у повітрі. Спроможний ідентифікувати та виокремлювати в повітрі залишки метану, пропану, бутану, водню, а також диму;
- модуль «Датчик температури та вологості» відповідає за зняття показів температури та вологості в приміщенні;
- модуль «Виконання внутрішніх процесів» вміщує в собі ініціалізацію змінних, програмне підключення датчиків до мікроконтролера, виконання калібрування датчиків перед виконанням зняття показів;
- модуль «Зчитування датчиків» відповідає за міри та величини даних отриманих з датчиків. А також відповідає за періоди опитування датчиків;
- модуль «Контроль флешпам'яті» відповідає за належний запис та зчитування даних до флешпам'яті. Також контролює, щоб не відбувалось перевищування загального обсягу даних в накопичувачі;
- модуль «Вебсервер» надає можливість контролеру працювати в режимі вебсервера. Таким чином контролер розповсюджує власну точку доступу для того, щоб налаштовувати параметри контролера віддалено з іншого будь-якого пристрою, що підтримує WiFi протокол;
- модуль «Клієнт» відповідає за своєчасні запити до віддаленого вебсервера для передачі масиву даних, отриманих з датчиків. Також даний модуль відповідає за відправку повідомлень до чат-боту.

## 4.2 Загальна схема взаємодії компонентів системи

Структурну схему взаємодії компонентів та модулів системи наведено в додатку Б.

На схемі датчик руху, датчик газу та диму, датчик температури та вологості, датчик освітленості – це незалежні один від одного датчики, що виконують роль збору даних з навколишнього середовища, поки це необхідно. Далі дані оброблюються за допомогою зчитувача датчиків.

Зчитувач датчиків – окрема підпрограма виконана на мові програмування C++, що обробляє дані із датчиків та передає їх в область даних.

Область даних – компонент, що зберігає змінні та об'єкти різних типів даних.

Вебсервер – модуль, що віддає клієнту (Браузер) HTML-сторінку для введення параметрів Wi-Fi мережі, через яку потім дані будуть передаватись на зовні. Тобто встановивши таким чином з'єднання матимемо змогу спілкуватись із клієнтською частиною по клієнт-серверній архітектурі.

SPIFFS – файлова система флешпам'яті, яка підключається по послідовному периферійному пристрою. Найголовніша функція SPIFFS в даній системі – зберігати налаштовані дані.

Контролер – підпрограма, що написана на мові C++. Виконує звернення до Telegram Bot API, наприклад в разі коли показ датчика перевищив граничне значення, в такому випадку Контролер передає дані з датчика та вказує по якій кінцевій точці необхідно виконати запит.

Клієнт – функціонал, що забезпечує опитування та передачу даних з датчиків до віддаленого вебсерверу.

Віддалений вебсервер – опрацьовує запити з Клієнта та виконує відповідний запит на операції CRUD до бази даних.

Користувач – має можливість отримувати повідомлення від системи, а також посилати команди на сервер.

Telegram Bot API – це сервіс зі сторони месенджера телеграм, через який відбувається взаємодія клієнта з сервером та навпаки. У випадку коли система

зафіксувала граничне значення датчика, тоді Контролер збирає необхідний масив даних та передає сервісу Telegram Bot API, який в свою чергу встановлює зв'язок з адресою користувача та в результаті віддає йому дані. Схему доставки повідомлень до користувача наведено в додатку Г.

Візуальне представлення спілкування клієнта та сервера за технологією “webhook” зображено на рис. 4.2.

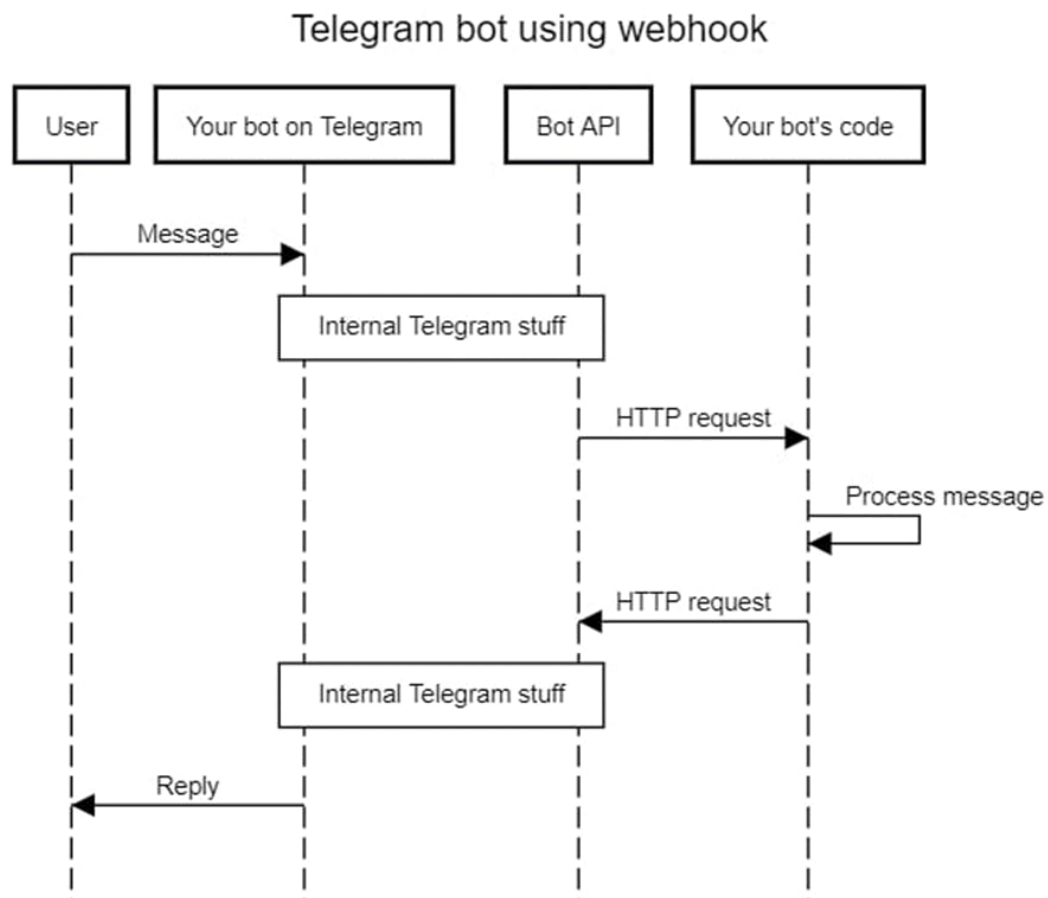


Рисунок 4.2 – Діаграма послідовності взаємодії по технології “webhook” [19]

### 4.3 Розроблення серверної частини системи

Розроблення серверної частини ділиться на дві частини. Перша – це коли мікроконтролер ESP32 працює як вхідна точка доступу та друга частина – де відбувається саме збір показів з датчиків та режим роботи контролера як повноцінний веб-сервер. На рис. 4.3 зображено спрощену блок-схему, де якщо модулем ESP32 приєднатись до існуючої точки доступу Wi-Fi з виходом в мережу, тоді в такому

випадку система буде працювати в стані веб-сервера, якщо ж ні, тоді як власна точка доступу. Більш детальний алгоритм роботи серверної частини надано в блок-схемі додатку В.

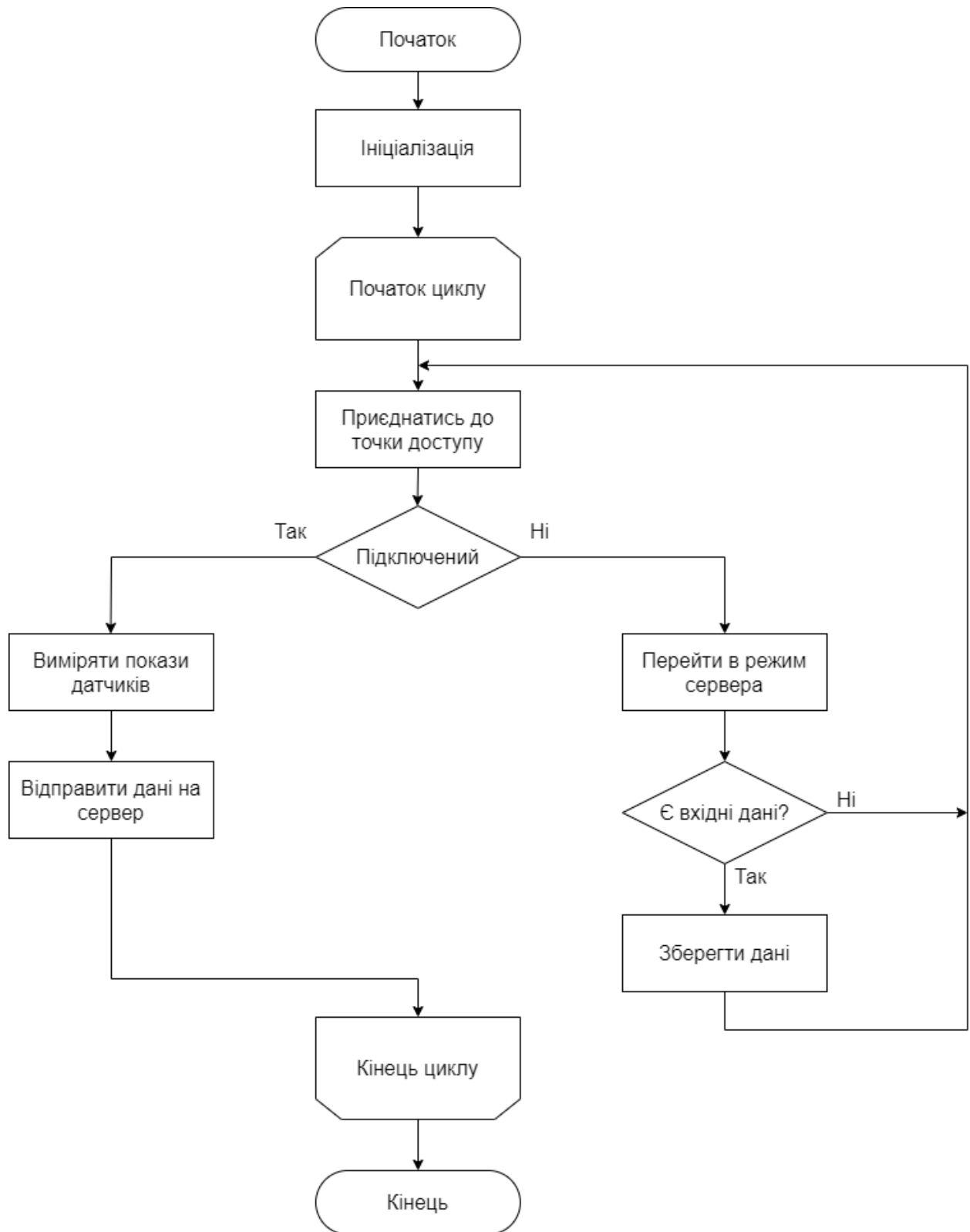


Рисунок 4.3 – Блок-схема принцип роботи веб-сервера

Розглянемо першу частину.

ESP32 створює власну мережу Wi-Fi та діє як розповсюджувач для кількох пристроїв та називається точкою доступу (Access Point). На відміну від Wi-Fi роутера в нього немає інтерфейса для дротової мережі. Такий режим роботи називається “Soft Access Point”. Також максимальна кількість пристроїв, що можуть підключитись обмежено п’ятьма.

Реалізація розпочинається з того, що підключаємо необхідні бібліотеки для розробки системи. На рис. 4.4 зображено підключення бібліотек.

```
#include <AsyncTCP.h>
#include <Wire.h>
#include <BH1750.h>
#include <WiFiMulti.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiAP.h>
#include "SPIFFS.h"
#include "FS.h"
#define FORMAT_SPIFFS_IF_FAILED true
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
```

Рисунок 4.4 – Підключення бібліотек

ESP32 в режимі точки доступу створює нову мережу Wi-Fi та встановлює для неї SSID (назва мережі) та PASS (пароль), але це не обов’язково. Також встановлюється IP-адреса з якою модуль ESP32 може віддавати веб-сторінки на всі підключені пристрої в своїй мережі. Методи, що встановлюють кредити для мережі, а також ініціалізація IP-адреси можна побачити на рис. 4.5.

```
WiFi.softAP(ssid, password);
IPAddress myIP = WiFi.softAPIP();
```

Рисунок 4.5 – Методи конфігурації “Access Point”

Далі точка доступу знаходиться в активному стані і може під'єднувати інші пристрої. Щоб звернутись до веб-сервера ESP32 необхідно в браузері ввести IP-адресу сервера. На рис. 4.6 зображено метод, який очікує вхідних клієнтів.

```
WiFiClient client = server.available();
```

Рисунок 4.6 – Метод WiFiClient для слухання клієнтів

Для того щоб веб-сервер зміг зрозуміти що саме необхідно повертати клієнту було реалізовано функцію-парсер, що зображено на рис. 4.7, яка в залежності від введених параметрів в URL стрічці могла зрозуміти яку сторінку потрібно повернути.

```
String parseQuery(String url, WiFiClient client){  
    Serial.println("**parseQuery**");  
    if (url.startsWith("GET / HTTP/1.1")) {  
        homePage(client);  
        Serial.println(url);  
    }  
    if(url.startsWith("GET /submit")){  
        homePage(client);  
        Serial.println(url);  
    }  
    else if(url.startsWith("GET /reset")){  
        Serial.println("RESET");  
        resetFunc();  
    }  
}
```

Рисунок 4.7 – Функція-парсер для розбиття URL на параметри

В даній реалізації для отримання сторінки з конфігуруванням кред потрібно ввести тільки IP-адресу веб-сервера. Після чого клієнт отримає конфігураційну сторінку для введення даних SSID та PASS щоб під'єднатись до Wi-Fi з виходом у глобальну мережу. Після введення даних по натисненню кнопки виконується запит

типу GET по методу “/submit”, який передає далі URL стрічку вигляду “https://192.168.4.1/submit?ssid=xxxx&pass=xxxx” функції, яка збирає параметри з стрічки та зберігає в окремі структури. Структуру зображено на рис. 4.8. Функція, що розбиває та зберігає значення ssid та pass до структури “StructForFile” зображено на рис. 4.9.

```
struct StructForFile {  
    char  ssid[64];  
    char  pass[64];  
    int minlux;  
} structForFile;
```

Рисунок 4.8 – Визначення структури, що зберігає кредити Wi-Fi каналу

```
if (Array1[counter].key == "ssid") {  
    structForFile.ssid[64] = {0};  
    const char* src = Array1[counter].value.c_str();  
    strncpy( (char*)structForFile.ssid, src, strlen(src) );  
    Serial.println(structForFile.ssid);  
}  
if (Array1[counter].key == "pass") {  
    structForFile.pass[64] = {0};  
    const char* src = Array1[counter].value.c_str();  
    strncpy( (char*)structForFile.pass, src, strlen(src) );  
    Serial.print(structForFile.pass);  
}
```

Рисунок 4.9 – Зберігання ssid та pass в структурі “StructForFile”

Було вирішено, що для оптимізації процесу збереження та підключення до Wi-Fi необхідно використовувати вбудовану пам'ять в мікроконтролері. Дане рішення є покращенням, оскільки немає необхідності конфігурувати кредити Wi-Fi кожен раз при відключенні системи від електромережі. Для реалізації було використано можливість SPIFFS. SPIFFS – це зпрощена файлова система, що призначена для мікроконтролерів з чіпом flash, який передає дані по шині SPI. Така файлова система

забезпечує доступ до флешпам'яті приблизно за таким принципом, що й звичайна файлова система в операційній системі на персональному комп'ютері, але простіша та має деякі обмеження.

SPIFFS корисна для використання ESP32 в наступних ситуаціях:

- Створення файлів для збереження налаштувань;
- Постійне збереження даних;
- Створення файлів для збереження невеликої кількості даних, замість використання карти MicroSD;
- Збереження HTML та CSS файлів.

Наприклад, файли HTML та CSS повинні бути збережені в папці data, що має бути всередині директорії із проєктом, що зображено на рис. 4.10.

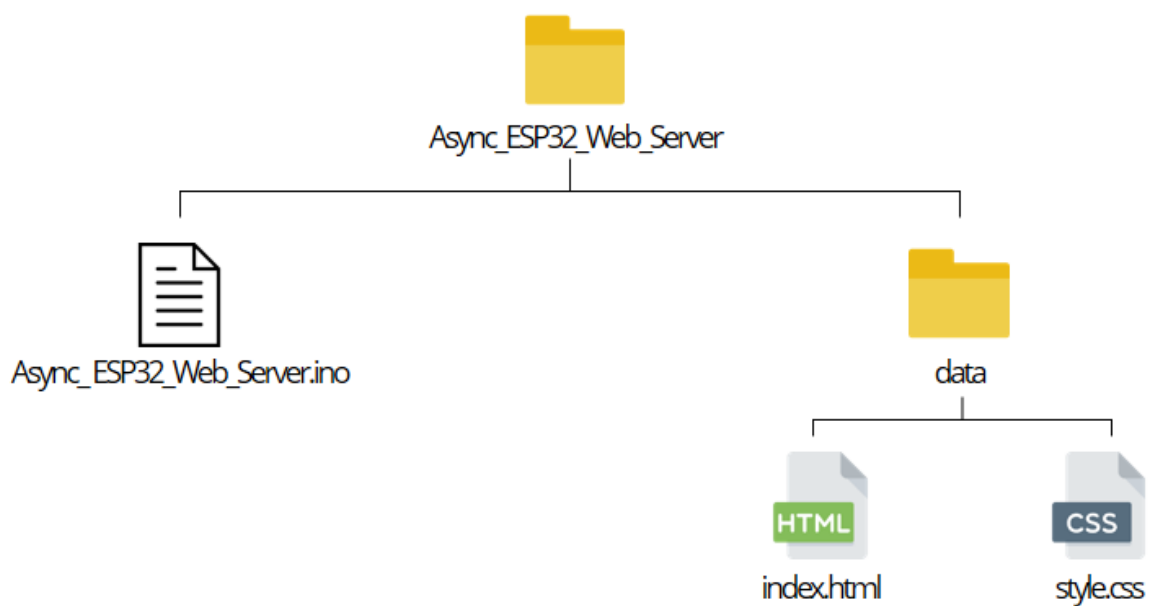


Рисунок 4.10 – Ієрархія директорій проєкту

На рис. 4.11 зображено фрагмент коду, де функція “SPIFFS.begin()” встановлює файлову систему SPIFFS. Вона виконується найпершою до виклику всіх інших функцій для роботи з файловою системою. Метод повертає значення “true”, якщо файлову систему успішно ініціалізовано та “false” якщо при ініціалізації виникли помилки.

```

if(!SPIFFS.begin(true)){
    Serial.println("An Error has occurred while mounting SPIFFS");
    return;
}

writeFile(SPIFFS, "/ssid.txt", structForFile.ssid);
Serial.println();

writeFile(SPIFFS, "/pass.txt", structForFile.pass);
Serial.println();

```

Рисунок 4.11 – Ініціалізація SPIFFS

Для запису даних в файл використовується функція “writeFile”. Реалізацію методу зображено на рис. 4.12.

```

void writeFile(fs::FS &fs, const char * path, const char * message){
    Serial.printf("Writing file: %s\r\n", path);

    File file = fs.open(path, FILE_WRITE);
    if(!file){
        Serial.println("- failed to open file for writing");
        return;
    }

    file.print("");
    if(file.print(message)){
        Serial.println("- file written");
    } else {
        Serial.println("- write failed");
    }

    file.close();
}

```

Рисунок 4.12 – Запис даних у файли

На самому початку ініціалізації компонентів та змінних програми, що виконується в блоці “setup()”, перед пошуком Wi-Fi мережі виконується читання даних ssid та pass із файлів “/ssid.txt” та “/pass.txt” відповідно. Після цього при наявності в файлах цих даних функція з’єднання до Wi-Fi мережі підставляє дані ssid та pass на вхід в аргументи функції. Виклик функції зображено на рис. 4.13.

```

server.begin();
Serial.println("Server started");

if(!SPIFFS.begin(true)){
    Serial.println("An Error has occurred while mounting SPIFFS");
    return;
}

readFile(SPIFFS, "/ssid.txt");
Serial.println();

readFile(SPIFFS, "/pass.txt");
Serial.println();

WiFi.begin(ssid1.c_str(), password1.c_str());

```

Рисунок 4.13 – Функція зчитування даних з файлу

Реалізацію функції зчитування зображено на рис. 4.14.

```

void readFile(fs::FS &fs, const char * path){
    Serial.printf("Reading file: %s\r\n", path);

    File file = fs.open(path);
    if(!file || file.isDirectory()){
        Serial.println("- failed to open file for reading");
        return;
    }
    Serial.println("- read from file:");
    while(file.available()){

        if(path == "/ssid.txt"){
            ssid1 = urldecode(file.readString());
            Serial.println("ssid1 = ");
            Serial.print(ssid1);
        }
        if(path == "/pass.txt"){
            password1 = urldecode(file.readString());
            Serial.println("password1 = ");
            Serial.print(password1);
        }
        Serial.write(file.read());
    }
    file.close();
}

```

Рисунок 4.14 – Реалізація функції зчитування даних з файлу

Першу частину серверного розроблення надано діаграмою послідовностей у додатку Е.

Розглянемо другу частину де безпосередньо відбувається збір даних з віддалених датчиків та контролер працює в режимі повноцінного веб-серверу.

Спочатку під'єднуємо датчики до відповідних пінів контролера та ініціалізуємо їх в коді. На рис. 4.15 зображено з'єднання всіх компонентів системи, а саме: датчик освітленості, датчик руху, датчик газу та датчики температури й вологості. Також до системи було додано два світлодіоди. Перший світлодіод вмикається при недостатньому рівні освітленості (параметри задає користувач) та другий світлодіод вмикається при спрацюванні датчика руху, тобто детектування незаконного чи непередбачуваного вторгнення в приміщення.

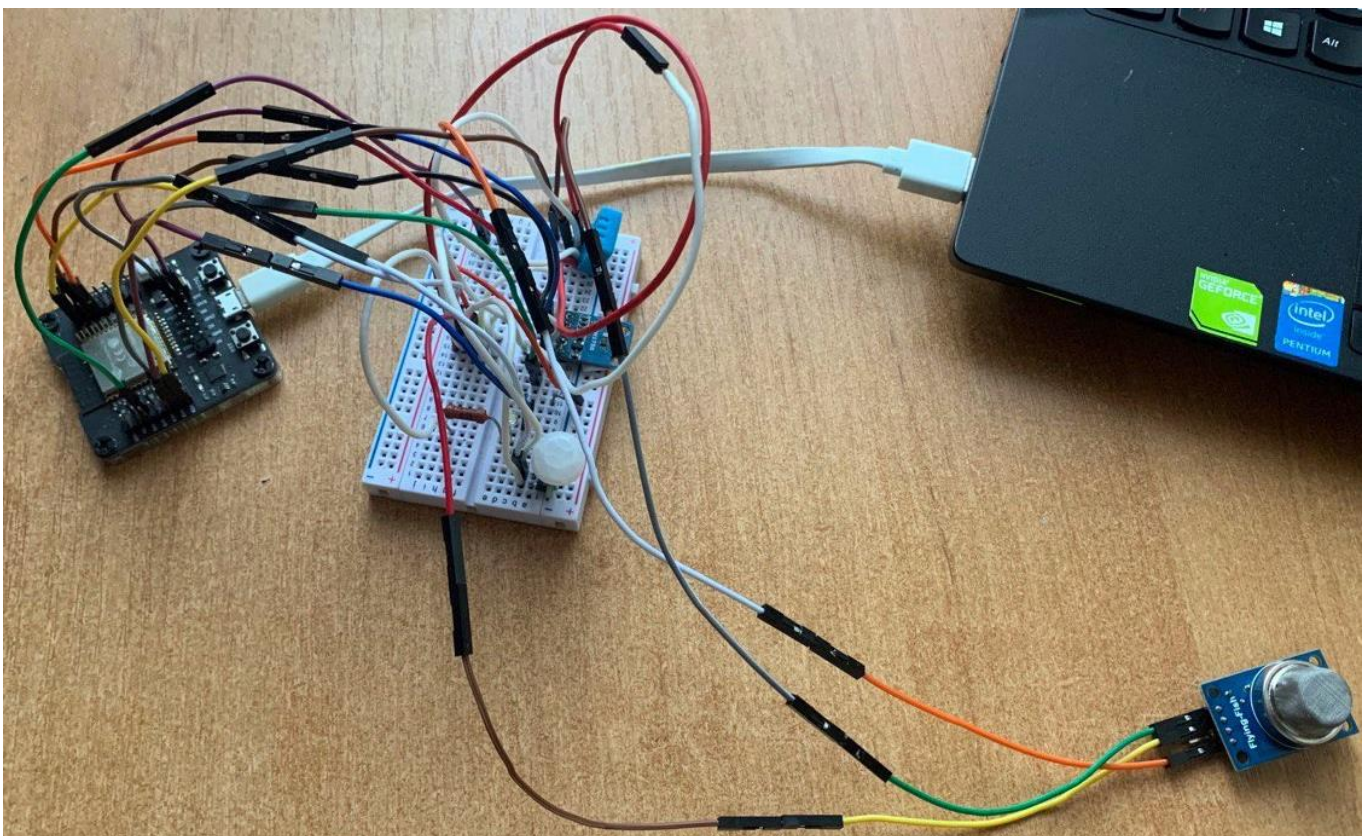


Рисунок 4.15 – Вигляд готової схеми із з'єднаними всіма компонентами

Наступним кроком необхідно підключити датчики в коді програми. Для цього потрібно ініціалізувати датчики в блоці “`setup()`”. Даний блок використовується для підключення змінних, визначення режимів роботи виводів пристроїв, запуску використовуваних бібліотек. Функція “`setup()`” запускається тільки один раз після

подачі живлення, або при перезавантаженні програми на мікроконтролері. Виконання підключення датчиків в коді зображено на рис. 4.16.

Окрім вбудованої функції “setup()” в Arduino IDE середовищі також присутня ще функція “loop” без якої програма не зможе працювати. Після виконання функції “setup()”, програма переходить до циклу “loop”, котрий буде виконуватись до тих пір поки на плату подано живлення.

```
//BH1750
BH1750 lightMeter(0x23);
lightMeter.begin();

//AM312
pinMode(motionSensor, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(motionSensor),
detectsMovement, RISING);
pinMode(ledPin26, OUTPUT);
digitalWrite(ledPin26, LOW);

//MQ-2
pinMode(MQ2PIN, INPUT);
mq2.calibrate();
Serial.println("Calibrated: " + String(mq2.getRo()));

//DHT-11
pinMode(DHTPin, INPUT);
dht.begin();
```

Рисунок – 4.16 Ініціалізація датчиків

На наступному фрагменті коду, що зображено на рис. 4.17 виконується зчитування даних з датчиків руху AM312 та датчика газу MQ-2.

В разі отримання неприйнятного показу, тобто таке значення, що заходить за межу граничного значення в даній реалізації передбачується надсилання користувачу покази в чат-бот через Telegram Bot API.

```

//AM312
if(motionDetected){
    if(motion_on){
        bot.sendMessage(CHAT_ID, "Motion detected!!", "");
        Serial.println("Motion detected");
        motionDetected = false;
        if(startTimer) {
            Serial.println("Motion stopped...");
            digitalWrite(ledPin26, LOW);
            startTimer = false;
        }
    }
}

//MQ-2
if(gas_on){
    if(mq2.readLPG() > 300){
        Serial.println("Gas Detected!!!");
        bot.sendMessage(CHAT_ID, "Gas detected!! Propan-butan is equal " +
            String(mq2.readLPG()) + "ppm", "");
    }
    if(mq2.readMethane() > 300){
        Serial.println("Gas Detected!!!");
        bot.sendMessage(CHAT_ID, "Gas detected!! Methane is equal " +
            String(mq2.readMethane()) + "ppm", "");
    }
    if(mq2.readSmoke() > 300){
        Serial.println("Gas Detected!!!");
        bot.sendMessage(CHAT_ID, "Gas detected!! Smoke is equal " +
            String(mq2.readSmoke()) + "ppm", "");
    }
    if(mq2.readHydrogen() > 300){
        Serial.println("Gas Detected!!!");
        bot.sendMessage(CHAT_ID, "Gas detected!! Hydrogen is equal " +
            String(mq2.readHydrogen()) + "ppm", "");
    }
}
}

```

Рисунок 4.17 – Зчитування показів датчиків

Для початку необхідно підключити бібліотеки із методами Telegram API, що зображено на рис. 4.18, а потім вказати ідентифікатор чату й токен доступу до бота, що зображено на рис. 4.19.

```
//Telegram API
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
```

Рисунок 4.18 – Визначення ідентифікатору чата та токен бота

```
#define BOTtoken "XXXXXXXXXX:XXXXXXXXXXXXXXXXXXXXX"
#define CHAT_ID "-XXXXXXX"
```

Рисунок 4.19 – Визначення ідентифікатору чата та токен бота

Наступним методом, що зображено на рис. 4.20 створюємо бота передаючи йому токен та клієнт.

```
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);
```

Рисунок 4.20 – Створення бота

Для відправлення повідомлення необхідно викликати метод “sendMessage()” та передати йому параметри CHAT\_ID, що відповідає визначеному ідентифікатору чата, в який будуть приходити повідомлення, а також передати необхідно саме повідомлення. Приклад використання метода “sendMessage()” зображено на рис. 4.21.

```
if(mq2.readLPG() > 300){
  Serial.println("Gas Detected!!!");
  bot.sendMessage(CHAT_ID, "Gas detected!! Propan-butan is equal " +
    String(mq2.readLPG()) + "ppm", "");
}
```

Рисунок 4.21 – Відправка повідомлення в чат-бот

Крім відправлення повідомлень в чат-бот при виявленні не нормованих значень показів датчиків також користувач може відправляти команди боту для отримання потрібної інформації про стан системи, або керувати включенням чи виключенням

датчиків. Команди посилаються у такому вигляді “/command”. Відповідно є функція “handleNewMessages()”, що обробляє ці команди. Перевірки на введену користувачем команду зображено на рис. 4.22.

```
if (text == "/led_on") {
    bot.sendMessage(chat_id, "LED state set to ON", "");
    ledState = HIGH;
    digitalWrite(ledPin, ledState);
}
if (text == "/led_off") {
    bot.sendMessage(chat_id, "LED state set to OFF", "");
    ledState = LOW;
    digitalWrite(ledPin, ledState);
}
if (text == "/temperature") {
    bot.sendMessage(chat_id, "The temperature is: " + String(Temperature) + "C", "");
}
if (text == "/humidity") {
    bot.sendMessage(chat_id, "The humidity is: " + String(Humidity) + "%", "");
}
if (text == "/light") {
    bot.sendMessage(chat_id, "The lighting is: " + String(lux) + " lux", "");
}
if (text == "/motion_off") {
    motion_on = false;
    bot.sendMessage(chat_id, "Motion sensor is off \n use /motion_on for on", "");
}
if (text == "/motion_on") {
    motion_on = true;
    bot.sendMessage(chat_id, "Motion sensor is on", "");
}
if (text == "/gas_off") {
    gas_on = false;
    bot.sendMessage(chat_id, "Gas sensor is off \n use /gas_on for on", "");
}
if (text == "/gas_on") {
    gas_on = true;
    bot.sendMessage(chat_id, "Gas sensor is on", "");
}
if (text == "/reset") {
    bot.sendMessage(chat_id, "RESETTING...", "");
    text = "";
    resetFunc();
}
```

Рисунок 4.22 – Обробка запитів користувача

На наступному фрагменті коду, що зображено на рис. 4.23 ініціюється виклик функції “handleNewMessages()”, де на кожній ітерації циклу “loop” відбувається перевірка на наявність нових повідомлень.

```
if (millis() > lastTimeBotRan + botRequestDelay) {  
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);  
  
    while(numNewMessages) {  
        Serial.println("got response");  
        handleNewMessages(numNewMessages);  
        numNewMessages = bot.getUpdates(bot.last_message_received + 1);  
    }  
    lastTimeBotRan = millis();  
}
```

Рисунок 4.23 – Виклик функції “handleNewMessages()”

Останнім етапом розробки є проектування та створення бази даних. Для цього виділили кластер MongoDB, де було створено базу даних esp32db, яка складається з однієї колекції esp32\_data. Завдяки можливостям функціональностей Mongo маємо можливість створити власний сервіс на віддаленому сервері Mongo, де й реалізуємо REST API взаємодії клієнт-сервер. На рис. 4.24 зображено візуалізацію методів, що працюють безпосередньо з базою даних. Структуру бази даних наведено в додатку И.

## </> esp32\_POST

| Incoming Webhooks       | Rules |
|-------------------------|-------|
| Incoming Webhook Name ▲ |       |
| post                    |       |
| get                     |       |
| getLast                 |       |

Рисунок 4.24 – Вхідні вебхуки

Розглянемо більш детально кожний з методів.

На рис. 4.25 зображено реалізацію методу post. В якості параметра payload очікується тіло body, що надходить з клієнтської сторони. Далі відбувається приведення отриманих даних до вигляду JSON та за допомогою функції insertOne() записуємо документ до колекції esp32\_data.

```
exports = function(payload){
  var atlas = context.services.get("mongodb-atlas");
  var coll = atlas.db("esp32db").collection("esp32_data");
  try {
    if (payload.body)
    {
      body = EJSON.parse(payload.body.text());
      console.log(body);
      body['ts'] = new Date();
    }
    return coll.insertOne(body);
  } catch (e) {
    console.log("Error inserting doc: " + e);
    return e.message();
  }
};
```

Рисунок 4.25 – Запис документа до колекції за допомогою методу post

Наступні методи get та getLast, що зображено на рис. 4.26 та 4.27, досить схожі за своєю поведінкою. Метод get використовує функцію find({}) в колекції esp32\_data, що означає пошук всіх існуючих документів в колекції. Метод getLast реалізований для отримання найновішого запису, що може бути корисно користувачу для розуміння поточного стану в даний момент. Відмінність від попереднього методу лише в використанні функції findOne({}, {sort: {\_id: -1}}), що виконує сортування документів за \_id по спаданню та бере перший документ з отриманої вибірки.

```
exports = function(payload) {
  const mongodb = context.services.get("mongodb-atlas");
  const collection = mongodb.db("esp32db").collection("esp32_data");
  return collection.findOne({}, {sort: {_id: -1}});
}
```

Рисунок 4.26 – Запис документа до колекції за допомогою методу post

```
exports = function(payload) {
  const mongodb = context.services.get("mongodb-atlas");
  const collection = mongodb.db("esp32db").collection("esp32_data");
  return collection.find({}).limit(10);
}
```

Рисунок 4.27 – Запис документа до колекції за допомогою методу post

На рис. 4.28 зображено три записи даних з датчиків в колекцію MongoDB. Запис в базу даних відбувається кожні 10 хвилин. Це реалізовано для можливості побудувати діаграму згідно отриманих показів за день. Також база даних є надійне сховище в якому можна переглянути всі записані дані в окремий час. Даний підхід є покращенням, оскільки таким чином не перенавантажується клієнт-серверна частина на стороні Telegram Bot API, а також користувач не отримує велику кількість тривіальних даних в чат. Структура документа складається з таких полів: “\_id” – унікальний ідентифікатор кожного документа, “ts” – час у форматі UTC, “temperature” – значення температури в цельсіях, “humidity” – значення відсотку вологості, “light” – значення рівня освітленості в lux, “gas\_smoke” – значення наявності задимленості в ppm, “gas\_propan” – значення наявності газу пропан в ppm, “gas\_methane” – значення наявності газу метан в ppm, “motion” – значення 0, або 1, що відповідно означає чи зафіксований рух.

```
_id: ObjectId("61ab8fe325623edd714c0808")
  sensors: Object
    gas_propan: 5
    gas_methane: 14
    gas_smoke: 24
    gas_hydrogen: 10
    temperature: 23.9
    humidity: 66
    light: 545.8333
    motion: true
  ts: 2021-12-04T15:57:23.624+00:00
```

Рисунок 4.28 – Перегляд документів в MongoDB

### 4.3 Розроблення клієнтської частини системи

На стороні користувача інтерфейс чат-боту виглядає таким чином як показано на рис. 4.29. А також приклад початкової команди “/start” щоб розпочати спілкування з ботом.

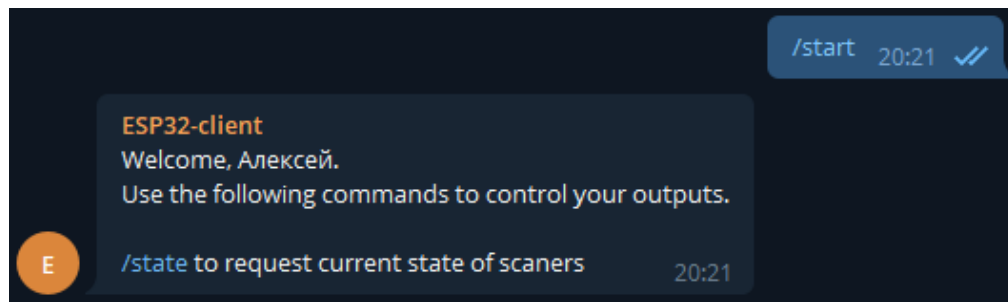


Рисунок 4.29 – Виклик функції “handleNewMessages()”

У відповідь на команду “/start” бот пропонує виконати команду “/state” для визначення стану в якому знаходяться датчики. Якщо надіслати боту цю команду, то у відповідь отримаємо перелік датчиків із поточним станом та з командою на включення або виключення датчика. Перелік станів датчиків зображено на рис. 4.30. Також запити на ввімкнення та вимкнення датчиків зображено на рис. 4.31.

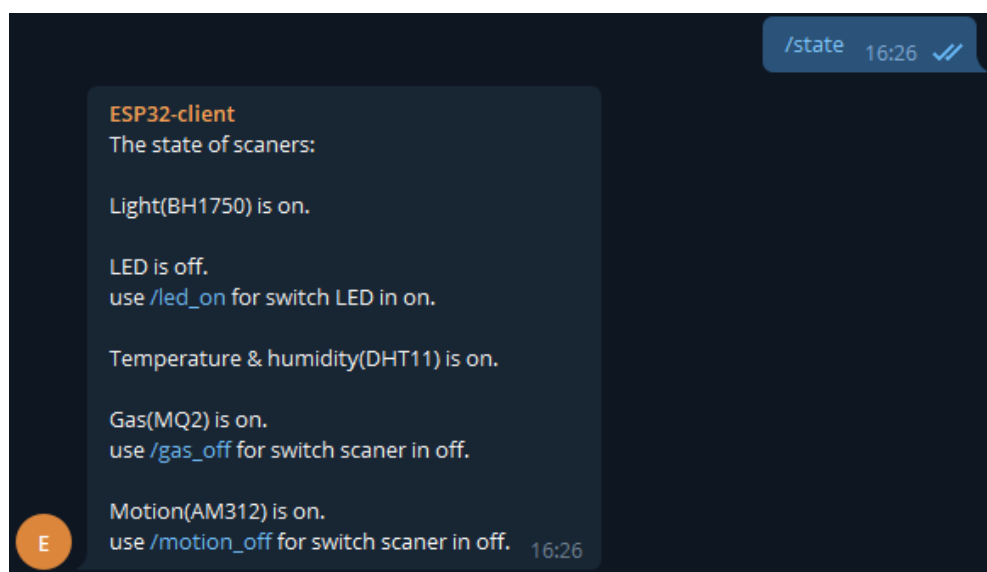


Рисунок 4.30 – Перелік датчиків із поточними станами

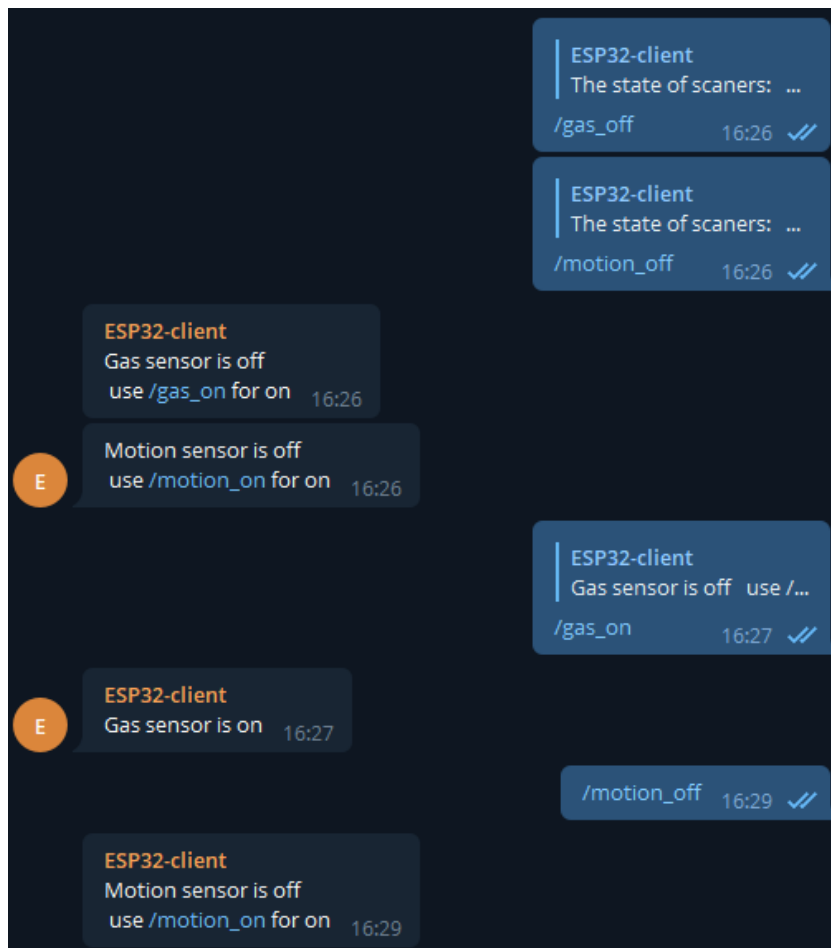


Рисунок 4.31 – Перелік датчиків із поточними станами

Загальний інтерфейс сторінки користувача наведено в додатку Ж. Даний функціонал призначений для налагодження параметрів Wi-Fi через веб-сторінку. Користувачу пропонується ввести два значення SSID та PASS від потрібної мережі. Інтерфейс форми для заповнення зображено на рис. 4.32.

**Admin page:**

☒ Remember

Рисунок 4.32 – Інтерфейс форми для заповнення параметрів

Розмітка сторінки подається у вигляді функції “homepage(client)”, якій на вхід подається клієнт та в середині функції записуються html теги. Відповідно введені значення полів ssid та pass записуються в раніше описану структуру “structForFile”.

Дану функцію зображено на рис. 4.33.

```
void homePage(WiFiClient client){
    client.println("HTTP/1.1 200 OK");
    client.println("Content-type:text/html");
    client.println();
    client.println("<!DOCTYPE html>");
    client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");
    css(client);
    client.println("</head>");
    client.print("<body>");
    client.print("<form action=\"/submit\">");
    client.print("<h1 style=\"margin-bottom: 10px;\">Admin page: </h1>");
    client.print("<input class=\"input \" name=\"ssid\" value=\"\" + String(structForFile.ssid) + \"\" type=\"text\" placeholder=\"SSID\"><br>");
    client.print("<input class=\"input\" name=\"pass\" value=\"\" + String(structForFile.pass) + \"\" type=\"password\" placeholder=\"PASS\"><br>");
    client.print("<label>Set the min value of light:</label>");
    client.print("<input class=\"input\" name=\"minlux\" value=\"\" + minlux + \"\" type=\"text\" placeholder=\"LUX\"><br>");
    client.print("<label><input class=\"checkbox\" name=\"remember\" type=\"checkbox\" checked=\"checked\">Remember</label><br>");
    client.print("<input class=\"inForm\" type=\"submit\" value=\"Send\"><br>");
    client.print("</form>");
    client.print("</body>");
    client.print("</html>");
}
```

Рисунок 4.33 – Реалізація функції “homepage()”

Далі після відправлення даних серверна частина намагається встановити з’єднання із відповідною мережею та відповідає про успішний результат, або ні. Якщо з’єднання успішне, тоді клієнт відображає повідомлення про коректні дані. Також відбувається збереження даних до сховища флешпам’яті, де сервер відповідає true, або false. Клієнт в залежності від значень відображає на сторінці повідомлення про збережені дані. Приклад повідомлень клієнта про успішне та не успішне з’єднання зображено на рис. 4.34 та рис. 4.35.

## Admin page:

The screenshot shows the 'Admin page' with a text input field containing 'home25', a password field with masked characters, a checked 'Remember' checkbox, and a 'Send' button. Below the form, two green-bordered boxes display success messages: 'Connection established. Correct Wi-Fi data.' and 'Entered data successfully saved on flash memory.'

home25

\*\*\*\*\*

☒ Remember

Send

Connection established.  
Correct Wi-Fi data.

Entered data successfully  
saved on flash memory.

Рисунок 4.34 – Повідомлення клієнта про успішне з’єднання з Wi-Fi

## Admin page:

The screenshot shows the 'Admin page' with a text input field containing 'SSID', a password field containing 'PASS', a checked 'Remember' checkbox, and a 'Send' button. Below the form, two boxes display error messages: a red-bordered box with 'Connection unestablished.' and an orange-bordered box with 'Check WiFi parameters!'.

SSID

PASS

☒ Remember

Send

Connection unestablished.

Check WiFi parameters!

Рисунок 4.35 – Повідомлення клієнта про не успішне з’єднання з Wi-Fi

Також для надання веб-формі дизайну було використано CSS стилі. Стили сторінки подаються у вигляді функції “css (client)”, якій на вхід подається клієнт та в

середині функції записуються класи стилів для окремих тегів HTML. Для того, щоб вказані стилі елементів розмітки були визначені необхідно їх помістити в теги `<style></style>`. Для застосування стилів потрібно функцію “css (client)” викликати всередині функції з розміткою, а саме між тегами `<head></head>`. Функцію “css (client)” зображено на рис. 4.36.

```
void css(WiFiClient client){
    client.println("<style>html { font-family: Helvetica; display: inline-block;
    margin: 0px auto; text-align: center;}");
    client.println(".button { background-color: #4CAF50; border: none;
    color: white; padding: 8px 30px;}");
    client.println("text-decoration: none; font-size: 30px; margin: 2px;
    cursor: pointer;}");
    client.println(".button2 {background-color: #555555;}");
    client.println(".red {color: red; border: 1px solid red;}");
    client.println(".orange {color: orange; border: 1px solid orange;}");
    client.println("form {display: flex; flex-direction: column;
    align-items: center;}");
    client.println(".input {height: 20px; margin: 15px; padding: 5px; width: 55%}");
    client.println(".mt-2 {margin-top: 20px;}");
    client.println(".checkbox {display: inline-block;}");
    client.println(".inForm {height: 30px; width: 55%}");
    client.println(".comment {margin: 0 auto; width: 55%;}");
    client.println(".green {color: green; border: 1px solid green;}</style>");
}
```

Рисунок 4.36 – Функція “css (client)” із описаними стилями

### Висновок до розділу

В даному розділі розглянуто загальну схему взаємодії компонентів згідно з якою виконувалось проектування та розроблення системи. Наприклад, компоненти Motion sensor, Smoke sensor, Climate sensor, Light sensor – це незалежні один від одного датчики, що виконують роль збору даних з навколишнього середовища, поки це необхідно. З датчиків зчитує дані Sensor Reader та передає в шину даних Data Area. Особливого значення має файлова система SPIFFS, що зберігає зконфігуровані дані всередині пристрою. Також досить важливий блок комунікація Webserver та Telegram Bot API. Відбувається це таким чином: при перевищенні показів датчика Webserver

передає повідомлення на сервіс Telegram Bot API, де далі знаючи CHAT\_ID та BotToken повідомлення надходить від бота до потрібного чату.

Далі велось розроблення спочатку серверної частини, а потім клієнтської. Розроблення серверної частини ділиться на дві частини. Перша – це коли мікроконтролер ESP32 працює як вхідна точка доступу та друга частина, де відбувається саме збір показів з датчиків та режим роботи контролера як повноцінний веб-сервер. В першому випадку ESP32 створює власну мережу Wi-Fi та діє як розповсюджувач для кількох пристроїв та називається точкою доступу. На відміну від Wi-Fi роутера в нього немає інтерфейса для дротової мережі. Такий режим роботи називається “Soft Access Point”. Також максимальна кількість пристроїв, що можуть підключитись обмежено п’ятьма.

Перед розробкою другої частини спочатку було ініціалізовано датчики в коді, а вже потім реалізовано відправлення повідомлень в чат-бот при виявленні не нормованих значень показів датчиків. Також користувач може відправляти команди боту для отримання потрібної інформації про стан системи, або керувати включенням чи виключенням датчиків.

Після реалізації серверної частини є необхідність в розробці клієнтської сторони. Даний функціонал призначений для налагодження параметрів Wi-Fi через веб-сторінку. Користувачу пропонується ввести два значення SSID та PASS від потрібної мережі. А також було реалізовано запитання даних із сервера через відповідні команди. Наприклад, початкова команда “/start” для того, щоб розпочати спілкування з ботом. У відповідь на команду “/start” бот пропонує виконати команду “/state” для визначення стану в якому знаходяться датчики. Якщо надіслати боту цю команду, то у відповідь отримаємо перелік датчиків із поточним станом та з командою на включення або виключення датчика.

## 5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

Виконаємо маркетинговий аналіз стартап-проєкту щодо збору даних з віддалених датчиків, щоб визначити його стратегії, можливості, технології та напрями реалізації на ринку.

### 5.1 Опис ідеї проєкту

Ключовою ідеєю стартап-проєкту – є розроблення системи автоматизованого збору та оброки даних з датчиків робототехнічного модуля в системі розумний будинок.

Основним фактором даної системи являється автоматизований збір даних в розумному будинку. Дотримуючи цього фактору, система швидко обробляє дані та надсилає користувачеві. Також постійно підтримується зв'язок клієнт-сервер це дає можливість клієнту надсилати команди, які конфігурують датчики або запити про стан системи. Крім цього, система повинна бути швидкодіючою та вчасно реагувати на перевищення граничних значень.

Зараз детальніше розглянемо де можна застосовувати цю ідею та яку вигоду отримає користувач (табл. 5.1).

Таблиця 5.1 – Опис ідеї стартапу

| Зміст ідеї                                                                         | Напрямки застосування                                                                            | Вигода для користувача                                                                                                                                                               |
|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Винайти таку систему, яка автоматизовано збирає, обробляє та надсилає користувачу. | Застосування на підприємствах із сучасним обладнанням і високим рівнем автоматизації виробництва | Користувачі отримують простий у використанні програмний продукт, який віддалено інформує про стан приміщення в реальному часі та зберігає потрібні дані на сервері для подальшого їх |
|                                                                                    | Застосування в приватних будинка чи квартирах                                                    |                                                                                                                                                                                      |

| Зміст ідеї | Напрямки застосування | Вигода для користувача                                                |
|------------|-----------------------|-----------------------------------------------------------------------|
|            |                       | використання. Є можливість реалізовувати рішення за допомогою команд. |

Тепер потрібно визначити основні переваги та недоліки, порівнюючи з аналогічними системами, що існують дотепер. Головними конкуруючими системами є Arduino та Raspberry PI. Далі порівняємо ці дві системи з власною розробкою і визначимо сильні та слабкі сторони кожної з них (табл. 5.2).

Таблиця 5.2 – Визначення сильних, слабких та нейтральних сторін системи

| № | Техніко-економічні характеристики ідеї | Продукція конкурентів |         |              | W<br>(слабка сторона) | N<br>(нейтральна сторона) | S<br>(сильна сторона)                                    |
|---|----------------------------------------|-----------------------|---------|--------------|-----------------------|---------------------------|----------------------------------------------------------|
|   |                                        | Моя система           | Arduino | Raspberry PI |                       |                           |                                                          |
| 1 | Зручність використання                 | +                     | +       | +            |                       | +                         |                                                          |
| 2 | Швидкість передачі даних               | +                     | -       | +            |                       | +                         |                                                          |
| 3 | Об'єм пам'яті на стороні пристрою      | +                     | -       | +            |                       | +                         |                                                          |
| 4 | Відкритість інформації                 | +                     | -       | -            |                       |                           | Задokumentована на відкрита інформація про всі необхідні |

| № | Техніко-економічні характеристики ідеї  | Продукція конкурентів |         |              | W<br>(слабка сторона) | N<br>(нейтральна сторона) | S<br>(сильна сторона)                       |
|---|-----------------------------------------|-----------------------|---------|--------------|-----------------------|---------------------------|---------------------------------------------|
|   |                                         | Моя система           | Arduino | Raspberry PI |                       |                           |                                             |
| 4 | Відкритість інформації                  | +                     | -       | -            |                       |                           | вузли компонентів системи                   |
| 5 | Вартість компонентів                    | +                     | +       | -            |                       | +                         |                                             |
| 6 | Вміст необхідних периферійних пристроїв | +                     | -       | -            |                       |                           | Вміщує в собі технології Bluetooth та Wi-Fi |

Виходячи із вище проведеного аналізу ми робимо висновок, що власна розроблена ідея має більше сильних сторін та переваг серед систем Arduino та Raspberry PI.

## 5.2 Технічний аудит проєкту

Тепер для реалізації цієї ідеї стартап-проєкту потрібно провести аналіз альтернативних технологій. Так як будь-який веб-додаток створюється за допомогою таких мов програмування, як HTML та CSS, або їх нащадків. Тож необхідно зробити аналіз технологій, що є важливими для створення серверної частини (табл. 5.3).

Таблиця 5.3 – Технології реалізації ідеї стартап-проєкту

| № | Ідея проєкту                                                        | Технології її реалізації                   | Наявність технологій | Доступність технологій |
|---|---------------------------------------------------------------------|--------------------------------------------|----------------------|------------------------|
| 1 | Розроблення системи автоматизованого збору та оброки.               | Використовування мови програмування C++    | Наявна               | Доступна               |
|   | даних з датчиків робототехнічного модуля в системі розумний будинок | Використовування мови програмування Java   |                      |                        |
| 2 |                                                                     | Використовування мови програмування Python |                      |                        |

Отже, провівши аналіз альтернативних технологій, можна сказати, що технологічна реалізація продукту можлива при використанні декількох різних технологій.

### 5.3 Аналіз ринкових можливостей запуску стартап-проєкту

Проведемо аналіз впровадження та виведення даного стартап-проєкту на ринок. Таким чином, визначимо загрози та можливості в процесі запуску ідеї (табл. 5.4).

Таблиця 5.4 – Попередня характеристика потенційного ринку

| № п/п | Показники стану ринку             | Характеристика                             |
|-------|-----------------------------------|--------------------------------------------|
| 1     | Наявність конкурентних виробників | 2 (виробники мікроконтролерів та датчиків) |
| 2     | Загальний обсяг продаж, грн/ум.од | ~150 000 грн/ум.од                         |

| №<br>п/п | Показники стану ринку                               | Характеристика                                           |
|----------|-----------------------------------------------------|----------------------------------------------------------|
| 3        | Оцінка ринку                                        | Зростає                                                  |
| 4        | Перешкоди для входження на ринок                    | Наявність смартфону та месенджеру Telegram у користувача |
| 5        | Специфічні вимоги до стандартизації та сертифікації | Ліцензія від засобів охоронного призначення              |
| 6        | Середня норма рентабельності в галузі, %            | 100%                                                     |

Провівши попереднє оцінювання потенційного ринку для даного стартап-проєкту можна сказати, що він є привабливим для нашого продукту. По таблиці 5.4 видно, що достатня кількість головних гравців, мінімальна кількість конкурентів, а також, прослідковується динаміка та рентабельність на ринку.

Не менш важливим елементом маркетингово аналізу є визначення цільової аудиторії. Тож визначимо цільову аудиторію стартап-проєкту (табл. 5.5).

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проєкту

| №<br>п/п | Потреба, що формує ринок          | Цільова аудиторія (цільові сегменти ринку)  | Відмінності у поведінці різних потенційних цільових груп клієнтів | Вимоги споживачів до товару                  |
|----------|-----------------------------------|---------------------------------------------|-------------------------------------------------------------------|----------------------------------------------|
| 1        | Попит у зручному способі контролю | Власники квартир та приватних будинків – це | Користувачі хочуть спостерігати за приміщенням в                  | Вимогами споживачів є точність, швидкість та |

| №<br>п/п | Потреба, що<br>формує ринок                                     | Цільова<br>аудиторія<br>(цільові<br>сегменти ринку)                                                                                                                        | Відмінності у<br>поведінці різних<br>потенційних<br>цільових груп<br>клієнтів | Вимоги<br>споживачів до<br>товару                                                                                                                 |
|----------|-----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|          | приміщенням за<br>допомогою<br>автоматизованої<br>системі даних | головна цільова<br>аудиторія. Крім<br>цього, цією<br>системою<br>можуть<br>користуватися<br>підприємства із<br>якісним,<br>сучасним та<br>автоматизованим<br>виробництвом. | будь-який час та<br>мати змогу<br>керувати ним<br>автономно.                  | стабільність<br>системи,<br>зручність<br>використання<br>боту, висока<br>керованість<br>самостійне<br>відновлення<br>даних системи<br>в разі збою |

Провівши аналіз цільової аудиторії клієнтів ми визначили їх зацікавленість, вимоги до товару та відмінності в їх поведінці. Як висновок можна сказати, що потенційними клієнтами можуть бути як власники квартир або будинків, так і підприємства.

Тепер проаналізуємо ринкове середовище, а саме: фактори, що сприяють ринковому впровадженню проєкту (табл. 5.6), та факторів, що перешкоджають (табл. 5.7).

Таблиця 5.6 – Фактори можливостей

| №<br>п/п | Фактор                                       | Зміст можливості                                                          | Можлива реакція компанії                                                              |
|----------|----------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 1        | Використання іноземних ринків збуту          | Розміщення товару за кордоном                                             | Зміна локалізація продукту на мову країни-замовника та розроблення рекламної кампанії |
| 2        | Зворотній зв'язок з потенційним користувачем | Кастомізація функціоналу в залежності від нових клієнтів або користувачів | Відповідна персональна зміна та додавання нових функціоналів до програмного продукту  |

Таблиця 5.7 – Фактори загроз

| №<br>п/п | Фактор                  | Зміст загрози                                                                          | Можлива реакція компанії                                     |
|----------|-------------------------|----------------------------------------------------------------------------------------|--------------------------------------------------------------|
| 1        | Конкурентність на ринку | Після виходу продукту на вітчизняний чи іноземний ринки з'являються конкуренти         | Нові інновації впровадження задля приваблення нових клієнтів |
| 2        | Поява перешкод          | У приміщенні може з'явитися перешкода, яка обмежить рух платформи                      | Розроблення алгоритму, що допоможе уникнути перешкоди        |
| 3        | Кібератака              | Проникнення до серверів боту, або їх злом, що може призвести до зупинки роботи системи | Завчасний аудит інформаційної безпеки                        |

Наступним кроком необхідно виконати ступеневий аналіз конкуренції (табл. 5.8).

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

| Особливості конкурентного середовища                    | В чому проявляється дана характеристика                             | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
|---------------------------------------------------------|---------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| 1. Вид конкуренції<br>– чиста                           | Продавець має такий товар, в якого нема конкурента.                 | Пропонує нові та по бажанню додаткові функції, що є більш оптимізованими та точними    |
| 2. За рівнем конкурентної боротьби<br>– міжнаціональний | Товарне виробництво орієнтований на збут за кордон                  | Локалізація продукту<br><br>Додаток може бути перекладено на найбільш вживані мови     |
| 3. За галузевою ознакою<br>– внутрішньогалузева         | Систему можна використовувати для збору даних з віддалених датчиків | Допомагає швидше отримувати інформацію                                                 |
| 4. Конкуренція за видами товарів:<br>– товарно-видова   | Головна функція – автоматизований збір даних                        | Можливість прискорити обробку даних                                                    |
| 5. За характером конкурентних переваг<br>– нецінова     | Система може взаємодіяти з іншими платформами                       | Можливість інтеграції бота до інших систем збору даних                                 |

|                                      |                                                                                                              |                                                                                        |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Особливості конкурентного середовища | В чому проявляється дана характеристика                                                                      | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
| 6. За інтенсивністю – не марочна     | Існуючі методи збору та обробки даних нікому не належать (певна сукупність функцій може йти у одному пакеті) | Розроблення такого бренду, що стане популярним у всьому світі                          |

Наступним кроком буде проведений аналіз конкурентності за М. Портером (табл. 5.9).

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

|                  |                                               |                                                                                                                               |                                                                                                  |                                                                            |                          |
|------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|--------------------------|
| Складові аналізу | Прямі конкуренти в галузі                     | Потенційні конкуренти                                                                                                         | Постачальники                                                                                    | Клієнти                                                                    | Товари-замінники         |
|                  | Виробники промислових систем розумний будинок | Певною перешкодою входження в ринок є неготовність клієнтів не тільки купувати, але й користуватися системою розумний будинок | Постачальники мають вміти налаштовувати бота під кожного користувача та супроводжувати платформу | Користувачі повинні вміти використувати функцію виклику технічного сапорту | Системи розумний будинок |

|           | Прямі конкуренти<br>в галузі                                                                                                                                                               | Потенційні<br>конкуренти                                                                                                                    | Постачальни<br>ки | Клієнти     | Товари-<br>замінники                    |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-------------|-----------------------------------------|
| Висновки: | Інтенсивність не є великою, оскільки на ринку досить мало конкурентів, які виготовляють систему розумний будинок, крім того всі вони використовують різні підходи до виготовлення продукту | Можливість виходу існує: пристрій використовує принципово новий підхід<br><br>Строки виходу на ринок: одразу після релізу готового продукту | Не диктують       | Не диктують | Пасивність споживачі до роботи пристрою |

Отже, після аналізу вище зазначеної інформації, можна сказати, що дана система є конкурентною. На ринку існують деякі прямі конкуренти, але вони мають достатньо недоліків. Ми дійшли до спільного рішення, що нашу систему слід розробити, врахувавши недоліки наявних систем та побажання користувача. Крім цього, ми провели ступеневий аналіз конкуренції та описали її характеристики й вплив на діяльність компанії.

Далі потрібно обґрунтувати фактори конкурентоспроможності стартап-проєкту (табл. 5.10). А також провести порівняльний аналіз сильних та слабких сторін системи (табл. 5.11).

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

| №<br>п/п | Фактор конкурентоспроможності | Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)                                                                                     |
|----------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | Функціональність              | Конкуренти не мають можливості передавати дані на великі відстані та віддалено конфігурувати налаштування системи, а також велика кількість команд для управління системою користувачем |
| 2        | Швидкість передачі даних      | Швидкість передачі даних є одним з основних факторів конкурентоспроможності, для даної тематики він є особливим тому що система швидко інформує про можливу небезпеку                   |
| 3        | Стабільність                  | Розроблення такого пристрою, що матиме змогу працювати без збоїв, а також самостійне відновлення системи у випадку знеструмлення електро-мережі                                         |
| 4        | Вартість компонентів системи  | Затратність на компоненти є меншою ніж у конкурентів, а отже в результаті отримаємо продукт по нижчій ринковій ціні                                                                     |

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін системи

| №<br>п/<br>п | Фактор конкурентоспроможності | Бали<br>1-20 | Рейтинг товарів-конкурентів у порівнянні з ... (назва підприємства) |    |    |   |    |    |    |
|--------------|-------------------------------|--------------|---------------------------------------------------------------------|----|----|---|----|----|----|
|              |                               |              | -3                                                                  | -2 | -1 | 0 | +1 | +2 | +3 |
| 1            | Функціональність              | 10           |                                                                     | *  |    |   |    |    |    |
| 2            | Швидкість передачі даних      | 17           | *                                                                   |    |    |   |    |    |    |

| №<br>п/<br>п | Фактор конкурентоспроможності | Бали<br>1-20 | Рейтинг товарів-конкурентів у<br>порівнянні з ... (назва<br>підприємства) |  |  |   |  |  |  |
|--------------|-------------------------------|--------------|---------------------------------------------------------------------------|--|--|---|--|--|--|
| 3            | Стабільність                  | 14           |                                                                           |  |  | * |  |  |  |
| 4            | Вартість компонентів системи  | 12           | *                                                                         |  |  |   |  |  |  |

Після аналізу конкурентоспроможності розглянемо SWOT аналіз (табл. 5.12).

Таблиця 5.12 – SWOT-аналіз стартап-проєкту

|                                                                                                                                                                                                                                                                               |                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Сильні сторони:</b></p> <ul style="list-style-type: none"> <li>– Віддалене керування через месенджер телеграм за допомогою сервісу Telegram Bot API</li> <li>– Висока швидкість передавання даних</li> <li>– Збереження даних до вбудованої пам'яті пристроя</li> </ul> | <p><b>Слабкі сторони:</b></p> <ul style="list-style-type: none"> <li>– Наявність конкурентів на ринку</li> <li>– Користувачі мають недовіру до нових продуктів</li> </ul> |
| <p><b>Можливості:</b></p> <ul style="list-style-type: none"> <li>– Кастомізація продукту під побажання клієнта</li> <li>– Масштабування</li> </ul>                                                                                                                            | <p><b>Загрози:</b></p> <ul style="list-style-type: none"> <li>– Складний процес інтеграції з продуктом</li> <li>– Цінова конкуренція</li> </ul>                           |

Після отриманого SWOT-аналізу потрібно створити альтернативи ринкової поведінки для того, щоб забезпечити запуск стартап-проєкту та визначити оптимальний час його виходу на ринок (табл. 5.13).

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

| №<br>п/п | Альтернатива<br>(орієнтовний комплекс<br>заходів) ринкової<br>поведінки                               | Ймовірність отримання<br>ресурсів | Строки реалізації                              |
|----------|-------------------------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------|
| 1        | Кастомізація продукту<br>під побажання клієнта<br>та особливості<br>юрисдикції клієнтської<br>сторони | Висока ймовірність<br>(80%)       | Середні строки<br>реалізації<br><br>(3 місяці) |
| 2        | Ефективна реклама<br>продукту, що<br>зацікавить клієнтів                                              | Середня ймовірність<br>(45%)      | Середні строки<br>реалізації<br><br>(2 місяці) |

За результатами проведеного аналізу можна зробити висновок, що необхідно презентувати системи на відповідних публічних заходах для ринкового впровадження стартап-проекту та виявити потенційні канали збуту.

#### 5.4 Розроблення ринкової стратегії стартап-проекту

Для визначення ринкової стратегії необхідно охопити весь ринок, зокрема аналіз потенційних клієнтів (табл. 5.14).

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

| № п/п                                                     | Опис профілю цільової групи клієнтів                                                | Готовність споживачів сприйняти продукт                                                                                            | Орієнтовний попит в межах цільової групи                                           | Інтенсивність конкуренції в сегменті                    | Простота входу у сегмент                                                            |
|-----------------------------------------------------------|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|---------------------------------------------------------|-------------------------------------------------------------------------------------|
| 1                                                         | Використання на підприємствах із якісним, сучасним та автоматизованим виробництвом. | Продукт є компромісним варіантом попередніх рішень збору даних в розумних будинках, тому готовність прийняття повинна бути високою | Враховавши недоліки попередніх рішень попит на продукт повинен бути досить високий | Так як стартап має нові функції, поки конкуренції немає | Власники сучасних підприємств та приватних осель готові використовувати мою систему |
| 2                                                         | Використання у власних будинках чи квартирах                                        |                                                                                                                                    |                                                                                    |                                                         |                                                                                     |
| Цільова аудиторія – підприємства, фірми та власники осель |                                                                                     |                                                                                                                                    |                                                                                    |                                                         |                                                                                     |

Отже, після аналізу цільових груп потенційних споживачів мого проєкту було визначено, що в ролі цільової групи є підприємства із якісним, сучасним та автоматизованим виробництвом та власники будинків, або квартир. Тому ми обрали стратегію масового маркетингу. Далі розглянемо визначення базової стратегії розвитку (табл. 5.15) та конкурентної поведінки (табл. 5.16).

Таблиця 5.15 – Визначення базової стратегії розвитку

| № п/п | Обрана альтернатива розвитку проєкту  | Стратегія охоплення ринку                                 | Ключові конкурентоспроможні позиції                             | Базова стратегія розвитку* |
|-------|---------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------------|----------------------------|
|       |                                       |                                                           | відповідно до обраної альтернативи                              | *                          |
| 1     | Ринок: підприємства та власники осель | Пропозиція доступної системи моніторингу стану приміщення | Оптимальне співвідношення роботи алгоритмів і функціональностей | Стратегія диференціації    |

Проаналізувавши стратегії охоплення ринку, альтернативу проєкту та її конкурентноспроможні позиції, можна зробити висновок, що базова стратегія розвитку для нашого проєкту – стратегія диференціації (за М.Портером). Реалізація даної стратегії є найбільш затратною. Так як ринок може прийняти нашу ціну і дозволяє компанії стати рентабельною, то потрібна успішна політика диференції.

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

| № п/п | Чи є проєкт «першопрохідцем» на ринку? | Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?  | Чи буде компанія копіювати основні характеристики товару конкурента, і які?                                                     | Стратегія конкурентної поведінки * |
|-------|----------------------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------|
| 1     | Так                                    | Мій продукт буде мати лише нових споживачів, працюючи паралельно з конкурентами | Можливо копіюватись буде тільки платформа (оскільки ми не розробляємо сам апаратний пристрій) на базі якої реалізована система. | Стратегія Лідера                   |

Отже, в ході аналізу наступних показників: чи є проєкт «першопрохідцем» на ринку, чи доведеться компанії шукати нових покупців та чи буде компанія копіювати основні характеристики товару конкурента було обрано стратегію лідера. Дана стратегія підходить для реалізації продукту на початкових стадіях розвитку товару, коли попит вже починає зростати, а тиск конкурентів є невеликий, або його ще немає.

На основі вимог споживачів до продукту та постачальника або стартап-компанії, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробляється стратегія позиціонування (табл. 5.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), з яким споживачі мають ідентифікувати торгівельну марку.

Таблиця 5.17 – Визначення стратегії позиціонування

| № п/п | Вимоги до товару цільової аудиторії                                       | Базова стратегія розвитку | Ключові конкурентоспроможні позиції власного стартап-проєкту                                             | Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту (три ключових)    |
|-------|---------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| 1     | Функціональність, відновлюваність при збоях, точність та швидкість роботи | Стратегія диференціації   | Оптимальне співвідношення роботи алгоритмів і функціональностей та зручний інтерфейс зворотнього зв'язку | Висока швидкість передачі даних<br>Зручність використання<br>Супроводження товару (support) |

Після проведення аналізу базової стратегії розвитку, конкурентної поведінки та позиціонування вирішено обрати – стратегію диференціації та стратегію заняття конкурентної ніші. В результаті, ми маємо отримати готову систему рішень ринкової поведінки та напрями роботи стартапу на ринку.

## 5.5 Розроблення маркетингової програми стартап-проєкту

Задля розробки маркетингової програми стартап-проєкту потрібно визначити переваги потенційного товару, описати рівні моделі товару. Крім цього, нам слід встановити вартість, сформувавши систему збуту та визначити маркетингову комунікацію (табл. 5.18).

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

| № п/п | Потреба                                                        | Вигода, яку пропонує товар           | Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)      |
|-------|----------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------|
| 1     | Нова система збору та обробки даних в системі розумний будинок | Швидкість, надійність, оперативність | Зручний інтерфейс взаємозв'язку з користувачем, швидкість обробки, доступна ціна. |

Наступним кроком розробляється трирівнева маркетингова модель товару: уточнюється сутність та складові продукту, його властивості, якість та товарне підкріплення (табл. 5.19).

Таблиця 5.19 – Опис трьох рівнів моделі товару

| Рівні товару        | Сутність та складові                                                                                                                                                                                                                                                                                                                                |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| I. Товар за задумом | Система збору даних повинна відправляти та отримувати пакети даних зі швидкістю інтернету не більше ніж 150Мб/с, мати не більше 4 Мб пам'яті в флеш-накопичувачу файлової системи, циклічне зчитування показів датчиків кожні 2с, застосування налаштувань датчиків не пізніше 2с після відправлення команд, датчики заздалегідь відкалібровані для |

|                                                                                                                                                |                                                                                                                                                                                                                          |      |                |
|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|----------------|
| Рівні товару                                                                                                                                   | Сутність та складові                                                                                                                                                                                                     |      |                |
|                                                                                                                                                | кожної одиниці системи, система у змозі самостійно відновити підключення до мережі передачі даних у разі збоїв в електромережі.                                                                                          |      |                |
| II. Товар у реальному виконанні                                                                                                                | Властивості/характеристики                                                                                                                                                                                               | М/Нм | Вр/Тх /Тл/Е/Ор |
|                                                                                                                                                | 1. Швидкість оптимізації                                                                                                                                                                                                 | М    | Тх             |
|                                                                                                                                                | 2. Об’єм даних, що зберігається                                                                                                                                                                                          | М    | Тх             |
|                                                                                                                                                | 3. Керованість                                                                                                                                                                                                           | М    | Тх             |
|                                                                                                                                                | Якість: Висока якість збору та обробки даних                                                                                                                                                                             |      |                |
|                                                                                                                                                | Пакування: набір з'єднаних датчиків до платформи, кількість яких може змінюватись від потреб клієнта, доступ до хмарного сховища де знаходиться вихідний код проєкту                                                     |      |                |
|                                                                                                                                                | Марка: SHS – Smart Home System                                                                                                                                                                                           |      |                |
| III. Товар із підкріпленням                                                                                                                    | Наявність гарантійного талону (дійсний від 1-3 років).<br>Супроводжується технічна підтримка товару, яка відповість на всі питання пов’язані із роботою системи, а також можливість діагностувати та виправити проблему. |      |                |
| За рахунок чого потенційний товар буде захищено від копіювання: завдяки сучасним ідеям оптимізації алгоритмів збору, обробки та передачі даних |                                                                                                                                                                                                                          |      |                |

Далі ми визначимо цінові межі, якими будемо керуватись для встановлення ціни на стартап-проект. Для цього потрібно проаналізувати ціни на аналогічні товари та оцінити рівень доходів потенційних клієнтів (табл. 5.20).

Таблиця 5.20 – Визначення меж встановлення вартості

| №<br>п/п | Рівень цін на<br>товари-<br>замінники                                                                                                                                  | Рівень цін на<br>товари-<br>аналоги                                                                      | Рівень доходів<br>цільової групи<br>споживачів | Верхня та нижня межі<br>встановлення ціни на<br>товар/послугу                                                                                         |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | Системи<br>зчитування<br>даних в яку<br>входять:<br>мікроконтролер,<br>набір датчиків,<br>відладочна<br>плата в<br>середньому<br>кошують від<br>800 грн до 4000<br>грн | Товари-<br>аналоги<br>існують, але<br>мають<br>застарівшу<br>реалізацію та<br>недосконалий<br>функціонал | Вільно високий                                 | Продаж продукту:<br>150\$<br>Гарантійний талон:<br>від 20\$<br>Технічна підтримка:<br>20\$ за виклик, у<br>випадку відсутнього<br>гарантійного талона |

Передостаннім етапом є формування оптимальної системи збуту, в межах якого приймається рішення (табл. 5.21).

Таблиця 5.21 – Формування системи збуту

| №<br>п/п | Специфіка<br>закупівельної<br>поведінки цільових<br>клієнтів             | Функції збуту, які<br>має виконувати<br>постачальник товару   | Глибина<br>каналу збуту                       | Оптимальна<br>система збуту                        |
|----------|--------------------------------------------------------------------------|---------------------------------------------------------------|-----------------------------------------------|----------------------------------------------------|
| 1        | Клієнти<br>(підприємства та<br>власники осель), що<br>прагнуть підвищити | Функція Billing<br><br>Функція refund<br><br>Функція delivery | Глибина 1:<br><br>для окремих<br>користувачів | Сервісна<br>модель збуту<br><br>без<br>альтернатив |

| №<br>п/п | Специфіка<br>закупівельної<br>поведінки цільових<br>клієнтів | Функції збуту, які<br>має виконувати<br>постачальник товару | Глибина<br>каналу збуту | Оптимальна<br>система збуту |
|----------|--------------------------------------------------------------|-------------------------------------------------------------|-------------------------|-----------------------------|
| 1        | рівень управління та<br>контролю стану<br>приміщення         | Функція settings                                            |                         |                             |

Останнім етапом маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 5.22).

Таблиця 5.22 – Концепція маркетингових комунікацій

| №<br>п/п | Специфік<br>а<br>поведінки<br>цільових<br>клієнтів | Канали<br>комунікацій,<br>якими<br>користуються<br>цільові<br>клієнти | Ключові<br>позиції,<br>обрані для<br>позиціонуванн<br>я | Завдання<br>рекламного<br>повідомлення                               | Концепція<br>рекламного<br>звернення                                                                     |
|----------|----------------------------------------------------|-----------------------------------------------------------------------|---------------------------------------------------------|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| 1        | Коливаль<br>ні<br>продажі                          | Технічна<br>підтримка,<br>колцентр,<br>персональний<br>кабінет        | Функціональн<br>ість, сучасне<br>рішення                | Зацікавити та<br>привернути<br>увагу клієнта<br>до нашого<br>проєкту | With the smart<br>home system<br>you can move<br>into the home of<br>the future and<br>comfort lifestyle |

Результатом даного пункту є маркетингова програма, яка складається з концепції товару, збуту, просування та попередній аналіз можливостей утворення ціни. Також аналіз ринку спирається на цінності та потреби потенційних клієнтів, їх

конкурентні переваги, стан та динаміку ринкового простору. Отже, проаналізувавши концепцій та складові майбутнього товару, ми обрали відповідну ринкову стратегію та поведінку.

### Висновок до розділу

В даному розділі було описано ідею самого проєкту, проведено технічний аудит, проаналізовано ринкові можливості запуску стартап-проєкту. В результаті опису стартап-проєкту ми визначили, що ключовою ідеєю даного стартап-проєкту – є розроблення системи автоматизованого збору та обробки даних з датчиків робототехнічного модуля в системі розумний будинок.

Основним фактором даної системи являється автоматизований збір даних в розумному будинку. Дотримуючи цього фактору, система повинна швидко обробляти дані та відправляти користувачу. Також постійно підтримується зв'язок клієнт-сервер, що надає можливість клієнту надсилати команди, або запити про стан системи. Крім цього, система повинна бути швидкодіючою та вчасно реагувати на перевищення граничних значень.

Далі провівши аналіз альтернативних технологій, можна сказати, що технологічна реалізація продукту можлива при використанні декількох різних технологій. Для даної ідеї стартап-проєкту обрано технологію №1 з використанням та застосуванням мови програмування C++.

Після аналізу ринкових можливостей запуску стартап-проєкту, можна сказати, що дана система є конкурентною, так як на ринку існують деякі прямі конкуренти. Ми дійшли до рішення, що нашу систему слід розробити, врахувавши недоліки наявних систем та побажання користувача. Крім цього, ми провели ступеневий аналіз конкуренції та описали її характеристики й вплив на діяльність компанії.

Останнім кроком було розроблення маркетингової програми, яка складається з концепції товару, збуту, просування та попереднього аналізу утворення ціни. Отже, проаналізувавши концепцій та складові майбутнього товару, ми обрали відповідну ринкову стратегію та поведінку.

## ВИСНОВКИ

В ході роботи над магістерською дисертацією було розроблено ефективний збір даних з датчиків, обробка та швидка передача даних користувачу в системі розумний будинок. Предметом даного дослідження були методи, технології та засоби для побудови апаратної та програмної частин з яких складається система розумний будинок. Також в ході розроблення даної роботи були виконані наступні завдання: розроблено аналіз та опис предметної області; було обрано компоненти й технології для розробки системи; розроблено програмне забезпечення та проведено маркетинговий аналіз стартап-проєкту.

Першим етапом в даній роботі було проведено аналіз по вибору компонентної бази та програмного забезпечення системи. Апаратна частина системи була розроблена на основі мікроконтролера ESP32. До апаратної частини також входять датчики освітленості, температури й вологості, а також датчик газів та датчик руху. Дана система має можливість віддалено інформувати користувача про стан приміщення, а також віддаленно налаштувати конфігурацію системи. Унікальною ознакою автоматизованої системи є функціонал, що надає можливість користувачу віддалено конфігурувати кредити підключення до Wi-Fi мережі, якою потім і буде виконано відправлення даних користувачу, а також можливість зберігати конфіденційні дані до флешпам'яті пристрою. Далі усі зібрані дані система надсилає на віддалений вебсервер Telegram, де згодом користувач зможе дізнаватись про стан приміщення шляхом відправки метрик в телеграм-чат.

В першому розділі було розглянуто структури підключення IoT-пристроїв у мережі та з'ясовано основні функціональні можливості і аналоги систем збору даних з віддалених датчиків. Врахувавши переваги та недоліки переглянутих систем було визначено мету, об'єкт та предмет дослідження. Було визначено основні завдання, які необхідно виконати для досягнення поставлених цілей.

В наступному розділі було розглянуто апаратні компоненти модулів системи. Було визначено такі функціональні модулі даної системи: Головний модуль "Home center", модуль "Light sensor", модуль "Motion sensor", модуль "Smoke sensor" та

модуль “Climate sensor”. В реалізації даної системи передбачається використання єдиного керуючого модуля «Home center» для обробки даних отриманих з датчиків та управління самими датчиками. Призначенням інших модулів є збір даних з датчиків та своєчасна реакція на перевищенні рівні безпечних показів, які регулюються користувачем. В якості платформи на якій буде базуватись головний модуль системи було визначено, що найкраще підходить мікроконтролер ESP32. Це досить сучасний мікроконтролер в якому наявні периферійні інтерфейси та протоколи безпроводного зв'язку, а також досить низька ціна за одиницю такого модуля. Тому саме через підходящі характеристики контролера та його достатню обчислювальну потужність під поставлену задачу було вирішено обрати даний мікроконтролер.

В третьому розділі ми розглянули технології згідно з якими буде розроблено систему автоматизованого збору та обробки даних з датчиків робототехнічного модуля в системі розумний будинок. Було з'ясовано, що на мові C++ ми будемо програмувати апаратне забезпечення в середі розробки Arduino IDE. Було проаналізовано, що досить зручно отримувати повідомлення про стан датчиків системи та керувати датчиками через чат-ботів. Керувати чат-ботом можна тільки по заздалегідь підключеній системі до Wi-Fi. Тому необхідно розробити таку сторінку з можливістю налаштовувати параметри підключення до мережі Wi-Fi. Для розробки такого функціоналу будуть використовуватись технології HTML, CSS та Javascript.

На етапі реалізації системи було визначено загальну схему взаємодії компонентів згідно з якою виконувалось проектування та розроблення системи. Наприклад, компоненти Motion sensor, Smoke sensor, Climate sensor, Light sensor – це незалежні один від одного датчики, що виконують роль збору даних з навколишнього середовища. Особливого значення має файлова система SPIFFS, що зберігає зконфігуровані дані всередині пристроя. Також розглянули досить важливий блок – комунікація вебсерверу ініціалізованого на стороні ESP32 та сервісу Telegram Bot API. Порівняли реляційні та нереляційні бази даних. З'ясували переваги та недоліки відповідно MySQL та MongoDB баз даних та як результат було обрано нереляційну базу даних MongoDB.

В останньому розділі було описано ідею самого проєкту, проведено технічний аудит, проаналізовано ринкові можливості запуску стартап-проєкту. Далі провівши аналіз альтернативних технологій, можна сказати, що технологічна реалізація продукту можлива при використанні декількох різних технологій. Для даної ідеї стартап-проєкту обрано технологію №1 з використанням та застосуванням мови програмування C++. Після аналізу ринкових можливостей запуску стартап-проєкту, можна сказати, що дана система є конкурентною. На ринку існують деякі прямі конкуренти, але вони мають певну кількість недоліків. Ми дійшли до спільного рішення, що нашу систему слід розробити, врахувавши недоліки наявних систем та побажання користувача. Крім цього, ми провели ступеневий аналіз конкуренції та описали її характеристики й вплив на діяльність компанії. Останнім кроком було розроблення маркетингової програми, яка складається з концепції товару, його збуту, просування товару та попередній аналіз можливостей утворення ціни. Також аналіз ринку спирається на цінності та потреби потенційних клієнтів, їх конкурентні переваги, стан та динаміку ринкового простору. Отже, проаналізувавши концепції та складові майбутнього товару, ми обрали відповідну ринкову стратегію та поведінку.

Результатом магістерської дисертації є розроблена автоматизована система збору та обробки даних з датчиків робототехнічного модуля, що забезпечує віддалений контроль стану приміщення та конфігурацію параметрів збору. Однією із основних функцій автоматизованої системи є можливість, що надає користувачу віддалено конфігурувати кредити підключення до Wi-Fi мережі, якою потім і буде виконано відправлення даних користувачу, а також можливість зберігати конфіденційні дані до флешпам'яті пристрою.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розумний будинок на основі Arduino – URL: <https://future2day.ru/umnyj-dom-na-osnove-arduino/>
2. Raspberry Pi – URL: <https://3d-diy.ru/wiki/arduino-platy/obzor-plat-raspberry-pi/>
3. Опис мікроконтролера ESP32 – URL: <http://micpic.ru/home/proekty-na-esp32/194-opisanie-mikrokontrollera-esp32.html>
4. Датчик інтенсивності світла GY-302 (BH1750) – URL: <https://3d-diy.ru/wiki/arduino-datchiki/datchik-intensivnosti-sveta-gy-302-bh1750/>
5. Датчик вологості та температури DHT11 – URL: <https://3d-diy.ru/wiki/arduino-datchiki/datchik-vlazhnosti-i-temperatury-dht11/>
6. Датчик MQ-2 – URL: <http://zi-zi.ru/module/senser-mq2>
7. Як працює датчик газу та диму MQ-2? – URL: <https://radioprogram.ru/post/737>
8. ESP8266/ESP32 Connect WiFi Made Easy – URL: <https://www.hackster.io/hieromon-ikasamo/esp8266-esp32-connect-wifi-made-easy-d75f45>
9. Шина I2C в мікроконтролері ESP32 – URL: [http://digitrode.ru/computing-devices/mcu\\_cpu/2297-esp32-i-interfeys-i2c.html](http://digitrode.ru/computing-devices/mcu_cpu/2297-esp32-i-interfeys-i2c.html)
10. Основи мови програмування C++ – URL: [http://www.znannya.org/?view=C++\\_basics](http://www.znannya.org/?view=C++_basics)
11. Робота з Arduino IDE – URL: <https://alexgyver.ru/lessons/arduino-ide/>
12. Запуск телеграм бота за допомогою PHP: <https://coderlessons.com/articles/veb-razrabotka-articles/kak-zapustit-bot-telegram-s-php>
13. Що таке HTML – URL: <https://www.computerhope.com/jargon/h/html.htm>
14. Визначення CSS – URL: <https://techterms.com/definition/css>
15. Знайомство з JavaScript – URL: <https://javascript.info/intro>
16. Як працює JavaScript – URL: <https://uk.education-wiki.com/7068159-how-javascript-works>
17. Введення в мову JavaScript – URL: <https://learn.javascript.ru/intro>
18. MySQL та MongoDB – URL: <https://habr.com/ru/post/322532/>

19. Вступ до Telegram Bot API – URL: <https://chatbotslife.com/introduction-to-the-telegram-bot-api-part-1-2ae36f7b30a4>
20. Розроблення стартап-проекту – URL: [https://ep.kpi.ua/files/navchannia/mag/roz\\_startap\\_proektiv\\_met\\_vk.pdf](https://ep.kpi.ua/files/navchannia/mag/roz_startap_proektiv_met_vk.pdf)
21. Русакова, Е. А. Системы сбора информации : учебное пособие /Е. А. Русакова. – Екатеринбург: УрГУПС, 2016. – С. 220 – 259.
22. Developing IoT Projects with ESP32, Published August 2021 by Packt Publishing Ltd, 2021. – С.24 – 45.
23. Electronics Projects with the ESP8266 and ESP32 Building Web Pages, Applications, and WiFi Enabled Devices, by Neil Cameron, Edinburgh, 2019. – С.102 – 120.
24. IoT Development for ESP32 and ESP8266 with JavaScript, by Peter Hoddie, Menlo Park, CA, USA, 2018. – С.56 – 78.
25. Introduction to BH1750 – URL: <https://wiki.dfrobot.com/Bh1750>
26. MQ-2 Gas Sensor Module – URL: [http://wiki.sunfounder.cc/index.php?title=MQ-2\\_Gas\\_Sensor\\_Module](http://wiki.sunfounder.cc/index.php?title=MQ-2_Gas_Sensor_Module)
27. Temperature and Humidity Sensor Module – URL: [http://wiki.sunfounder.cc/index.php?title=Humiture\\_Sensor\\_Module](http://wiki.sunfounder.cc/index.php?title=Humiture_Sensor_Module)
28. Temperature and Humidity Sensor Module – URL: [http://wiki.sunfounder.cc/index.php?title=Humiture\\_Sensor\\_Module](http://wiki.sunfounder.cc/index.php?title=Humiture_Sensor_Module)
29. Використання датчиків руху на базі модуля ESP32 – URL: <https://diytech.ru/projects/rabota-s-ik-datchikom-dvizheniya-s-ispolzovaniem-preryvaniy-i-taymerov-na-esp32>
30. Що таке HTML – URL: <https://uk.cmgindia.org/html-2019>
31. ESP32 – розшифрування пінів та характеристики – URL: <http://skproj.ru/esp32-raspinovka-i-harakteristiki>