

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ**

**Кафедра радіотехніки та радіоелектронних систем**

До захисту допущено:

«На правах рукопису»

Завідувач кафедри \_\_\_\_\_ Ігор АНІСІМОВ

« \_\_ » червня 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему:

**«СТВОРЕННЯ СИСТЕМИ КОМУНІКАЦІЇ PEER-TO-PEER НА БАЗІ  
МІКРОКОНТРОЛЕРІВ ESP32 ДЛЯ ДОМАШНЬОЇ АВТОМАТИЗОВАНОЇ  
МЕРЕЖІ SMARTHOME»**

**Виконав:**

студентка 4-го курсу

денної форми навчання

спеціальності 172 - Телекомунікації та радіотехніка

ОП «Інформаційна безпека телекомунікаційних систем і мереж»

Литовченко Олександра Петрівна \_\_\_\_\_

**Науковий керівник:**

к.т.н., с.н.с. Гахович Сергій Вікторович \_\_\_\_\_

**Рецензент:**

к.т.н., доц. Савченко Тетяна Віталіївна \_\_\_\_\_

Засвідчую, що у цій бакалаврській роботі

немає запозичень з праць інших авторів без

відповідних посилань

Студент \_\_\_\_\_

Робота допущена до захисту в ЕК рішенням кафедри радіотехніки та радіоелектронних систем від «23» червня 2023 р., протокол № 22.

Завідувач кафедри радіотехніки та радіоелектронних систем,

доктор фіз.-мат. наук, професор

Анісімов Ігор Олексійович \_\_\_\_\_

Київ 2023

**ЗМІСТ**

|   |    |
|---|----|
| ВСТУП.....  | 3  |
| РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРИ.....  | 5  |
| 1.1 Протокол ESP-NOW.....   | 5  |
| 1.2 Використання IoT в системах домашньої автоматизації (Smart-Home)..... | 6  |
| 1.3 Актуальність IoT.....   | 10 |
| 1.4 Обґрунтування вибору MC ESP-NOW.....                                  | 14 |
| РОЗДІЛ 2 АПАРАТНА ЧАСТИНА СИСТЕМИ SMART-HOME.....                         | 17 |
| 2.1 Сімейство мікросхем ESP32, ESP8266.....                               | 17 |
| 2.2 Ознайомлення с давачами mq7, mq135, dht11.....                        | 21 |
| 2.3 Детальний опис ESP-NOW.....   | 24 |
| 2.4 Peer-to-peer комунікація.....   | 30 |
| 2.5 Збір повної системи SMART-HOME.....                                   | 34 |
| РОЗДІЛ 3 ПРОГРАМНИЙ КОД СИСТЕМИ SMART-HOME.....                           | 42 |
| 3.1 Програмний код мікроконтролера ESP32.....                             | 42 |
| ВИСНОВКИ.....   | 47 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....   | 50 |

## ВСТУП

**Актуальність вибору теми** Актуальність теми «Створення системи комунікації peer-to-peer на базі мікроконтролерів ESP32 для домашньої автоматизованої мережі SmartHome» полягає в тому, що з ростом розповсюдження різних пристроїв для SmartHome, з'являється потреба в створенні зручної та ефективної системи керування цими пристроями.

Одним із ключових факторів успіху в цьому напрямку є можливість забезпечити надійний та швидкий обмін даними між різними пристроями.

При цьому, цілком зрозуміло, що така система має бути простою та зручною у використанні для користувача, а також повинна мати низькі вимоги до енергоспоживання.

Саме тому використання мікроконтролерів ESP32 у системі SmartHome стає дедалі популярнішим, оскільки вони забезпечують низькі витрати енергії та мають широкий функціональний діапазон.

Створення системи комунікації peer-to-peer на базі мікроконтролерів ESP32 для домашньої автоматизованої мережі SmartHome має великий потенціал для подальшого розвитку та вдосконалення зручності та функціональності систем SmartHome.

**Метою даної роботи** є розробка та реалізація системи комунікації peer-to-peer на базі мікроконтролерів ESP32 для домашньої автоматизованої мережі SmartHome, що забезпечуватиме ефективний обмін даними та керуванням підключеними до системи пристроями.

**Об'єктом дослідження** даної роботи є система комунікації peer-to-peer на базі мікроконтролерів ESP32 для домашньої автоматизованої мережі SmartHome. Основна мета дослідження полягає у створенні такої системи, яка забезпечить ефективну та зручну взаємодію різних пристроїв в домашній мережі SmartHome через бездротовий канал зв'язку.

Для досягнення цієї мети буде розглянуто та проаналізовано різні аспекти створення такої системи комунікації, включаючи вибір конкретних

пристроїв, розробку алгоритмів взаємодії між ними, вибір та налаштування мікроконтролерів ESP32 тощо.

**Предметом дослідження** є розробка та налагодження системи комунікації peer-to-peer на базі мікроконтролерів ESP32 для домашньої автоматизованої мережі SmartHome.

Дослідження передбачає аналіз можливостей цієї системи, вирішення проблем, що виникають при її розробці та використанні, а також практичне використання розробленої системи в реальних умовах.

## РОЗДІЛ 1

### ОГЛЯД ЛІТЕРАТУРИ

#### 1.1 Протокол ESP-NOW

Технологія ESP-NOW – це спрощений протокол зв'язку WiFi з передачею коротких пакетів між парами парних пристроїв, розроблений та випущений Espressif в 2016 році для мікроконтролерів ESP8266 та ESP32. Додаткові процедури, пов'язані з підтримкою протоколу WiFi, не використовуються, що прискорює процес обміну пакетами.

ESP-NOW може застосовуватися в Інтернеті Речей для керування інтелектуальними джерелами світла, реле, розетками, іншими пристроями дистанційного керування, отримання інформації від датчиків та інших програм.

ESP-NOW підтримує такі функції:

Зашифрований і незашифрований зв'язок між парами пар.

Змішані зашифрований та незашифрований зв'язок між сполученими пристроями.

Передача до 250 байт корисної інформації.

Налаштування функції зворотного дзвінка для інформування прикладного рівня, зокрема, про успішність або збій передачі.

ESP-NOW також має такі особливості та обмеження:

Швидкість передачі — трохи більше 1 мбіт/с частоті 2,4 ГГц, тобто. ESP-NOW працює на тій же частоті та каналах, що і роутер WiFi.

Протокол WiFi не використовується. Використовується протокол (протокол ESP-NOW використовується як альтернатива протоколу WiFi для реалізації комунікації в системі Smart-Home на базі мікроконтролерів ESP32.) аналогічний протоколу з низьким енергоспоживанням, який використовується в бездротовій миші 2,4 ГГц. Потрібне лише початкове сполучення. Після сполучення з'єднання не розривається. Широкомовна передача не підтримується - тільки множинна роздача парам парних

пристроїв. Максимум 20 пар, включаючи зашифровані, підтримуються на одному пристрої, включаючи зашифровані пари. Максимум 10 пар зашифрованих підтримуються в режимі Station. Максимум 6 у режимі SoftAP або SoftAP+Station. Шифрування багатоадресної розсилки не підтримується.

## 1.2 Використання IoT в системах домашньої автоматизації (Smart-Home)

IoT системи включають в себе сенсори, пристрої збору даних, мережеві з'єднання та облікові записи для обміну та аналізу даних. Основна ідея полягає в тому, що різні пристрої можуть спілкуватися між собою та з обліковими записами через мережеві з'єднання, щоб збирати, передавати і аналізувати дані[1].

Використання IoT систем відкриває безліч можливостей у різних сферах, включаючи:

1. Смарт-дом: IoT дозволяє з'єднати різні пристрої в домашній мережі, що дозволяє контролювати освітлення, опалення, кондиціонування повітря, системи безпеки, аудіо- та відеосистеми, електроприлади та інші пристрої за допомогою мобільних додатків або голосових помічників.



Рисунок 1.1 - Смарт-дом: IoT

2. Смарт-місто: IoT може використовуватися для управління різними аспектами міського середовища, такими як освітлення вулиць, управління водопостачанням, оптимізація транспортних потоків, збір даних про якість повітря та інші параметри.



Рисунок 1.2 - Смарт-місто: IoT

3. Здоров'я та медицина: IoT може бути використаний для моніторингу стану здоров'я пацієнтів, віддаленого контролю за хронічними захворюваннями, відстеження фізичної активності та взаємодії з медичними пристроями.

4. Промисловість: IoT дозволяє створювати "розумні" заводи та машини, які автоматизують процеси виробництва, моніторять та аналізують стан обладнання, забезпечують прогнозування поломок та оптимізацію енергоспоживання. Використання IoT в промисловості дозволяє підвищити ефективність виробництва, знизити витрати на обслуговування та покращити безпеку працівників.

5. Сільське господарство: IoT може бути використаний для моніторингу рослин, ґрунту та погодних умов, автоматизації поливу, внесення добрив та контролю над врожайністю. Це допомагає фермерам здійснювати розумне та ефективне управління сільськогосподарськими процесами.

6. Транспорт: IoT може використовуватися для відстеження, моніторингу та оптимізації руху транспорту, контролю технічного стану транспортних засобів, покращення безпеки на дорогах та впровадження "розумних" систем паркування.

7. Роздрібна торгівля: IoT дозволяє реалізувати концепцію "розумного" магазину, де пристрої, сенсори та аналітика допомагають відстежувати запаси, покупки клієнтів, оптимізувати розміщення товарів та надавати персоналізовані послуги покупцям.

Використання IoT систем дозволяє збирати великі обсяги даних, забезпечувати взаємозв'язок між пристроями та системами, автоматизувати процеси, покращувати ефективність та якість життя. Водночас, це ставить нові виклики щодо безпеки, конфіденційності та управління великими обсягами даних. Отже, ефективне використання IoT вимагає ретельного планування, розробки і впровадження механізмів захисту даних та встановлення надійних з'єднань між пристроями. Для успішної реалізації IoT систем необхідно також забезпечити масштабованість і гнучкість в системі, щоб вона могла легко адаптуватися до змінних потреб та технологічних розвитків.

Оцінка ефективності використання IoT систем в порівнянні з існуючими традиційними системами комунікації залежить від конкретних вимог та ситуації. Однак, IoT системи зазвичай мають деякі переваги, такі як:

1. Збір та аналіз великого обсягу даних: IoT дозволяє збирати і обробляти великі обсяги даних з різних джерел, що дозволяє отримати більш детальну та точну інформацію для прийняття рішень.

2. Автоматизація та оптимізація процесів: IoT дозволяє автоматизувати багато рутинних та повторюваних завдань, що сприяє збільшенню продуктивності, зниженню помилок та оптимізації ресурсів.

3. Взаємодія між пристроями та системами: IoT системи дозволяють пристроям спілкуватися між собою та взаємодіяти з іншими системами, що створює нові можливості для інтеграції та розширення функціональності.

4. Гнучкість та масштабованість: IoT системи можуть бути легко розширювані та адаптовані до змінних потреб, що дозволяє розширювати їх функціональність і використовувати їх у різних сферах.

Проте, використання IoT систем також може мати свої виклики, включаючи проблеми з безпекою даних, конфіденційністю, надійністю та приватністю, потребу високошвидкісного та стійкого мережевого з'єднання, складність інтеграції з існуючими системами та стандартами, а також питання енергоефективності.

Оцінка ефективності використання IoT систем залежить від конкретного застосування і вимог користувача. Для деяких сфер, таких як смарт-доми, IoT може значно полегшити і поліпшити повсякденне життя, забезпечуючи зручний та централізований контроль над різними пристроями та системами. У промисловості, IoT може допомогти впровадити моніторинг та автоматизацію процесів, що призводить до зниження витрат та підвищення продуктивності. В сільському господарстві, IoT може покращити ефективність рослинництва та тваринництва, дозволяючи контролювати та оптимізувати умови росту та вирощування.

Порівняно з традиційними системами комунікації, IoT системи зазвичай пропонують більш широкі можливості і гнучкість у зборі та обробці даних, автоматизації процесів та взаємодії між пристроями. Вони також можуть бути більш доступними у плані вартості та масштабованості. Однак, враховуючи проблеми безпеки та конфіденційності, важливо ретельно проектувати та реалізовувати заходи захисту даних та мережевої інфраструктури в IoT системах.

У підсумку, використання IoT в системах комунікації, зокрема в розумних будинках, має потенціал значно поліпшити зручність, ефективність та функціональність.

Враховуючи суперечливість ситуацій та виклики, пов'язані з безпекою та приватністю, важливо враховувати ці аспекти під час розробки та використання IoT систем.

Крім того, успішне використання IoT систем вимагає детального планування та аналізу вимог, вибору підходящих пристроїв та сенсорів, розробки мережевої інфраструктури, розробки програмного забезпечення для збору, передачі та аналізу даних, а також надійного забезпечення безпеки та захисту даних.

Важливим етапом є також інтеграція IoT систем з існуючими інфраструктурами та додатками, що може вимагати розробки спеціальних інтерфейсів та протоколів зв'язку.

Успішне впровадження IoT систем вимагає спільної роботи між розробниками, виробниками, постачальниками послуг та користувачами, щоб забезпечити відповідність до стандартів, ефективну інтеграцію та вирішення проблем, що виникають. Зрозуміння потреб та вимог користувачів, а також оцінка ризиків та можливостей є ключовими факторами успіху впровадження IoT систем в IoT системи комунікації.

Враховуючи всі ці аспекти, використання IoT систем в IoT системах комунікації може принести значні переваги, забезпечуючи зручність, ефективність та покращення якості життя. Важливо пильно стежити за проблемами безпеки, приватності та інтеграції, щоб забезпечити надійну та безпечну роботу цих систем.[2]

### **1.3 Актуальність IoT**

Актуальність Internet of Things (IoT) - полягає в тому, що ця технологія перетворює наш світ, надаючи нам здатність підключати різні фізичні об'єкти до Інтернету і забезпечувати їх взаємодію та обмін даними.

ІоТ відкриває безліч нових можливостей для сполучення, моніторингу, контролю та оптимізації різних аспектів нашого життя.

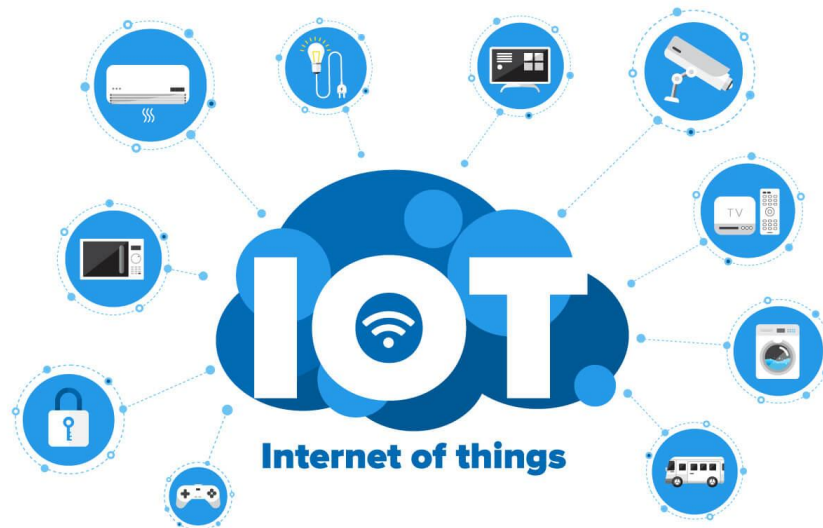


Рисунок 1.3 - Internet of Things

Одна з ключових актуальних галузей застосування ІоТ - це сфера "розумного дому" (Smart Home), де різні пристрої та системи в будинку підключаються до мережі Інтернет, що дозволяє користувачам контролювати та управляти своїм житлом з будь-якого місця за допомогою мобільних пристроїв або голосових асистентів. Це дає можливість забезпечити комфорт, енергоефективність, безпеку та зручність у побутових умовах.

Інші актуальні застосування ІоТ включають в себе "розумне місто" (Smart City), де різні міські системи, такі як освітлення, транспорт, водопостачання та управління відходами, підключаються до мережі Інтернет з метою покращення ефективності та якості міського середовища. ІоТ також знаходить застосування в галузях промисловості, сільського господарства, охорони здоров'я, логістики, енергетики та багатьох інших.

Ця технологія дозволяє збирати великі обсяги даних з різних джерел, аналізувати їх і використовувати для прийняття розумних рішень та

оптимізації процесів. В додаток до цього, IoT може покращити ефективність використання ресурсів, зменшити витрати, підвищити безпеку та зручність для користувачів. Наприклад, в сфері енергетики IoT може допомогти встановити "розумні лічильники", які автоматично передають дані про споживання електроенергії, дозволяючи споживачам ефективно контролювати та оптимізувати своє енергоспоживання. У сфері транспорту IoT може використовуватися для впровадження "розумних систем управління транспортом", які оптимізують рух транспортних засобів, зменшують затори та витрати палива.

Проте, разом з перевагами IoT також існують певні виклики та проблеми. Одна з них - це забезпечення безпеки та конфіденційності зібраних даних. З великою кількістю підключених пристроїв, існує ризик злому та несанкціонованого доступу до особистої інформації. Також важливим викликом є стандартизація і сумісність між пристроями та системами IoT, оскільки вони можуть бути розроблені різними виробниками з різними протоколами комунікації.

Актуальність IoT полягає в його потенціалі трансформувати різні сфери нашого життя, надаючи нові можливості для покращення ефективності, зручності та якості послуг.

Для успішної реалізації IoT необхідно вирішити виклики, пов'язані з безпекою, конфіденційністю, стандартизацією та інтегрованими, що становить виклик для індустрії та науково-дослідницьких організацій.

Децентралізація вбудованих системах:

Децентралізація в вбудованих системах відноситься до розподіленої архітектури, де обчислювальні задачі розподілені між окремими вузлами або пристроями, реалізована вбудованих системана (Рис 1.4).

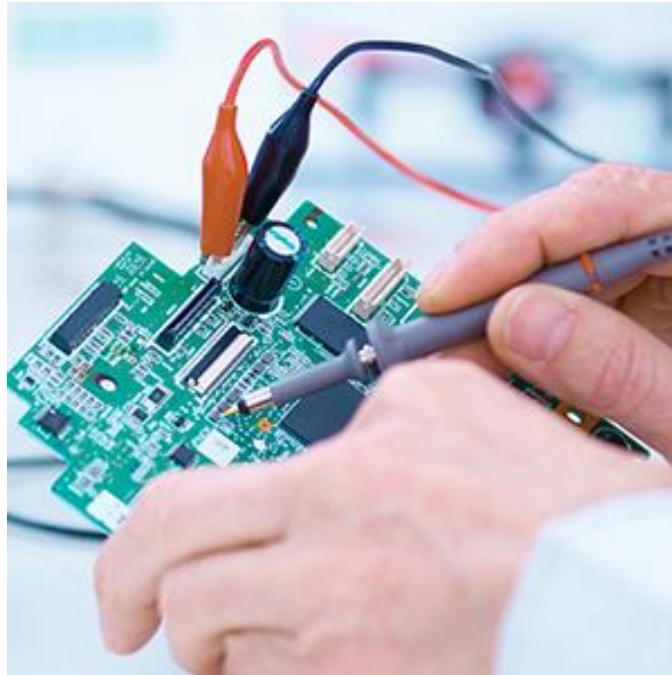


Рисунок 1.4 Вбудована система

Це означає, що функціональні можливості інтегрованих систем не обмежені централізованим контролем.

Натомість, вони можуть бути розподілені між багатьма незалежними елементами, які можуть обмінюватися даними і виконувати обчислення без необхідності взаємодії з центральною точкою керування.

Децентралізована архітектура може мати кілька переваг в порівнянні з централізованою архітектурою. Вона може забезпечувати більшу надійність і стійкість до відмов, оскільки вона не залежить від єдиного центрального вузла. Крім того, децентралізована архітектура може забезпечувати більшу масштабованість, оскільки вона дозволяє додавати нові пристрої без необхідності перерозподілу ресурсів на центральному вузлі.

Децентралізовані системи можуть бути реалізовані за допомогою різних технологій, включаючи мережі сенсорів, мережі IoT та блокчейн. Вони можуть бути застосовані в багатьох галузях, включаючи автоматизацію будинків, медичну діагностику, промисловість та багато іншого.

## 1.4 Обґрунтування вибору MC ESP-NOW -?

Вибір мікроконтролера ESP-NOW для дипломної роботи є обґрунтованим з кількох причин.

По-перше, ESP-NOW є потужним інструментом для створення системи комунікації peer-to-peer (P2P) на базі мікроконтролера ESP32. ESP32 володіє високою продуктивністю, низьким енергоспоживанням і широким спектром функціональних можливостей, що дозволяє реалізувати різноманітні задачі в сфері домашньої автоматизації.

По-друге, ESP-NOW є спеціально розробленою протоколом для бездротової комунікації між пристроями на базі ESP32. Цей протокол забезпечує низьку затримку, високу швидкість передачі даних і надійність зв'язку. Він також підтримує режим безперервного моніторингу, що дозволяє ефективно використовувати ресурси мікроконтролера.

По-третє, ESP-NOW є відкритим протоколом, що забезпечує гнучкість і можливість розширення функціональності системи комунікації. Він підтримує передачу різних типів даних, включаючи текстові повідомлення, бінарні дані і сигнали керування. Крім того, він може бути легко інтегрований з іншими компонентами системи IoT, що дозволяє розширювати функціональність розумного будинку.



Рисунок 1.5 – ESP-NOW

Одним із завдань дипломної роботи є дослідження можливостей ESP-NOW і розробка системи комунікації на його базі для домашньої автоматизованої мережі SmartHome. Використання ESP-NOW дозволить створити надійну, швидку та ефективну систему зв'язку між різними пристроями у розумному будинку, такими як освітлення, термостати, датчики руху та інші. Це дозволить користувачам зручно керувати різними функціями будинку за допомогою одного центрального інтерфейсу, такого як мобільний додаток або голосовий помічник.

ESP-NOW володіє низькою вартістю і доступністю, що робить його привабливим вибором для реалізації системи комунікації в розумному будинку. Крім того, його низьке енергоспоживання дозволяє забезпечити тривалу автономну роботу пристроїв, що оснащені мікроконтролерами ESP32.

Таблиці, формули та ілюстрації в межах одного завдання у роботі нумеруються наскрізно (для прикладу: Рис. 1.2 – 2 рисунок в межах виконання 1 завдання). У правому верхньому куті пишеться слово “Таблиця” із зазначенням її номера. При посиланні на таблицю вказують її повний номер, а слово „Таблиця” пишуть у скороченому вигляді, наприклад, “(табл. 1.3)”. Під словом “Таблиця” розміщується заголовок таблиці – симетрично до форми таблиці. Слово “Таблиця” і заголовок починаються з великої букви. Назва не підкреслюється. При переносі частини таблиці на інший аркуш (сторінку) слово “Таблиця” і номер її вказують один раз справа над першою частиною таблиці, над іншими частинами пишуть слова “Продовження табл.” і вказують номер таблиці, наприклад: “Продовження табл. 1.2”. Кожна таблиця має мати заголовок, в якому вказується адресність і період, що розглядається. У таблиці мають бути вказані одиниці виміру поданих величин. Якщо розмірність їх всіх однакова, то вона вказується у дужках під назвою таблиці.

Ілюстрації і таблиці необхідно подавати в роботі безпосередньо після тексту, де вони згадані вперше, або на наступній сторінці. Ілюстрації позначають словом “Рис.” і нумерують послідовно в межах завдання, за виключенням ілюстрацій, винесених у додатки. Номер ілюстрації повинен складатися з номера завдання і порядкового номера ілюстрації, між якими ставиться крапка. Наприклад, Рис. 1.2 - другий рисунок першого завдання. Номер ілюстрації, її назва і пояснювальні підписи розміщують послідовно під ілюстрацією.

Актуальність самої ESP-NOW наведено на табл. 1.1.

Таблиця 1.1

#### Актуальність ESP-NOW

| <b>Аспекти актуальності</b> | <b>Опис</b>  |
|-----------------------------|--|
| Ефективність                | ESP-NOW забезпечує швидку та ефективну передачу даних між пристроями IoT.  |
| Низька енергоспоживання     | Мікроконтролери ESP32, що використовують ESP-NOW, мають низьке енергоспоживання, що сприяє тривалій роботі на батареї. |
| Простота використання       | ESP-NOW має простий інтерфейс програмування та легко інтегрується з іншими пристроями та платформами IoT.              |
| Надійність                  | Протокол ESP-NOW працює на низькому рівні і забезпечує стійку та надійну передачу даних без зайвих перешкод.           |
| Сумісність                  | ESP-NOW сумісний з мікроконтролерами ESP32, що робить його доступним для широкого спектра пристроїв IoT.               |

Вибір ESP-NOW для дипломної роботи є обґрунтованим з технічних та практичних міркувань. Використання цього протоколу дозволить розробити ефективну та надійну систему комунікації в розумному будинку, що задовольнятиме потреби користувачів у зручному керуванні та автоматизації різних пристроїв.

## РОЗДІЛ 2

### АПАРАТНА ЧАСТИНА ДЛЯ СИСТЕМИ SMART-HOME

#### 2.1 Сімейство мікросхем ESP32, ESP8266

1) ESP32 - серія недорогих мікросхем з малим енергоспоживанням китайської компанії Espressif Systems . Є системою на кристалі з інтегрованим контролерами радіозв'язку Wi-Fi , Bluetooth і Thread . У серіях ESP 32 та ESP32-S використовуються процесорні ядра з архітектурою компанії Tensilica , а в серіях ESP32-C та ESP32-H – ядра з відкритою архітектурою RISC-V .

У мікросхемі інтегрований радіочастотний тракт: симетруючий трансформатор , вбудовані антенні комутатори, радіочастотні компоненти, підсилювач малощумний , підсилювач потужності, фільтри і модулі управління живленням. ESP32 створений і розроблений компанією, розташованою в Шанхаї , а виробляється компанією TSMC за техпроцесом 40 нм та 28 нм. Серія є наступником мікросхем ESP8266.

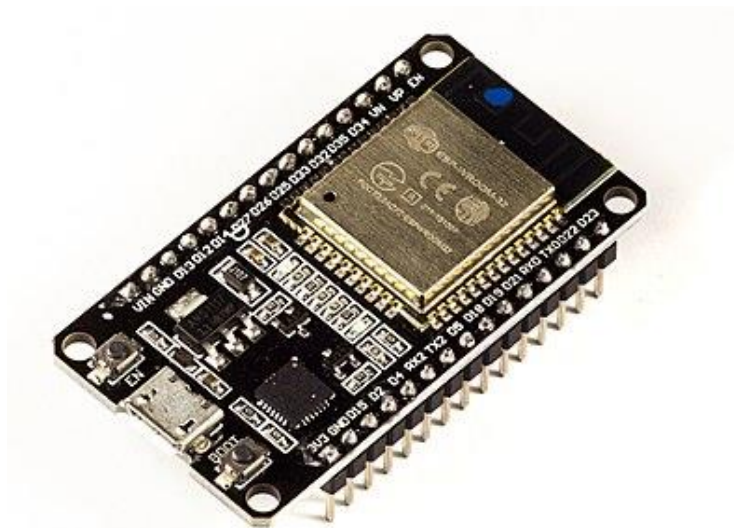


Рисунок 2.1 – ESP32

Особливості серії ESP32 і ESP32-S включають:

Мікроконтролер та управління

Tensilica Xtensa LX6 двоядерний (або одноядерний) 32-розрядний процесор, з тактовою частотою 160 або 240 МГц і продуктивністю до 600 DMIPS ( Dhrystone MIPS )

Співпроцесор із ультранизким енергоспоживанням

Пам'ять: 520 КБ пам'яті SRAM

Бездротовий зв'язок:

Wi-Fi : 802.11 b/g/N

Bluetooth : v4.2 BR/EDR and BLE

Периферійні інтерфейси:

12-розрядний АЦП до 18 каналів

2 × 8 біт ЦАПа

10 - портів для підключення ємнісних датчиків (вимірюють ємність GPIO )

Давач температури відсутній. Інформація про нього видалена зі специфікації V2.2 [2]

4 × SPI майстер-інтерфейсу ( провідні пристрої )

2 × I<sup>2</sup>S майстер-інтерфейсу

2 × I<sup>2</sup>C майстер-інтерфейсу

3 × UART інтерфейсу

SD /SDIO/CE-ATA/ MMC / eMMC хост-контролер

SDIO/SPI слейв-контролери ( відомі пристрої )

Ethernet MAC interface з виділеним DMA та IEEE 1588 Precision Time Protocol support

CAN bus 2.0

ІЧ дистанційне керування (передавач/приймач, до 8 каналів)

Можливість підключення двигунів та світлодіодів через ШІМ -вихід

Датчик холу

Аналоговий підсилювач низького енергоспоживання

Безпека:

Підтримуються всі функції безпеки стандарту IEEE 802.11 , у тому числі WPA, WPA / WPA2 та WAPI

Безпечне завантаження

Шифрування флеш-диску

1024-бітний ключ до 768 біт для клієнтів

Криптографічне апаратне прискорення: AES , SHA-2 , RSA , криптографії на основі еліптичних кривих (ECC), апаратний генератор випадкових чисел при включеному WiFi або Bluetooth, інакше використовується генератор псевдовипадкових чисел

Управління живленням:

Лінійний регулятор з низьким рівнем падіння напруги

Індивідуальне харчування для RTC

споживання 5-2,5 мкА в режимі "глибокий сон"

Пробудження з переривання від GPIO, таймера, вимірювання АЦП, переривання ємнісного сенсорного датчика

Робоча напруга від 2,2-3,6 В

Робоча температура від -40 °С до +125 °С

Максимальна швидкість передачі даних 150 Мбіт/с при 11n HT40, 72 Мбіт/с при 11n HT20, 54 Мбіт/с @ 11g, і 11 Мбіт/с при 11b

Максимальна потужність передачі 19,5 дБм @ 11b, 16,5 дБм @ 11 г, 15,5 дБм @ 11n

Мінімальна чутливість приймача: 98 дБм

Стійка пропускна здатність UDP 135 Мбіт/с

2) ESP8266

ESP8266 - мікроконтролер китайського виробника Espressif Systems з інтерфейсом Wi-Fi . Крім Wi-Fi, мікроконтролер відрізняється відсутністю флеш-пам'яті в SoC, програми користувача виконуються із зовнішньої флеш-пам'яті з інтерфейсом SPI .

Мікроконтролер привернув увагу в 2014 році у зв'язку з виходом перших товарів на його основі за дуже низькою ціною.

Виробництво ESP8285, що поєднує ESP8266 та флеш-пам'ять на 1 МБайт. Espressif представила розвиток лінійки — мікросхему ESP32 та модулі на її основі.

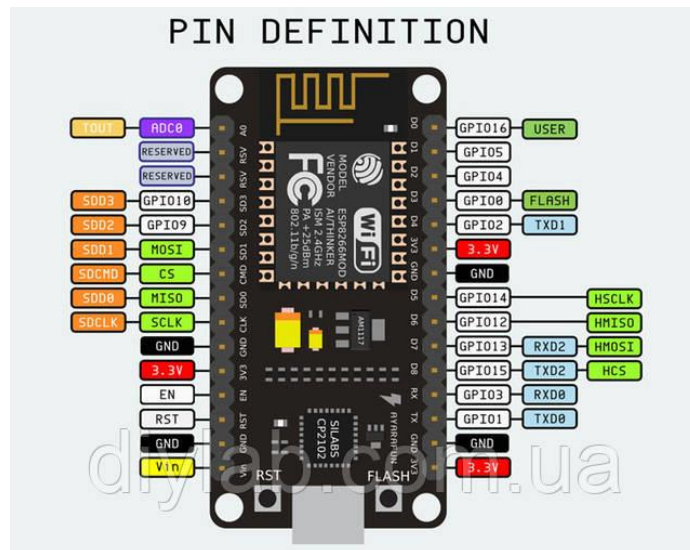


Рисунок 2.2 - ESP8266

Програмні засоби розробки (програмний комплект розробника, SDK) складаються з:

Компілятори. Компілятор для Xtensa LX106 входить до пакету компіляторів GNU Compiler Collection . Компілятор має відкриті вихідні тексти. У різних SDK можуть міститися різні зборки цього компілятора, що трохи відрізняються підтримуваними опціями.

Бібліотек до роботи з периферією контролера, стеків протоколів WiFi, TCP/IP .

Засобів завантаження файлу, що виконується, в пам'ять програм мікроконтролера.

Опціональної IDE .

Espressif вільно розповсюджує свій комплект розробника. У цей комплект входить компілятор GCC, бібліотеки Espressif та завантажувальна

утиліта XTCOM. Бібліотеки постачаються у вигляді скомпільованих бібліотек, без вихідних текстів. Espressif підтримує дві версії SDK: одна на основі RTOS , інша на основі зворотних викликів (callback).

Крім офіційної SDK існує низка проектів альтернативних SDK.

Ці SDK використовують бібліотеки Espressif або пропонують власний еквівалент бібліотек Espressif, отриманий методами реверсинжинірингу.

## 2.2 Ознайомлення с давачами mq7, mq135, dht11

### 1) Давач - mq7

Давач MQ-7 CO Gas Sensor реагує на наявність у повітрі чадного газу . Також датчик має чутливість до деяких інших газів, що мають вуглець у своєму складі (пропан, пропілен, бутилен). Датчик має високу чутливість та тривалий термін служби[8].

У модулі MQ-7 є два виходи - аналоговий - A0 і дискретний - DO. Напруга на аналоговому виході змінюється пропорційно концентрації чадного газу CO в повітрі . Дискретний вихід видає логічний "0" при концентрації газу нижче, виставленої вбудованим потенціометром, та лог "1" - при перевищенні цього порога.

На платі встановлено два світлодіоди: PWR-LED – наявність живлення, DO-LED – присутність високого рівня на дискретному виході.



Рисунок 2.3 – Давач MQ7

Таблиця 2.1 – Характеристика давача MQ7

| Характеристика              | Значення               |
|-----------------------------|------------------------|
| Напруга живлення            | 5В                     |
| Вихідний сигнал             | High/Low та аналоговий |
| Використовуваний компаратор | LM393                  |
| Споживаний струм            | 150 мА                 |
| Діапазон вимірів            | 20-2000ppm             |
| Діапазон робочих температур | -20...70 °С            |
| Розміри                     | 32x20x12 мм            |

## 2) Давач - mq135

Модуль давача якості повітря MQ135 призначений для визначення вмісту та кількості шкідливих і небезпечних газів в повітрі таких як: NH<sub>3</sub>, NO<sub>x</sub>, пари алкоголю, бензину, диму, CO<sub>2</sub> і т.д. Побудований на датчику MQ135.

На платі модуля передбачений компаратор для визначення порогового значення концентрації шкідливих речовин. Поріг спрацьовування задається потенціометром.

Характеристики:

Вихідний сигнал TTL рівня

Аналоговий вихід датчика 0 - 5В. Чим вище концентрація газів тим вище напруга на виході

Напруга живлення модуля: 5В постійного струму

Розміри: 32мм X22мм X27мм

Тип давача: MQ-135



Рисунок 2.4 – Давач MQ135

### 3) Давач - dht11

Давач DHT11 – це цифровий датчик температури та вологості, що дозволяє калібрувати цифровий сигнал на виході. Складається з емнісного датчика вологості та термістора[7].

Також датчик містить в собі АЦП для перетворення аналогових значень вологості та температури.

Характеристики:

Модель виробника: ASAIR DHT11

Визначення вологості: 5 – 95% RH  $\pm$  5% (макс.)

Визначення температури: -20 ~ +60 °C  $\pm$  2% (макс.)

Харчування: 3.5-5.5 В

Частота опитування: трохи більше 1 Гц

Розміри 15.5 x 12 x 5.5 мм

Висновки:

1. VCC (3-5В живлення)
2. Data Out – Виведення даних
3. NC – не використовується
4. Загальний

При підключенні до мікроконтролера ви можете між висновками Vcc і Data розмістити підтягуючий pull-up резистор номіналом 10 кОм. Плата Arduino має інтегровані pull-up, але вони дуже слабкі - близько 100 кОм.

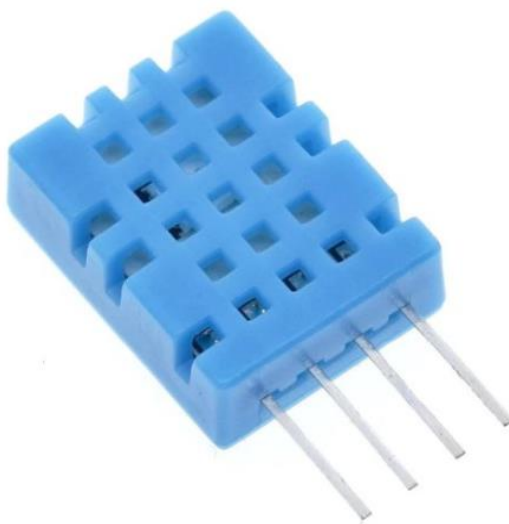


Рисунок 2.5 – Давач DHT11

### 2.3 Детальний опис ESP-NOW

ESP-NOW застосовує технологію кадрів IEEE802.11 Action Vendor разом з функцією IE, розробленою Espressif, та технологією шифрування CCMP, реалізуючи безпечне комунікаційне рішення без встановлення з'єднання. Пристрій з Wi-Fi підтримує основний майстер-ключ (PMK) та кілька локальних майстер-ключів (LMK)

PMK використовується для шифрування LMK за допомогою алгоритму AES-128.

LMK парного пристрою використовується для шифрування інформації користувача методом CCMP. Максимальна кількість різних LMK — 6. Якщо LMK для парного пристрою не заданий, дані користувача шифруватися не будуть.

Базовий рівень:

На нижньому рівні протоколу ESP\_NOW підтримується пов'язаний список, що містить інформацію про локальний пристрій та про сполучений пристрій, у тому числі MAC-адреси та ключі. ESP-NOW також зберігає дані, що часто використовуються, для прикладного рівня, щоб уникнути накладних витрат на повторну обробку зв'язаного списку. Інформація про пристрої використовується для надсилання та отримання даних і включає інформацію про локальний пристрій:

PMK : 16 байт - основний майстер-ключ, який використовується для шифрування ключа на приєднаному пристрої (KOK в API) ESP\_NOW підтримує PMK за промовчаням, тому налаштування не потрібне. Якщо необхідно, переконайтеся, що PMK збігається з локальним пристроєм.

Режим : 1 байт — режим локального пристрою, що визначає передавальний WiFi інтерфейс (SoftAP або STA) ESP-NOW. Режим пари не впливає на будь-яку функцію, а лише зберігає інформацію про режим для прикладного рівня. У режимі STA WiFi застосовується лише Station і SoftAP WiFi — лише SoftAP.

Режими ESP\_NOW роботи локального пристрою:

Інформацію про пару в парі (включаючи інформацію, що часто використовується, та іншу користувальницьку інформацію):

LMK : 16 байт - локальний майстер-ключ, який використовується для шифрування ключа корисної інформації під час зв'язку в цій парі.

MAC-адреса : 6 байт - адреса сполученого пристрою, збігається з адресою відправника. Наприклад, якщо пакет відправляється зі Station, MAC-адреса має збігатися з адресою Station.

Режим : 1 байт — режим локального пристрою, що визначає передавальний інтерфейс (SoftAP або STA) ESP-NOW.

Канал : 1 байт - канал, через який обмінюються даними пристрою, з'єднані в пару. Може мати значення 0..255. Канал не впливає на жодну функцію, а лише зберігає інформацію про канал для прикладного рівня. Значення визначається прикладним рівнем. Наприклад, 0 означає, що канал

не визначено; 1 ~ 14 означає дійсні канали; всім іншим значенням можуть бути призначені функції, визначені прикладним рівнем.

Espressif не рекомендує використовувати тривалі операції у функціях зворотного виклику при надсиланні/відправці пакетів, що пов'язано, ймовірно, з реалізацією алгоритмів, що використовують механізм переривань. На користь цього припущення також говорять і проблеми, пов'язані з динамічним виділенням пам'яті в функціях зворотного виклику, що вирішується кращим використанням статичних змінних, а також неоднозначність застосування механізму виключень MicroPython .

Реалізація асинхронності процесів запуску/завершення, встановлення зв'язку в парі, отримання/передачі пакетів Espressif не описується, що не полегшує застосування ідеології asyncio MicroPython .

Формат пакету ESP-NOW:

Заголовок MAC : 24 байти.

Категорія : 1 байт, що вказує на категорію творця пакета. Встановлено значення (127).

ID організації : 3 байти, містить унікальний ідентифікатор, який є першими трьома байтами MAC-адреси, застосованої Espressif. Встановлено значення (0x18fe34)

Випадкове значення : 4 байти, використовується для захисту даних.

Дані творця пакету : 7-255байт

Дані автора пакета містять такі поля:

ID : 1 байт, Встановлено значення (221).

Довжина : 1 байт, загальна довжина ID організації, типу, версії та даних користувача.

ID організації : 3 байти, містить унікальний ідентифікатор, який є першими трьома байтами MAC-адреси, застосованої Espressif. Встановлено значення (0x18fe34)

Тип : 1 байт, протокол ESP-NOW. Встановлено значення (4)

Версія : 1 байт, поточна версія ESP-NOW. Встановлено (1)

Вміст : 0-250 байт дані користувача.

FCS : 4 байти, контрольна сума

Оскільки ESP-NOW не використовує протокол WiFi, заголовок MAC трохи відрізняється від заголовка стандартних пакетів. Біти FromDS та ToDS поля FrameControl дорівнюють 0. У першому полі адреси задана адреса призначення. У другому полі адреси вказано адресу джерела. Третє поле адреси встановлено як широкомовна адреса (0xff: 0xff: 0xff: 0xff: 0xff: 0xff).

Основний алгоритм застосування, початок та завершення:

Перед використанням ESP-NOW рекомендується встановити інтерфейс Wi-Fi у потрібному режимі. Зазвичай інтерфейс Station встановлюється для CONTROLLER , інтерфейс SoftAP для SLAVE та COMBO . Також доцільно зупинити Wi-Fi після завершення використання ESP-NOW.

Для початку роботи ESP-NOW викликати `esp_now_init()` та `esp_now_deinit()` для завершення. Коли викликається `esp_now_deinit()` , вся інформація про сполучені пристрої видаляється.

Прив'язка функцій зворотного дзвінка

Функція обробки виклику при надсиланні пакета `esp_now_register_send_cb()` може використовуватися для інформування прикладного рівня сторони, що відправляє в парі про успішність або невдачу передачі, наприклад, якщо інформація на підрівні MAC передана успішно.

При використанні `esp_now_register_send_cb()` слід враховувати:

У парі:

Якщо прикладний рівень не отримав пакет, але функція зворотного виклику повернула "success", причиною може бути:

- Атаки від шахрайського пристрою
- помилки встановлення зашифрованого ключа
- втрати пакетів на прикладному рівні Espressif

Якщо прикладний рівень отримав пакет, але функція зворотного виклику повертає помилку, причиною може бути:

Канал зайнятий, і АСК не отримано.

При множинному зв'язку з усіма парами локального пристрою:

Якщо функція зворотного дзвінка повертає «success», це означає, що пакет було надіслано успішно.

Якщо функція зворотного дзвінка повертає помилку, це означає, що пакет не було успішно надіслано.

Функція обробки виклику при отриманні пакета `esp_now_register_receive_cb()` повертає інформацію, що включає MAC-адресу відправляючого пристрою в парі і корисну інформацію. Так само може використовуватися для інформування прикладного рівня відправляючого пристрою в парі від того, що приймає про те, що пакет був успішно прийнятий.

Додавання пари парних пристроїв:

Перш ніж надсилати дані, необхідно додати пристрій до списку пар пар пристроїв викликом `esp_now_add_peer()`. Перед надсиланням групі встановлених пар необхідно додати пристрій з груповою MAC-адресою. Діапазон каналу парних пристроїв становить від 0 до 14. Якщо канал встановлено на 0, дані будуть надсилатися поточним каналом. В іншому випадку канал повинен бути встановлений як канал, на якому знаходиться локальний пристрій.

Безпека:

Якщо ключ потрібно зашифрувати, для налаштування РМК можна викликати `esp_now_set_pmck()`. Якщо РМК не встановлено, буде використовуватися РМК за промовчанням і вибрати той самий ключ для всіх пристроїв. Також для вибраних пар встановити LMK.

Надсилання даних пакетами ESP-NOW

Для надсилання даних ESP-NOW використовується `esp_now_send()`. При цьому функція, встановлена раніше в `esp_now_register_send_cb()`,

поверне `ESP_NOW_SEND_SUCCESS` при надсиланні функції зворотного виклику, якщо дані були успішно прийняті на рівні MAC. Інакше повернеться `ESP_NOW_SEND_FAIL`. Декілька причин можуть призвести до того, що ESP-NOW не зможе надіслати дані. Зокрема,

- цільовий пристрій не існує;
- канали пристроїв не збігаються;
- дані губляться під час передачі.

Не гарантується, що прикладний рівень обов'язково прийме дані. У разі потреби можна надіслати підтвердження при отриманні даних ESP-NOW. При виникненні тайм-ауту підтвердження, передачу даних ESP-NOW слід повторити. Порядковий номер може бути призначений для даних ESP-NOW для видалення дублікатів даних.

При надсиланні даних ESP-NOW через `esp_now_send()` слід враховувати, що за один раз можна надсилати не більше 250 байтів інформації.

**УВАГА!** Надто короткий інтервал між відправкою двох пакетів ESP-NOW може призвести до помилок виконання функції зворотного дзвінка, у зв'язку з чим рекомендується надсилати наступний пакет даних ESP-NOW після того, як функція зворотного дзвінка при обробці попереднього відправлення успішно завершилася. Функція зворотного дзвінка надсилається із високопріоритетного завдання Wi-Fi. Тому не рекомендується виконувати тривалі операції у функції зворотного дзвінка. Натомість можна розмістити необхідні дані у статичну чергу і обробити їх із процесом з нижчим пріоритетом.

Якщо функція надсилання повертає MAC-адресу, вона буде відправлена на пристрій з цією MAC-адресою. Якщо функція надсилання повертає `NULL`, то пакет буде надіслано всім пристроям, приєднаним до відправника, що може призвести до збою передачі або затримки через навантаження мережі.

Отримання даних пакетами ESP-NOW

Функція зворотного дзвінка також запускається із завдання Wi-Fi. Тому не рекомендується виконувати тривалі операції у функції зворотного дзвінка. Натомість можна розмістити необхідні дані у статичну чергу та обробити їх процесом з нижчим пріоритетом.

## **2.4 Peer-to-peer комунікація**

Peer-to-peer, P2P (з англ. — рівний до рівного) — варіант архітектури системи, в основі якої стоїть мережа рівноправних вузлів.

Комп'ютерні мережі типу peer-to-peer (або P2P) засновані на принципі рівноправності учасників і характеризуються тим, що їх елементи можуть зв'язуватися між собою, на відміну від традиційної архітектури, коли лише окрема категорія учасників, яка називається серверами, може надавати певні сервіси іншим.

В чистій мережі «peer-to-peer» не існує поняття клієнтів або серверів, лише рівні вузли, які одночасно функціонують як клієнти та сервери по відношенню до інших вузлів мережі. Ця модель мережевої взаємодії відрізняється від клієнт-серверної архітектури, в якій зв'язок відбувається лише між клієнтами та центральним сервером. Така організація дозволяє зберігати працездатність мережі при будь-якій конфігурації доступних її учасників. Проте практикується використання P2P-мереж, які все ж таки мають сервери, але їх роль полягає вже не у наданні сервісів, а у підтримці інформації з приводу сервісів, що надаються клієнтами мережі.

В системі P2P автономні вузли взаємодіють з іншими автономними вузлами. Вузли є автономними в тому сенсі, що не існує загальної влади, яка може контролювати їх. В результаті автономії вузлів вони не можуть довіряти один одному та покладатися на поведінку інших вузлів, тому проблеми масштабування та надмірності стають важливішими ніж у випадку традиційної архітектури.

Сучасні P2P-мережі набули розвитку завдяки ідеям, пов'язаними з обміном інформацією, які формувалися у руслі того, що кожен вузол може

надавати й отримувати ресурси, які надаються будь-якими іншими учасниками. У випадку мережі Napster це був обмін музикою, в інших випадках це може бути надання процесорного часу для пошуку інопланетних цивілізацій (SETI@home) або ліків проти раку.

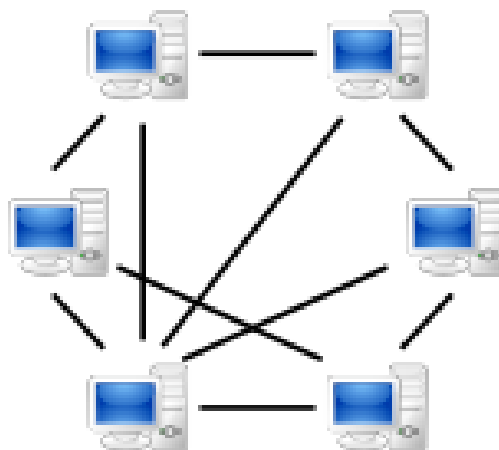


Рисунок 2.6 – мережа P2P

Переваги P2P:

1) Розподіл/зменшення вартості. Централізовані системи, які обслуговують багато клієнтів, зазвичай складають більшість вартості системи. Коли ця вартість стає дуже великою, архітектура P2P може допомогти розподілити вартість серед користувачів. Наприклад, серед систем файлообміну Napster дозволив розподілити вартість зберігання файлів і міг підтримувати індекс, потрібний для сумісного використання. Економія коштів здійснюється за допомогою використання та об'єднання ресурсів, які в іншому випадку не використовуються (наприклад SETI@home). Оскільки вузли зазвичай є автономними, важливо розподіляти витрати справедливо.

2) Об'єднання ресурсів. Децентралізований підхід веде до об'єднання ресурсів. Кожен вузол в системі P2P приносить певні ресурси, як наприклад обчислювальна потужність або пам'ять. У програмах, які потребують величезну кількість цих ресурсів, як наприклад intensive-

моделювання або розподілені файлові системи, природно використовувати P2P, щоб залучити ці ресурси. Розподілені обчислювальні системи, як наприклад SETI@Home, distributed.net, і Endeavours — очевидні приклади цього підходу. Об'єднуючи ресурси тисяч вузлів, вони можуть виконувати важкі з точки зору кількості обчислень функції. Файлобмінні системи, як наприклад Napster, Gnutella, і так далі, також об'єднують ресурси. У цих випадках це дисковий простір, щоб зберігати дані, та пропускна спроможність, щоб їх передавати, і багато інших переваг в P2P.

Класифікація P2P:

За функціями:

Розподілені обчислення. Обчислювальна проблема розподіляються на невеликі незалежні частини. Обробка кожної з частин робиться на індивідуальному ПК і результати збираються на центральному сервері. Цей центральний сервер відповідальний за розподілення елементів роботи серед окремих комп'ютерів в Інтернеті. Кожен із зареєстрованих користувачів має клієнтське програмне забезпечення. Воно користується періодами бездіяльності в ПК (часто це характеризується часами активації скрінсейверів), щоб виконувати деяке обчислення, надане сервером. Після того, як обчислення закінчене, результат посилається назад до сервера, і клієнту передається нова робота.

Файлообмін. Зберігання та обмін даними — це одна з областей, де технологія P2P була найуспішнішою. Мультимедійні дані, наприклад, вимагають великих файлів. Napster і Gnutella використовувались користувачами, щоб обійти обмеження пропускної спроможності, які роблять передачу великих файлів неприйнятними.

Співпраця. Природа технології P2P робить її добре придатною для забезпечення співпраці між користувачами. Це може бути обмін повідомленнями, онлайн-ігри, сумісна робота над документами в бізнесі, освіті та дома.

За ступенем централізації:

Чисті системи peer-to-peer. Вузли є рівними, поєднуючи ролі серверу та клієнту. Не існує центрального сервера, що керує мережею. Прикладами таких систем є Gnutella та Freenet

Гібридні системи peer-to-peer. Мають центральний сервер, що зберігає інформацію про вузли та відповідає на запити відносно цієї інформації. Вузли займаються забезпеченням ресурсами (тому що центральний сервер їх не має), повідомленням сервера про наявність цих ресурсів надання ресурсів іншим вузлам, які бажають ними скористатися.

В залежності від того, як вузли з'єднуються один з одним, можна поділити мережі на структуровані та неструктуровані:

Неструктурована мережа P2P формується, коли з'єднання встановлюються довільно. Такі мережі можуть бути легко сконструйовані, оскільки новий вузол, який хоче приєднатися до мережі, може скопіювати існуючі з'єднання іншого вузла, а вже потім почати формувати свої власні. У неструктурованій мережі P2P, якщо вузол бажає знайти певні дані в мережі, запит доведеться передати майже через всю мережу, щоб охопити так багато вузлів, як можливо. Головним недоліком таких мереж є те, що запити, можливо, не завжди вирішуються. Скоріш за все популярні дані будуть доступні в багатьох вузлів та пошук швидко знайде потрібне, але якщо вузол шукає рідкісні дані, наявні лише в декількох інших вузлів, то надзвичайно малоймовірно, що пошук буде успішним. Оскільки немає ніякої кореляції між вузлами та даними, що вони зберігають, немає ніякої гарантії, що запит знайде вузол, який має бажані дані.

Структурована мережа P2P використовує єдиний алгоритм, щоб гарантувати, що будь-який вузол може ефективно передати запит іншому вузлу, який має бажаний файл, навіть якщо файл надзвичайно рідкісний. Така гарантія потребує структуровану систему з'єднань. У наш час найпопулярнішим типом структурованої мережі P2P є розподілені хеш-таблиці, в яких хешування використовується для встановлення зв'язку між даними та конкретним вузлом, який за них відповідає[4].

## 2.5 Збір повної системи SMART-HOME

Мікроконтролер ESP32 для розробки системи розумного будинку, необхідно визначити технічні рішення та схемне рішення для розробки пристрою.

Технічні рішення можуть включати в себе вибір режиму роботи пристрою (наприклад, точка доступу Wi-Fi або режим клієнта), а також вибір додаткових модулів, наприклад, сенсорів температури або вологості, датчиків руху, мікрофонів тощо.

Схемне рішення мікроконтролеру ESP32 може бути складним та включати в себе розробку плати, на якій будуть розміщені компоненти для забезпечення функціональності пристрою наведені на таблиці 3.1.

Таблиця 2.1 Компоненти забезпечення функціональності приладу ESP32.

| Компоненти                          | Кількість | Ціна грн. за шт. |
|-------------------------------------|-----------|------------------|
| DHT(D5)                             | 1         | 80-85            |
| ESP-NOW - PTP ESP32                 | 2         | 400-500          |
| LCD 1602 (D1 – SCL Ft<br>(D2 – SDA) | 1         | 107              |
| Макетна дошка                       | 1         | 75               |
| З'єднувальні дроти                  | 7         | 1                |

Для розробки схемного рішення можна скористатися різноманітними програмними засобами, одна із таких є Arduino IDE.

Окрім того, після вибору технічних та схемних рішень, необхідно виконати програмну розробку пристрою.

Для мікроконтролера ESP32 можна використовувати різноманітні середовища програмування, такі як Arduino IDE, PlatformIO тощо.

Програмна розробка повинна забезпечувати можливість керування пристроєм та збір даних з сенсорів, а також можливість зв'язку з іншими пристроями системи розумного будинку.

Пристрій мікроконтролер ESP-NOW - RTP ESP32 для визначення його можливостей в розумному будинку та його сумісності з іншими пристроями для домашньої автоматизації.

Даний пристрій має низьку вартість та невеликий розмір, що робить його зручним для встановлення в різних місцях в будинку.

Також, він має швидкість передачі даних до 250 кбіт/с, що забезпечує швидку передачу даних між пристроями.

У разі використання даного пристрою в розумному будинку, можна забезпечити контроль за освітленням, температурою та вентиляцією приміщень, а також віддалене управління цими системами з використанням мобільних пристроїв.

ESP-NOW – RTP ESP32 є відкритим протоколом комунікації для обміну даними між двома ESP32 без точки доступу Wi-Fi або інших мережевих інфраструктур.

Цей протокол дозволяє взаємодіяти з пристроями без підключення до хмари, що дозволяє забезпечити приватність та безпеку даних.



Рисунок 2.7 – Загальний вид мікроконтролера ESP-NOW - PTP ESP32

Переваги ESP-NOW:

- Надійний зв'язок між пристроями без підключення до хмари;
- Можливість налаштування Peer-to-Peer мережі;
- Низький рівень енергоспоживання;
- Відкритий протокол комунікації.

Недоліки ESP-NOW:

- Обмежена кількість пристроїв, з якими можна підключатись до мережі;
- При збільшенні відстані між пристроями можуть виникати проблеми зі з'єднанням;
- Потребує вручну сконфігурувати всі пристрої в мережі.

Підключення датчиків та виконавчих механізмів до мікроконтролера ESP32:

Враховуючи різні фактори, такі як функціональність, вартість, можливості комунікації та інші технічні характеристики, мікроконтролер ESP-NOW - PTP ESP32 є комфортним для подальшого використання в розумному будинку, рішення підключення мікроконтролера наведено на (Рис 3.2).

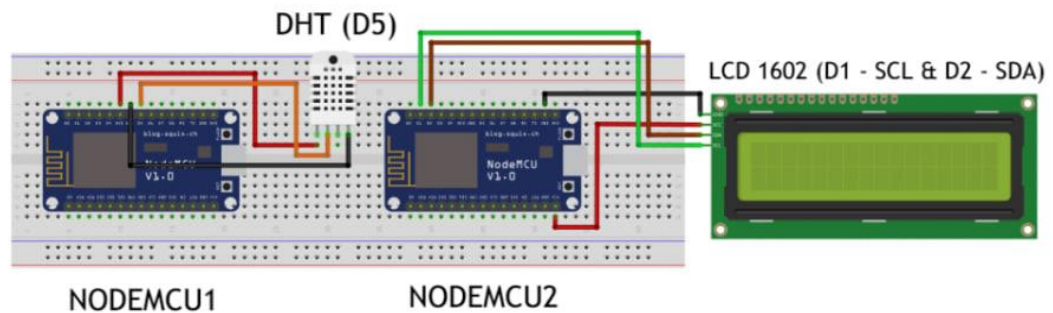


Рисунок 2.8 – Схема підключення мікроконтролера ESP-NOW - PTP ESP32

На схемі мікроконтролера ESP-NOW - PTP ESP32 зображено основні компоненти пристрою. Зокрема, на схемі видно, що мікроконтролер базується на ESP32, який має вбудований WiFi та Bluetooth модулі. Крім того, на схемі присутні інші компоненти, такі як резистори, конденсатори, індикатор стану і так далі, які необхідні для належного функціонування мікроконтролера[16].

ESP-NOW - це протокол мережі, який дозволяє обмінюватися даними між ESP32 мікроконтролерами в режимі P2P (точка-точка) без необхідності використовувати мережу WiFi. Цей протокол є дуже ефективним і дозволяє передавати дані на відстань до 200 метрів при високій швидкості передачі.

Схема мікроконтролера ESP-NOW - PTP ESP32 включає в себе все необхідне для забезпечення функціональності пристрою і обміну даними між ними.

Основні технічні характеристики ESP32 включають:

- Два 32-бітних процесори Xtensa LX6, з тактовою частотою до 240 МГц

- Інтерфейс Wi-Fi 802.11 b/g/n/e/i, що підтримує режими STA, AP та STA+AP

- Bluetooth 4.2 та Bluetooth Low Energy (BLE)

- Підтримка Ethernet, CAN та Infrared (IR)

- 34 GPIO-піни

- Два ADC-канали з роздільною здатністю 12 біт

- Два ЦАП-канали з роздільною здатністю 8 біт

- П'ять прискорювачів (Accelerometer, Gyroscope, Magnetometer, Temperature, Hall)

- Чотириканальний ШІМ та підтримка PWM до 16 каналів

- Розмір оперативної пам'яті (RAM) - 520 КБ

- Вбудована флеш-пам'ять - 4 МБ

- Підтримка різних протоколів, включаючи HTTP, MQTT, CoAP, HTTPS, WebSocket та інші

Мікроконтролер ESP32 також має високу енергоефективність та може працювати в режимі низького енергоспоживання, що робить його ідеальним вибором для реалізації проектів Інтернету речей та розумних пристроїв.

Реалізація системи комунікації на базі мікроконтролеру ESP32 з використанням компонентів, таких як DHT (D5), ESP-NOW - PTP ESP32, LCD 1602 (D1 – SCL Ft (D2 – SDA), макетної дошки та з'єднувальних дротів, дозволяє забезпечити автоматизоване керування показниками температури та вологості у приміщенні.

DHT - це цифровий датчик температури та вологості, який забезпечує вимірювання показників температури та вологості в приміщенні та передає отримані дані до мікроконтролеру ESP32.

ESP-NOW - PTP ESP32 є протоколом обміну даними між пристроями на базі мікроконтролеру ESP32. Він забезпечує бездротовий обмін даними

між різними пристроями без використання маршрутизатора та інших мережевих пристроїв. Цей протокол дозволяє передавати дані між пристроями з швидкістю до 2 Мбіт/с.

LCD 1602 (D1 – SCL Ft (D2 – SDA) - це символний дисплей, який дозволяє відображати дані на екрані. У даній реалізації він використовується для відображення показників температури та вологості у приміщенні.

Макетна дошка та з'єднувальні дроти дозволяють підключити всі компоненти до мікроконтролера ESP32 та забезпечити їхню взаємодію для реалізації автоматизованого керування показниками температури та вологості у приміщенні.

Окрім цього, для збору даних про температуру та вологість використовується датчик DHT11, який з'єднаний з мікроконтролером через вхід/вихід D5. Для виведення отриманих даних на екран використовується LCD дисплей 1602, який з'єднаний з мікроконтролером через вхід/вихід D1 (SCL) та D2 (SDA).

Загальна схема реалізації системи комунікації на базі мікроконтролеру ESP32 може виглядати наступним чином:

1. Датчик DHT11 зчитує дані про температуру та вологість повітря та передає їх до мікроконтролеру ESP32.

2. Мікроконтролер ESP32 обробляє отримані дані та передає їх за допомогою ESP-NOW - PTP ESP32 до іншого пристрою в системі розумного будинку або на сервер.

3. Отримані дані виводяться на LCD дисплей 1602 для подальшого зручного відображення користувачеві.

Зв'язок між компонентами забезпечується за допомогою з'єднувальних дротів, які з'єднують вхід/вихід кожного компонента з мікроконтролером ESP32.

Загальна схема реалізації та тестування мікроконтролера ESP32 також інших компонентів наведено на (Рис 3.3).

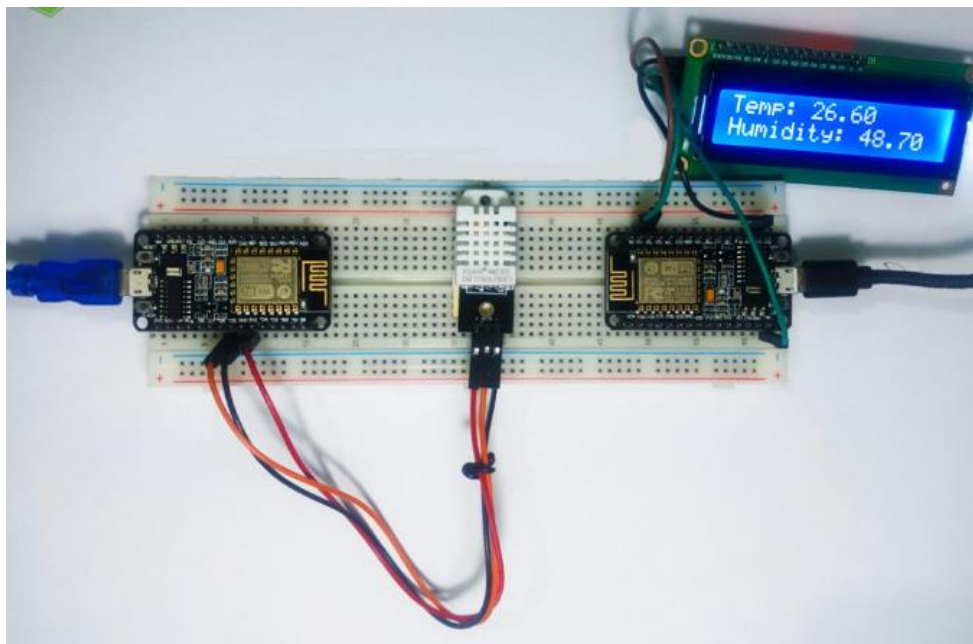


Рисунок 2.9 – Загальна схема, тестування мікроконтролера ESP32

Оцінка ефективності розробленої системи комунікації на базі мікроконтролера ESP32 може бути проведена за такими критеріями:

1. Швидкість передачі даних: Швидкість передачі даних в системі між пристроями на базі ESP32 залежить від дистанції між ними та кількості пристроїв, які беруть участь у передачі даних.

У середньому, швидкість передачі даних складає 250 кбіт/с на відстані до 400 метрів. Порівняно з іншими системами комунікації для розумного будинку, ESP32 має достатню швидкість передачі даних.

2. Стійкість до перешкод: Система ESP-NOW має вбудовану захист від перешкод, який забезпечує стійкість до впливу зовнішніх чинників, таких як шуми та перешкоди.

Це робить систему більш надійною та менш схильною до помилок порівняно з іншими системами комунікації.

3. Вартість: Вартість розробки системи на базі мікроконтролера ESP32 є відносно низькою порівняно з іншими системами комунікації. ESP32 доступний на ринку за ціною від \$5 до \$10, що робить його доступним для широкого кола користувачів.

4. Масштабованість: ESP32 має вбудований протокол мережі, який дозволяє легко масштабувати систему до більшої кількості пристроїв. Це робить систему більш гнучкою та зручною для користування.

У порівнянні з іншими системами комунікації для розумного будинку, такими як Wi-Fi, Bluetooth та Zigbee, система на базі мікроконтролера ESP32 має переваги у швидкості передачі даних та стійкості до перешкод, проте може бути менш ефективною з точки зору енергоспоживання система комунікації на базі мікроконтролера ESP32 з використанням ESP-NOW є досить ефективною, оскільки вона вимагає дуже мало енергії для передачі даних між пристроями.

ESP32 використовує низькоенергетичну мережу ESP-NOW, що дозволяє забезпечити ефективну передачу даних між пристроями при досить низькому рівні споживання енергії.

Це особливо важливо для пристроїв, які працюють в режимі очікування, оскільки вони повинні споживати дуже мало енергії, щоб продовжувати функціонувати протягом тривалого періоду часу.

У порівнянні з іншими системами комунікації для розумного будинку, такими як Wi-Fi, Bluetooth або Zigbee, ESP-NOW є енергоефективнішою системою. Wi-Fi і Bluetooth мають більш високий рівень енергоспоживання, що може призводити до швидкого розрядження батареї, особливо в пристроях з низьким рівнем споживання енергії, таких як датчики.

Зігбі також може бути менш ефективним використанням енергії, оскільки він потребує більшої кількості енергії для передачі даних на далекі відстані. Отже, система на базі мікроконтролера ESP32 з використанням ESP-NOW може бути більш енергоефективною системою комунікації для розумного будинку.

## **РОЗДІЛ 3**

### **ПРОГРАМНИЙ КОД СИСТЕМИ SMART-HOME**

#### **3.1 Програмний код мікроконтролера ESP32:**

Реалізація програмної частини буде полягати в тому, що потрібно буде написати код для передавача, до якого будуть під'єднані сенсори, та приймача, до якого буде під'єднаний дисплей. Комунікація такої системи буде відбуватись в реальному часі та мати закінчений вигляд.

Для початку розглянемо код передавача, в якому буде прописана логіка обробки даних з сенсора dht. Після первинної оброки цих даних, вони будуть відправлені на приймач для цього буде прописане певне налаштування протоколу ESP-NOW.

Насамперед, потрібно підключити бібліотеки для роботи з датчиком dht, Wi-Fi модулем та протоколом ESP-NOW.

```
#include <ESP8266WiFi.h>
#include <espnow.h>
#include <DHT.h>
```

Рисунок 3.1 – Код підключення бібліотеки

Після підключення бібліотек, потрібно створити певні змінні та структури для подальшої роботи з датчиком dht, а саме макропідстановки та змінні для визначення об'єкту та збереження даних від нього.

```
#define DHTPin D5
#define DHTType DHT22
DHT dht(DHTPin, DHTType);
float temperature, humidity;
```

Рисунок 3.2 – Код створення змінних

Таке ж саме робимо для роботи з протоколом ESP-NOW. Створюємо структуру повідомлення та масив з адресою приймача.

```

uint8_t receiverAddress[] = {0xB4, 0xE6, 0x2D, 0x4A, 0x3E, 0x24};

typedef struct struct_message {
    float temperature;
    float humidity;
} struct_message;

struct_message myData;

```

Рисунок 3.3 – Код створення структури повідомлення й масиву

Далі програма ділиться на функції `setup` та нескінчений цикл.

В функції `setup` буде відбуватись ініціалізація Wi-Fi модуля, датчика `dht` та протоколу ESP-NOW.

```

void setup() {
    Serial.begin(115200);
    delay(100);

    dht.begin();

    WiFi.mode(WIFI_STA);

    if (esp_now_init() != 0) {
        Serial.println("Error initializing ESP-NOW");
        return;
    }

    esp_now_set_self_role(ESP_NOW_ROLE_CONTROLLER);
    esp_now_add_peer(receiverAddress, ESP_NOW_ROLE_SLAVE, 1, NULL, 0);
}

```

Рисунок 3.4 – Код функції «`setup`»

Та в нескінченному циклі `loop` буде відбуватись зчитування даних з сенсорів, формування повідомлення та сама відправка цього повідомлення через протокол ESP-NOW із затримкою в 2 секунди.

```

void loop() {
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();

  myData.temperature = temperature;
  myData.humidity = humidity;

  esp_now_send(receiverAddress, (uint8_t *)&myData, sizeof(myData));

  delay(2000);
}

```

Рисунок 3.5 – Код функції «loop»

Розглянемо код приймача. Для приймача вся структура буде досить схожою. Підключення певних бібліотек, які відповідають за комунікацію з екраном та за роботу протоколу ESP-NOW.

```

#include <ESP8266WiFi.h>
#include <espnw.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

```

Рисунок 3.6 – Код підключення бібліотек для роботи з протоколом ESP-NOW

В даному випадку дисплей буде працювати по інтерфейсу I2C. Далі створимо змінні, структури для роботи з екраном та протоколом ESP-NOW.

```

LiquidCrystal_I2C lcd(0x27, 16, 2);

typedef struct struct_message {
  float temperature;
  float humidity;
} struct_message;

struct_message receivedData;

```

Рисунок 3.6 – Код для змінних для роботи з ESP-NOW

В функції `setup` буде здійснена ініціалізація дисплею, протоколу ESP-NOW та виклики функції, які будуть приймати повідомлення від передавача.

```
void setup() {
  Serial.begin(115200);
  delay(100);

  lcd.begin(16, 2); // Ініціалізація LCD екрану
  lcd.backlight(); // Увімкнення підсвічування

  WiFi.mode(WIFI_STA);

  if (esp_now_init() != 0) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }

  esp_now_set_self_role(ESP_NOW_ROLE_SLAVE);
  esp_now_register_recv_cb(OnDataReceived);
}
```

Рисунок 3.7 – Код виклику функцій приймання повідомлень від передавача

Нескінченний цикл `loop` не буде використовуватись, так як логіка роботи приймача полягає в тому, що функція, яка приймає повідомлення, буде автоматично викликатись та здійснювати подальшу логіку обробки повідомлення.

Важливим є те, що в функції `esp_now_register_recv_cb`, яка буде відповідати за приймання повідомлення, аргументом приймає callback-функцію, в якій прописано роботу з екраном. Callback-функція має назву `OnDataReceived`, котрій копіює в собі повністю текст повідомлення у відповідну структуру створено раніше та використовує методи об'єкту `lcd`, за допомогою яких здійснюється показ інформації про температуру та

ВІДНОСНУ ВОЛОГІСТЬ НА ЕКРАНІ.

```
void OnDataReceived(uint8_t *mac_addr, uint8_t *data, uint8_t data_len) {  
    memcpy(&receivedData, data, sizeof(receivedData));  
  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Temp: ");  
    lcd.print(receivedData.temperature);  
    lcd.print(" C");  
  
    lcd.setCursor(0, 1);  
    lcd.print("Humidity: ");  
    lcd.print(receivedData.humidity);  
    lcd.print(" %");  
}
```

Рисунок 3.7 – Код функції «OnDataReceived»

Таким чином, створено повніціну програмну частину для двох мікроконтролерів (передавача та приймача), які комунікують по протоколу ESP-NOW.

## ВИСНОВКИ

### Розділ 1: Огляд літератури

#### Протокол ESP-NOW

У даному розділі був розглянутий протокол ESP-NOW, його основні характеристики та принцип роботи. Було проведено аналіз його переваг та недоліків у контексті використання в системах домашньої автоматизації.

#### Використання IoT в системах

В цьому підрозділі було досліджено використання IoT (Internet of Things) у системах домашньої автоматизації. Були розглянуті принципи роботи IoT, його переваги та можливості застосування в розумних будинках.

#### Актуальність IoT

Актуальність використання IoT в системах домашньої автоматизації була обґрунтована з урахуванням сучасних тенденцій у розвитку технологій, популярності розумних пристроїв та зростання потреб споживачів у зручності, безпеці та енергоефективності.

#### Чому саме ESP-NOW...

У цьому підрозділі були представлені обґрунтування вибору протоколу ESP-NOW для реалізації системи комунікації в розумному будинку. Були розглянуті його переваги та можливості, які роблять його оптимальним вибором для даного дипломного проекту.

### Розділ 2: Апаратна частина системи Smart-Home

#### ESP32, ESP8266

У даному розділі було детально розглянуто мікроконтролери ESP32 та ESP8266, їх характеристики, функціональні можливості та особливості використання в системах домашньої автоматизації.

#### MQ7, MQ135, DHT11

В цьому підрозділі були описані датчики MQ7, MQ135 та DHT11, які використовуються

в системі Smart-Home. Були розглянуті їх принципи роботи, технічні характеристики та можливості вимірювання відповідних параметрів, таких як рівень вуглекислого газу, якість повітря та вологість.

#### Детальний опис ESP-NOW

У даному підрозділі було проведено докладний опис протоколу ESP-NOW, включаючи його архітектуру, режими роботи, методи комунікації та взаємодії з мікроконтролерами ESP32. Були надані приклади коду для налаштування та використання ESP-NOW у системі Smart-Home.

#### Peer-to-peer комунікація

У цьому підрозділі була описана концепція та переваги peer-to-peer комунікації, що використовується в системі Smart-Home з використанням ESP-NOW. Було пояснено, як пристрої можуть безпосередньо обмінюватися даними між собою, утворюючи децентралізовану мережу.

Для розробки системи було використано датчики температури та вологості, LCD-екран, з'єднувальні дроти та макетну дошку.

Описано алгоритм обміну даними між пристроями на базі мікроконтролеру ESP32 та проаналізовано ефективність нової системи порівняно з існуючими системами комунікації для розумного будинку.

Отримані результати показали, що система на базі мікроконтролеру ESP32 забезпечує надійний та швидкий зв'язок між пристроями, а також має низький рівень енергоспоживання.

Порівняно з іншими системами комунікації для розумного будинку, нова система є більш ефективною та доступною з точки зору вартості.

#### Розділ 3: Програмний код системи Smart-Home

У цьому розділі було представлено програмний код для реалізації передавача з під'єднаними сенсорами й приймача з дисплеєм, комунікація між якими буде відбуватися за протоколом ESP-NOW в реальному часі й котра матиме закінчений вигляд.

Отже, розробка та використання розумних систем у будівництві та експлуатації житлових приміщень є актуальним та перспективним напрямком розвитку технологій, що дозволяє забезпечувати більш комфортні та ефективні умови для життя людей.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Х. Альхакамі, А. Е. Саддік, «Системи домашньої автоматизації: огляд літератури», у матеріалах 2016 IEEE 14-ї міжнародної конференції з промислової інформатики (INDIN), липень 2016 р., стор. 234-239.
2. К. М. Веласкес, «Огляд розумних будинків — минуле, теперішнє та майбутнє», IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, вип. 1, стор. 3-17, січень 2018.
3. Ніл Кемерон електронні проекти на основі ESP8266 та ESP32: Створення програм та пристроїв з підтримкою Wi-Fi / пров. з англ. Ю. В. Ревіча. - М. ДМК Прес, 2022. - 456 с.: Іл. ISBN 978-5-93700-141-2
4. P2P [Електронний ресурс]. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Peer-to-peer>
5. Мікроампер [Електронний ресурс]. – Режим доступу до ресурсу: <https://uamper.com>
6. Вікіпідія [Електронний ресурс]. – Режим доступу до ресурсу: <https://learn.circuit.rocks/esp-now-the-fastest-esp8266-protocol>
7. DHT11 [Електронний ресурс]. – Режим доступу до <https://arduino.ua/ru/prod185-datchik-vlajnosti-i-temperaturi-dht11>
8. MQ7 [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.mini-tech.com.ua/datchik-gaza-mq-7-modul>
9. MQ135 [Електронний ресурс]. – Режим доступу до ресурсу: <https://arduino.ua/prod1201-modyl-datchika-kachestva-vozdyha-mq135>
10. ESP8266 [Електронний ресурс]. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/ESP8266>
11. З чого зібрати розумний будинок [Електронний ресурс]. – Режим доступу до ресурсу: <https://habr.com/ru/company/mvideo/blog/499706/>
12. Давачі [Електронний ресурс]. – Режим доступу до ресурсу: [https://supereyes.ru/articles/other/obzor\\_kontrollerov\\_apparatnoy\\_platformy\\_arduino/](https://supereyes.ru/articles/other/obzor_kontrollerov_apparatnoy_platformy_arduino/)

13. Офіційний сайт виробника - Espressif Systems: [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.espressif.com/en/products/socs/esp32/overview>

14. Документація по ESP32: [Електронний ресурс]. – Режим доступу до ресурсу: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>

15. Форум Espressif Systems: [Електронний ресурс]. – Режим доступу до ресурсу: <https://esp32.com/>

16. Документація Arduino для ESP32: [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.arduino.cc/reference/en/platforms/esp32/>