

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра електроніки, загальної та прикладної фізики

«До захисту допущено»
Завідувачка кафедри

_____Лариса ОДНОДВОРЕЦЬ
_____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 171 Електроніка освітньо-професійної програми «Електронні інформаційні системи»

на тему «**Розподілені системи моніторингу і керування на основі однорангових мереж**»

Здобувачки групи ЕП-01 _____ Лебедянської Анни Сергіївни _____

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

_____ Анна Лебедянська _____

Керівник ст.викладач, канд. ф.-м.н. Костянтин Тищенко _____

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра електроніки, загальної та прикладної фізики
Спеціальність 171 – Електроніка, освітньо-професійна програма
«Електронні інформаційні системи»

ЗАТВЕРДЖУЮ
Зав. кафедри ЕЗПФ
Л.В.Однодворець
«01» травня 2024 року

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
Лебедянської Анни Сергіївни

Тема роботи: **«РОЗПОДІЛЕНІ СИСТЕМИ МОНІТОРИНГУ І КЕРУВАННЯ НА ОСНОВІ ОДНОРАНГОВИХ МЕРЕЖ»**

затверджена наказом СумДУ від «24» квітня 2024 р., № 0417-VI

2. Термін здачі здобувачем закінченої роботи 24 травня 2024 року

3. Вихідні дані до роботи (актуальність, мета):

Мікроконторолерні інформаційні системи моніторингу та керування є основною складовою інтернету речей. Забезпечення функцій керування системою типу IoT може бути реалізована багатьма способами, один із яких є написання власного програмного забезпечення для контролера керування вузлом окремим вузлом мережі. Такий підхід дозволяє конфігурувати поведінку системи, залежно від потреб кінцевого користувача та супутніх факторів. Також використання мікроконтролерних платформ, що випускаються серійно, дозволяє уніфікувати кінцеву систему, та забезпечити сумісність (як програмну, так і апаратну) з розповсюдженими хмарними сервісами.

Мета роботи полягає в розробці схеми та програмного забезпечення вузла однорангової IoT мережі на базі платформи ESP32 для контролю параметрів датчиками фізичних величин та автоматизації окремих процесів з подальшою їх передачею до хмарного сервісу.

4. Зміст текстової частини роботи (перелік питань, які необхідно розробити):

1. Архітектура систем типу Інтернет речей
2. Однорангові мережі в системах типу Інтернет речей
3. Протоколи передачі даних в однорангових мережах
4. Мікроконтролерні платформи для IoT
5. Розробка програмного забезпечення вузла однорангової мережі.
6. Висновки.
7. Список використаних джерел.

5. Перелік графічного матеріалу для презентації:

Слайди № 1-2 – Загальна інформація

Слайди № 3-6 – Архітектура однорангових мереж, особливості реалізації розподілених систем моніторингу

Слайди № 7-9 – Мікроконтролерні платформи для побудови вузлів IoT-систем

Слайди № 10 – Методика експерименту

Слайди № 11-13 – Експериментальні результати з розробки програмного забезпечення вузла однорангової мережі

Слайд № 14 – Висновки

6. Дата видачі завдання 01.05.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Термін виконання етапів роботи	Примітка про стан вик. роботи
1.	Аналіз літературних даних	до 07.05.2024 р.	<i>вик.</i>
2.	Проведення вимірювань, моделювання, розрахунків, обробка результатів	до 22.05.2023 р.	<i>вик.</i>
3.	Оформлення тексту кваліфікаційної роботи	до 26.05.2023 р.	<i>вик.</i>
4.	Попередній захист роботи	31.05.2024 р., 10-00, онлайн	<i>вик.</i>
5.	Захист кваліфікаційної роботи	04.06.2024 р., 10-00, онлайн	

Здобувач вищої освіти

Анна ЛЕБЕДЯНСЬКА

Керівник

Костянтин ТИЩЕНКО

АНОТАЦІЯ

Кваліфікаційна робота викладена на 35 сторінках, зокрема, містить 11 рисунків, список використаних джерел із 15 найменувань.

Актуальність теми: сьогодні широкого поширення набули системи типу Інтернет речей, вони є не лише системами автоматизованого керування процесами, а й інтелектуальною мережею пристроїв, які на умовах рівних прав взаємодіють між собою, з людиною та навколишнім середовищем. Такі системи для комунікації часто використовують однорангові мережі, які дозволяють створювати надійну цілісну архітектуру в умовах динамічних змін топології мережі. Насьогодні є багато мікроконтролерних платформ, які дозволяють розробляти елементи IoT мережі, кожна із них має набір характеристик, які забезпечують її найефективніше використання у певній сфері.

Мікроконтролерна платформа ESP32 може бути використана для побудови вузла IoT мереж, оскільки вона забезпечує підключення великої кількості сенсорів навколишнього середовища та має вбудований передавач радіочастотного діапазону 2,4 ГГц.

Мета кваліфікаційної роботи бакалавра полягає в розробці схеми та програмного забезпечення вузла однорангової IoT мережі на базі платформи ESP32 для збору даних та комунікації з MQTT брокером.

Під час виконання роботи використовували онлайн середовище проєктування електронних схем Wokwi.com, яке містить віртуальні аналоги мікроконтролерів, датчиків фізичних величин та базові електронні компоненти.

У результаті проведених проєктування та розробки одержано схему та програмне забезпечення вузла однорангової мережі який реалізує комунікацію з MQTT брокером через відкриті комунікаційні протоколи, а також отримує дані з датчиків фізичних величин.

Ключові слова: інтернет речей, шлюз, мікроконтролер, ESP32, wokwi

ЗМІСТ

	с.
ВСТУП	6
РОЗДІЛ 1. ОДНОРАНГОВІ МЕРЕЖІ В СУЧАСНІЙ КОМУНІКАЦІЙНІЙ МОДЕЛІ ІНТЕРНЕТУ РЕЧЕЙ	7
1.1 Комунікаційна мережа Інтернет речей	7
1.1.1 Модель IoT мережі.....	7
1.1.2 Принцип роботи IoT	9
1.2 Однорангові мережі, як елемент Інтернету речей	10
1.2.1 Приклад архітектури мережі P2P	12
1.2.2 Типи платформ P2P IoT.....	14
1.3 Протоколи передачі даних в мережі IoT	14
1.3.1 Класична топологія передачі даних в IoT.....	15
1.3.2 Протокол прикладного рівня DDS	16
1.3.3 Відкритий протокол XMPP	17
1.3.4 Спеціалізований протокол CoAP.....	17
1.3.5 Відкритий протокол MQTT.....	18
1.4 Поширені платформи P2P для IoT	19
РОЗДІЛ 2. МЕТОДИКА І ТЕХНІКА ЕКСПЕРИМЕНТУ	22
2.1 Апаратне забезпечення для проектування вузла однорангової IoT мережі	22
2.2 Програмне забезпечення для проектування вузла однорангової IoT мережі	23
РОЗДІЛ 3. КОМПОНУВАННЯ ТА ПРОГРАМУВАННЯ КОМПОНЕНТА ОДНОРАНГОВОЇ ІОТ МЕРЕЖІ	27
ВИСНОВКИ	33
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	34

ВСТУП

Особистий комфорт у сучасному житті залежить від безлічі факторів, які значною мірою впливають на наше щоденне існування. Один із таких факторів – це використання різноманітних електронних пристроїв. Використання домашніх та портативних гаджетів не лише сприяє зручності, а й стимулює розвиток різних галузей промисловості. З кожним новим днем світ швидко розвивається, і ми все більше чуємо про інноваційні винаходи, такі як "розумний будинок" та "розумний холодильник"

У світі сучасних технологій, де вимоги до ефективності, надійності та масштабованості систем постійно зростають, розподілені системи моніторингу та керування займають ключове місце у вирішенні складних завдань автоматизації та контролю. Зокрема, однорангові мережі, що базуються на принципах рівноправності та взаємодопомоги, виявляються досить перспективним інструментом для створення таких систем.

Метою цієї роботи є дослідження, аналіз та розробка розподіленої системи моніторингу та керування на основі однорангових мереж. У рамках бакалаврської роботи розглянуто основні принципи роботи розподілених систем, а також досліджено можливості застосування однорангових мереж у контексті моніторингу та керування. Особлива увага приділена архітектурі системи, механізмам комунікації між вузлами мережі, а також алгоритмам розподіленого керування та реакції на зміни у середовищі.

Важливим етапом дослідження є аналіз існуючих підходів та технологій у цій області, а також їх порівняння з запропонованим рішенням. На основі отриманих результатів розроблено прототип вузла однорангової мережі, який може здійснювати комунікацію з MQTT брокером.

Результати роботи оприлюднені та обговорені на міжнародній конференції «Фізика, електроніка, електротехніка – 2024» [15].

РОЗДІЛ 1. ОДНОРАНГОВІ МЕРЕЖІ В СУЧАСНІЙ КОМУНІКАЦІЙНІЙ МОДЕЛІ ІНТЕРНЕТУ РЕЧЕЙ

Однорангові мережі (P2P – Peer-to-peer) сьогодні стають одним із основних засобів зв'язку між об'єктами, підключеними до мережі і пропонують унікальний підхід до комунікації та обміну даними між пристроями. Однорангові мережі відрізняються від класичної ієрархічної моделі, кожен пристрій в такій мережі виступає як рівноправний учасник, який може як приймати, так і передавати дані. Це створює гнучкість та надійність в комунікації між підключеними пристроями. Їх роль в Інтернеті речей полягає в забезпеченні ефективної комунікації та обміну даними без необхідності централізованого посередника. Це особливо важливо в контексті IoT, де велика кількість пристроїв повинна співпрацювати між собою для досягнення спільних цілей. Такий підхід знижує витрати на мережевий трафік і робить комунікацію більш незалежною та ефективною [1, 2].

Тому, однорангові мережі відіграють ключову роль в розвитку Інтернету речей, забезпечуючи гнучкість, надійність і ефективність у комунікації між підключеними пристроями. Вони відкривають нові можливості для спільної роботи та взаємодії пристроїв у сучасному цифровому світі [2].

1.1 Комунікаційна мережа Інтернет речей

1.1.1 Модель IoT мережі

Інтернет речей (IoT) – це глобальна система, що дозволяє надавати різноманітні послуги від сполучення між собою фізичних та віртуальних пристроїв за допомогою існуючих сумісних технологій зв'язку та обміну інформацією [3]

Архітектура пристроїв Інтернету речей (IoT) має чотири основні рівні: рівень датчиків, мережевий рівень, рівень обробки даних та прикладний рівень (Рис. 1.1).

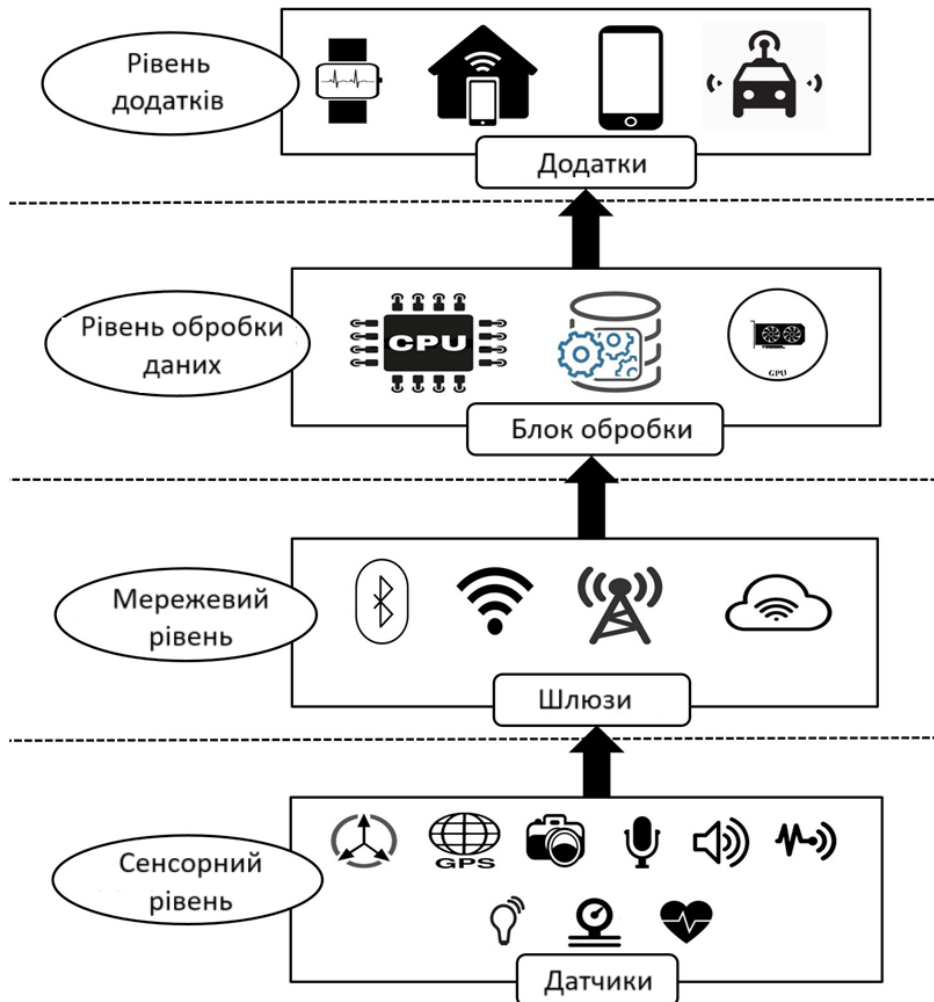


Рисунок 1.1 – Архітектура мережі IoT. Адаптовано з праці [3]

Рівень пристроїв (датчиків) в архітектурі IoT є базовим, включаючи всі підключені пристрої або "речі", які збирають і передають дані. Ці пристрої можуть включати датчики, приводи, камери та інші типи обладнання. Завданням рівня пристроїв є збір даних з фізичного середовища та перетворення їх у цифровий формат для подальшого використання на вищих рівнях.

Мережевий рівень відповідає за передачу даних між пристроями та хмаровими сервісами. Він включає різноманітні протоколи, такі як Wi-Fi, Bluetooth, Zigbee та інші, які забезпечують зв'язок між пристроями. Мережевий рівень забезпечує безпечну та ефективну передачу даних.

Рівень проміжного програмного забезпечення діє як посередник між рівнем пристроїв та прикладним рівнем. Він відповідає за керування та обробку даних, гарантуючи їхню ефективну доставку до відповідних програм. Рівень проміжного ПЗ може включати обробку, зберігання та аналіз даних, а також забезпечувати безпеку та інші сервіси.

Рівень додатків є найвищим рівнем в архітектурі IoT і відповідає за надання послуг і додатків для користувачів. Сюди входять інформаційні панелі, аналітичні інструменти та інші програми, які дозволяють користувачам взаємодіяти з підключеними пристроями. Рівень додатків відіграє ключову роль у розумінні та використанні даних IoT. [3, 4]

1.1.2 Принцип роботи IoT

Робота мережі Інтернет речей зазвичай ділиться на чотири етапи [5]:

1. Зчитування інформації за допомогою датчиків;
2. Передача даних від датчиків до хмарних сховищ;
3. Обробка даних отриманих за допомогою датчиків;
4. Передача інформації на інтерфейс користувача.

Розглянемо детальніше кожен із цих етапів.

По-перше, датчики або пристрої збирають дані з навколишнього середовища. Це може бути як простий процес, такий як вимірювання температури, або складніший, як запис відео за допомогою камери відеоспостереження. На цьому етапі також можуть використовуватися пристрої, оскільки декілька датчиків можуть бути об'єднані разом або датчики

можуть бути частиною пристрою, який виконує більше, ніж просто аналіз даних.

Датчики можуть бути підключені до хмари за допомогою різних методів, включаючи стільникову мережу, супутникову мережу, Wi-Fi, Bluetooth, малопотужні широкосмугові мережі (LPWAN), або з'єднання безпосередньо з Інтернетом через Ethernet. Кожен варіант підключення має свої компроміси між споживанням енергії, діапазоном та пропускнуою здатністю, але всі вони мають одну мету - передачу даних до хмарового сховища.

Після того, як дані потрапляють до хмари, програмне забезпечення виконує певну обробку цих даних. Це може включати перевірку того, що зчитувана температура або інші параметри знаходяться в межах допустимого діапазону, або може бути обробкою інформації, наприклад, за допомогою комп'ютерного програмного забезпечення для ідентифікації об'єктів на відео (наприклад, виявлення зловмисників у вашому будинку) [5, 6].

На останньому етапі корисна інформація передається кінцевому користувачеві. Це може бути через сповіщення (електронна пошта, текстові повідомлення тощо). Наприклад, текстове сповіщення може надсилатися у разі виконання певних умов - тригерів. Крім того, користувач може мати інтерфейс, який дозволяє йому активно моніторити систему, наприклад, через мобільну програму або веб-сторінку. Також можуть бути виконані автоматичні дії, наприклад, автоматичне регулювання температури або автоматичне сповіщення екстреним службам [6].

1.2 Однорангові мережі, як елемент Інтернету речей

Платформи P2P IoT – однорангові платформи Інтернету речей, забезпечують пряме спілкування та обмін даними між пристроями IoT без централізованого сервера. Ці платформи використовують потужність розподілених мереж для забезпечення безперебійного підключення та

співпраці між пристроями IoT. Проте не кожна платформа IoT має можливості P2P зв'язку [1, 2].

Архітектура P2P – це комп'ютерна мережа, де кожен комп'ютер має однакові можливості та обов'язки (Рис. 1.2). Це відрізняється від традиційної архітектури клієнт-сервер, де одні комп'ютери обслуговують інші.

У однорангових мережах програмне забезпечення може діяти як клієнт і сервер. Ця архітектура зазвичай використовується для розповсюдження програмного забезпечення, потокового мультимедіа та обміну файлами. Він також має застосування в науці, мережах, пошуку та комунікаційних мережах.

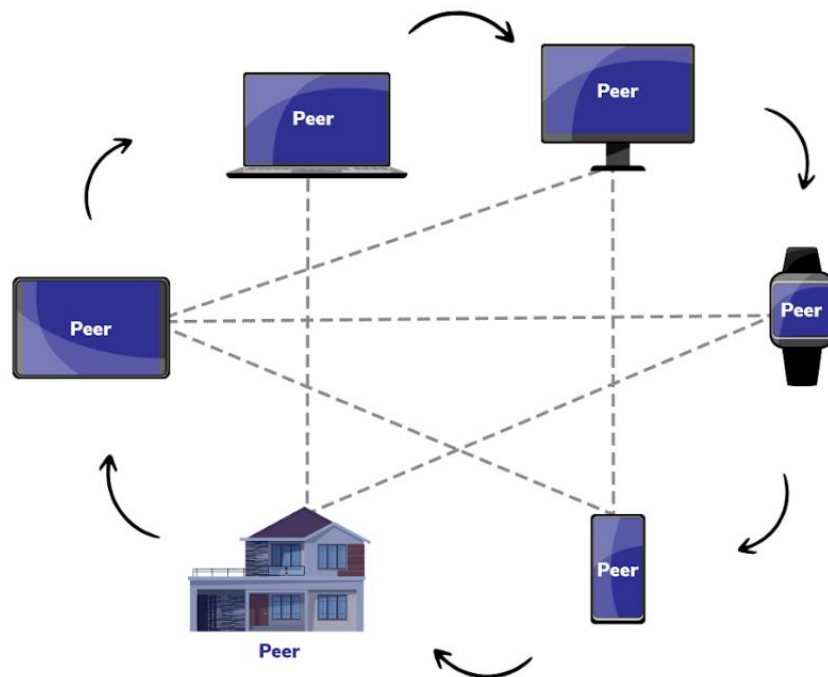


Рисунок 1.2 – Архітектура мережі P2P [1]

У структурованому P2P робочі станції та ресурси організовані на основі певних критеріїв і алгоритмів. Якщо порівнювати P2P з архітектурою клієнт/сервер, P2P-мережі мають клієнти з такими ресурсами, як пропускна здатність, простір для зберігання та потужність обробки. Оскільки до мережі приєднується більше комп'ютерів, загальна пропускна здатність зростає.

Це робить мережі P2P більш захищеними від атак. У мережах клієнт-сервер ресурси розподіляються між клієнтами. Отже, чим більше клієнтів приєднується, тим менше ресурсів стає доступним.

Вибір правильної платформи для проектів IoT має вирішальне значення, оскільки це може значно вплинути на успіх і ефективність розгортання. Платформа має відповідати конкретним вимогам проекту та пропонувати функції та можливості, які оптимізують зв'язок пристроїв, керування даними та загальну продуктивність системи. Правильно підібрана платформа може спростити процеси розробки, інтеграції та обслуговування, що призведе до економії коштів і прискореного виходу на ринок.

Платформи P2P IoT працюють шляхом встановлення прямих каналів зв'язку між пристроями IoT. Замість того, щоб покладатися на центральний сервер для маршрутизації даних і керування взаємодією, пристрої підключаються безпосередньо, утворюючи розподілену мережу [2].

Протоколи та алгоритми P2P дозволяють пристроям виявляти сусідні однорангові пристрої та підключатися до них, обмінюватися даними та координувати дії. Цей децентралізований підхід покращує масштабованість, стійкість і оперативність систем IoT.

1.2.1 Приклад архітектури мережі P2P

Розглянемо концепцію архітектури мережі P2P на прикладі організації робочої місця, де комп'ютери підключаються до робочої групи, щоб ділитися файлами, мати доступ до Інтернету та спільно використовувати принтери. На рисунку 1.3 показано принцип взаємодії пристроїв у мережі з серверною (а) та одноранговою (б) архітектурою [4, 6].

Перша архітектура є класичною, взаємодія між кожними окремими пристроями у ній чітко регламентується налаштуваннями сервера, також будь-які дані передаються через нього, як посередника. За такого підходу виникає надмірне навантаження (у моменти інтенсивної передачі трафіку) на

центральний вузол, і як наслідок продуктивність такої системи падає, а якщо описати цей процес більш абстрактно, то пропускна здатність каналів зв'язку в мережах із сервером буде зворотно пропорційна інтенсивності передачі даних у мережі.



Рисунок 1.3 – Організація взаємодії пристроїв у мережі з серверною (а) та одноранговою (б) архітектурою. Адаптовано з праці [6]

У мережевій архітектурі P2P взаємодія відбувається інакшим чином, і продуктивність мережі набагато менше залежить від інтенсивності трафіку оскільки кожен вузол у мережі має однакові обов'язки та можливості. Вони всі можуть виконувати однакові завдання та допомагати один одному. Кожен мережевий пристрій діє як клієнт і сервер. Це означає, що вони можуть попросити речі з інших комп'ютерів і щось їм дати.

Така архітектура гарно працює в обмежених мережевих екосистемах, наприклад будинках, офісах або невеликих компаніях, де кожен пристрій працює незалежно і не покладається на сервер, як пристрій керування взаємодією елементів мережі. Також кожен вузол у мережі може ділитися своїми даними з іншими, що полегшує комунікацію та обмін даними.

1.2.2 Типи платформ P2P IoT

В мережах типу Інтернет речей існує два підходи до створення комунікаційних платформ, а саме чисті та гібридні платформи P2P IoT.

Чисті платформи P2P покладаються виключно на одноранговий зв'язок без участі центрального сервера. Пристрої підключаються безпосередньо один до одного й автономно обробляють усі аспекти зв'язку, керування даними та координацію. Вони забезпечують максимальну децентралізацію, підвищену конфіденційність і підвищену відмовостійкість. Тим не менш, для ефективного керування мережею їм можуть знадобитися більш складні алгоритми та протоколи.

Гібридні P2P-платформи поєднують як однорангову, так і серверну архітектуру. Хоча пристрої можуть безпосередньо спілкуватися один з одним, може існувати центральний сервер, який допомагає виконувати певні завдання, такі як виявлення пристрою, автентифікація або зберігання даних.

Гібридні платформи збалансовують децентралізацію та централізований контроль, забезпечуючи гнучкість і масштабованість.

1.3 Протоколи передачі даних в мережі IoT

Більшість сервісів Інтернету речей ґрунтуються на принципі обробки інформації від безлічі вузлів, що суттєво відрізняється від архітектур, що використовуються в класичних мережах, таких як абонент-вузол для телефонії або клієнт-сервер для передачі даних. Це приводить до виникнення нової архітектури, де існує багато джерел і багато одержувачів, при цьому обсяг трафіку від сенсорного вузла може коливатися від дуже малих до дуже великих значень. Звичні прикладні протоколи для передачі повідомлень не завжди придатні для використання у таких системах. Розглянемо найбільш популярні протоколи передачі даних для IoT [7, 8].

1.3.1 Класична топологія передачі даних в IoT

Показана на рисунку 1.4 топологія відповідає шаблону проектування передачі повідомлень, відомому як "видавець-підписник" (Publisher-Subscriber). У такій схемі вводяться поняття видавця – джерела інформації та передплатника – одержувача інформації. Термін «підписка» стосується операції, за допомогою якої передплатник отримує інформацію від певного видавця, а також упорядкування збору інформації – параметрів періодичності отримання та інших відповідних показників (в залежності від конкретної реалізації).

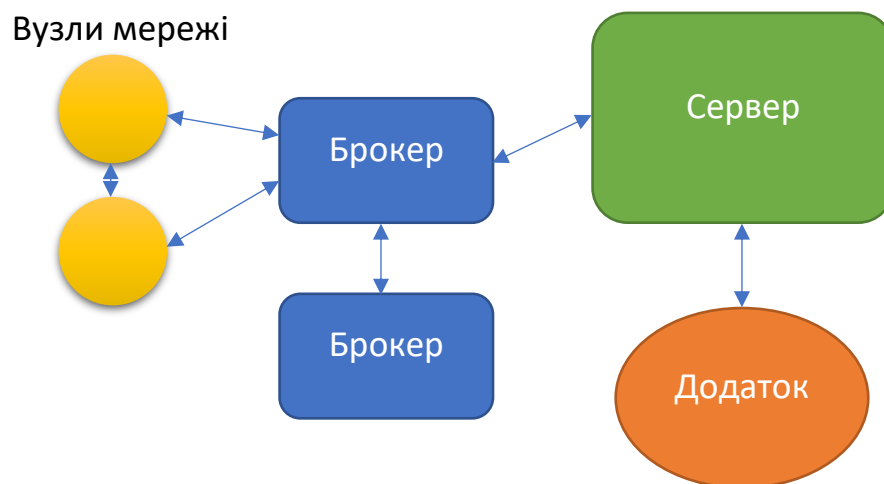


Рисунок 1.4 – Топологія системи передачі даних між вузлами мережі IoT. Адаптовано з праці [7]

За такої конфігурації сенсорний вузол (Node) збирає інформацію від різних датчиків (наприклад, дані про температуру середовища) і надсилає її відповідно до умов передплати або за запитом, також дані можуть бути надіслані автоматично через певний проміжок часу. Зазвичай самі датчики є досить простими і просто надсилають інформацію про вимірювані параметри. Тому потрібно об'єднувати датчики у вузли, обладнані мікроконтролерами, які відповідають за зчитування даних і передачу їх на сервер за певними алгоритмами.

Така топологія також передбачає використання брокера – сервера, який приймає інформацію від видавців і передає її відповідним передплатникам. У складних системах брокер може виконувати різні операції, такі як аналіз та обробка надісланих даних, а також встановлювати пріоритети з'єднання та формувати черги для передачі повідомлень.

1.3.2 Протокол прикладного рівня DDS

DDS (Data Distribution Service) – протокол прикладного рівня для систем реального часу, що ґрунтується на моделі "видавець-передплатник". Основна мета протоколу полягає в тому, щоб забезпечити зв'язок між пристроями за допомогою шини обміну повідомленнями (рис. 1.5).

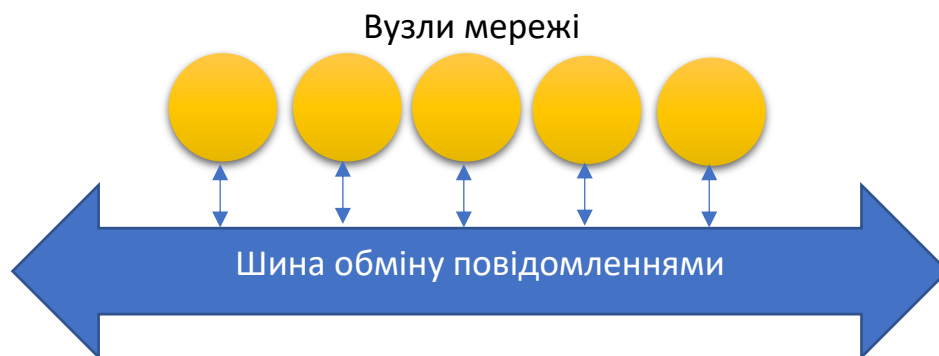


Рисунок 1.5 – З'єднання пристроїв за допомогою протоколу DDS в IoT.

Адаптовано з праці [8]

Прискорене функціонування є ключовою особливістю пристроїв в реальному часі. Часові рамки часто вимірюються в долях мікросекунд, що вимагає складних шляхів зв'язку. Прості TCP потоки даних обмежуються у таких умовах. DDS, натомість, надає деталізований контроль якості обслуговування, багатоадресну передачу та повну масштабованість. Також він відрізняється високим рівнем розгалуження даних та забезпечує потужні методи фільтрації та відбору даних за адресами призначення, здатних обслуговувати тисячі синхронних одержувачів даних.

Для обробки даних від пристроїв зіркоподібні мережі не підходять. DDS натомість реалізує прямий шинний зв'язок між пристроями, базуючись на реляційній моделі даних. Цю мережу також називають шиною даних (DataBus), оскільки вона функціонує аналогічно базі даних, але на мережевому рівні. Так само, як база даних керує доступом до збережених даних, шина даних керує доступом до даних та їх оновленнями для багатьох користувачів одночасно. Це особливо важливо для високопродуктивних пристроїв, які мають працювати як єдина система [8].

1.3.3 Відкритий протокол XMPP

XMPP (eXtensible Messaging and Presence Protocol) - це відкритий протокол, побудований на основі XML, призначений для обміну миттєвими повідомленнями та інформацією про присутність, який працює в режимі, наближеному до реального часу. Протокол використовує текстовий формат як вбудований тип, що забезпечує природний спосіб комунікації.

XMPP надає зручний метод адресації пристроїв, особливо коли дані передаються між віддаленими, незалежними точками, наприклад, у випадку зв'язку між абонентами. Проте цей протокол не характеризується високою швидкістю [7, 8].

1.3.4 Спеціалізований протокол CoAP

CoAP (Constrained Application Protocol) (рис. 1.6) – спеціалізований протокол передачі даних для використання у мережах і пристроях з обмеженими ресурсами та використовує UDP як транспортний протокол.

Протокол CoAP використовує невелику кількість типів повідомлень, більшість з яких є запитами-відповідями: GET (отримати інформацію про ресурс), PUT (задати нові дані для ресурсу), POST (здійснити зміни в ресурсі), DELETE (видалити ресурс), CONNECT (встановити з'єднання). Клієнти

(користувацькі додатки) використовують ці повідомлення для управління та спостереження за ресурсами. При запиті встановлюється прапор спостереження, і сервер продовжує надсилати оновлення після передачі початкового повідомлення. Це дозволяє серверам організувати потік змін стану датчиків [7, 8].



Рисунок 1.6 – Елемент мережі де використовується протокол CoAP та XMPP. Адаптовано з праці [8]

Отже, на ділянці мережі між сенсорним вузлом і брокером, де забезпечується їх взаємодія для реєстрації, конфігурації та передачі інформації, найчастіше використовуються два протоколи - XMPP і CoAP. Вибір конкретного протоколу залежить від умов, які присутні у мережі та вимог до функціональності системи.

1.3.5 Відкритий протокол MQTT

MQTT (Message Queue Telemetry Transport) - це легкий, компактний і відкритий протокол обміну даними, розроблений для передачі інформації на віддалених локаціях, де обмежені ресурси, такі як обмежена пропускна здатність каналу та потреба в мінімальному розмірі коду. Ці характеристики роблять його ідеальним для використання в системах M2M (машина-машина).

Крім того, існує версія протоколу MQTT-SN (MQTT для мереж датчиків), раніше відома як MQTT-S, яка спеціально призначена для вбудованих бездротових пристроїв, що не підтримують TCP/IP мережі [7–9].

Спрощений процес роботи протоколу MQTT (рис. 1.7) має наступний алгоритм:

Видавець передає повідомлення з певними даними (наприклад, інформацію з датчиків вологості) на брокера, вказуючи при цьому тему (Торіс), до якої ці дані відносяться (наприклад, "вологість"). **Брокер** аналізує, які із передплатників мають підписку на певні теми, в даному випадку – на тему "вологість". **Передплатникам**, які підписані на тему "вологість", брокером буде відправлено повідомлення з інформацією від датчиків вологості.

Таким чином, безліч передплатників можуть бути підписані на різноманітні теми і в залежності від цих підписок отримувати необхідну їм інформацію, не спілкуючись з видавцем безпосередньо [9].

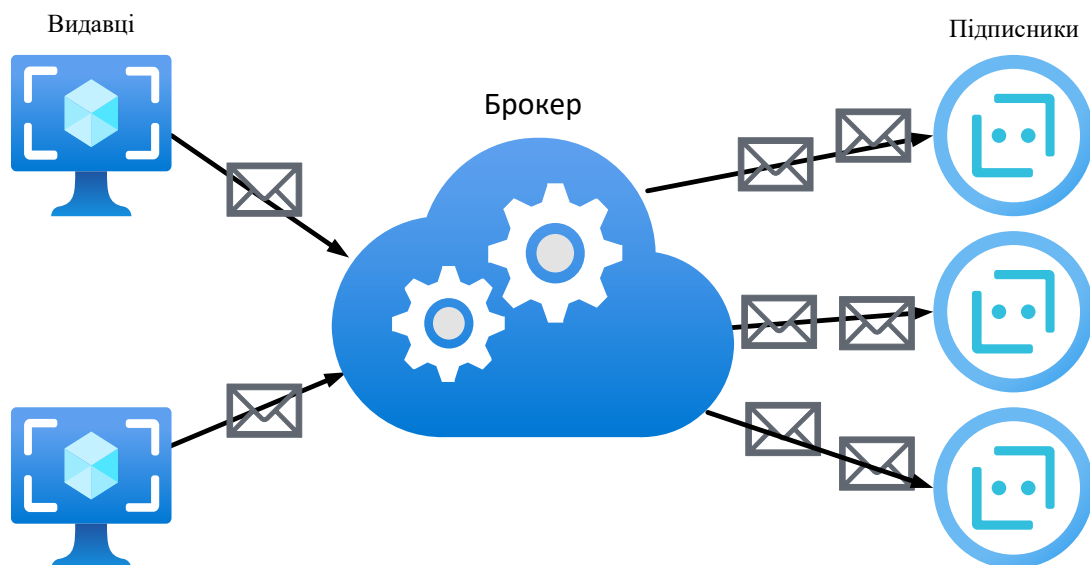


Рисунок 1.7 – Принцип роботи протоколу MQTT. Адаптовано з праці [9]

1.4 Поширені платформи P2P для IoT

Платформи P2P IoT – це інноваційні рішення, які дозволяють пристроям спілкуватися напряду, не покладаючись на центральний сервер. Вибір правильної платформи має вирішальне значення для проектів IoT для забезпечення оптимальної продуктивності та ефективності. Доступні гібридні

та чисті платформи P2P, кожна з яких має свої переваги та обмеження. Розглянемо деякі платформи IoT, які використовують архітектуру P2P для зв'язку між пристроями.

ІОТА – це технологія розподіленої книги (DLT) для Інтернету речей. Вона використовує унікальну архітектуру P2P під назвою Tangle, яка формує спрямований ациклічний граф (DAG) та забезпечує прямий обмін даними і цінностями між пристроями IoT. Ця технологія підходить для мікротранзакцій і міжмашинного зв'язку (M2M).

Filament надає децентралізовану платформу IoT, зосереджену на безпечному та автономному спілкуванні. Він використовує протоколи P2P і технологію під назвою Blocklet для прямого та швидкого підключення між пристроями, що дозволяє пристроям IoT створювати та керувати безпечними ідентифікаторами, сприяючи автономному та стійкому зв'язку. Платформа включає в себе технологію блокчейн для незмінності даних і довіри.

Slock.it поєднує технологію блокчейну та IoT для безпечних і автономних транзакцій і спільного використання ресурсів. Slock.it використовує блокчейн Ethereum і смарт-контракти для забезпечення прямої взаємодії між пристроями та дозволяє користувачам безпечно й автономно орендувати або ділитися доступом до фізичних активів через підключені пристрої а також виключає потребу в посередниках.

Ці платформи використовують архітектуру P2P для прямого зв'язку між пристроями, забезпечуючи безпечну, децентралізовану та автономну взаємодію для додатків IoT. Незважаючи на популярність у деяких випадках використання, платформи P2P IoT мають деякі труднощі, які необхідно вирішити. Ось деякі з них:

–платформам P2P IoT потрібно керувати багатьма пристроями. Це може бути важко, оскільки кількість пристроїв зростає, що ускладнює координацію їх взаємодії.

–безпека є великою проблемою для платформ P2P IoT. Важливо переконатися, що пристрої безпечно спілкуються та обмінюються даними, захищаючи від несанкціонованого доступу та зловмисних дій.

–платформи P2P IoT покладаються на підключені пристрої для спілкування та обміну даними. Але буває важко підтримувати надійне з'єднання, особливо в умовах динамічної зміни топології мережі.

–пристрої IoT, як-от обчислювальна потужність, пам'ять і термін служби акумулятора, часто мають обмежені можливості.

–розробка ефективних способів виявлення потрібних пристроїв і підключення до них може бути складною.

–платформи P2P IoT можуть мати різні пристрої, які використовують різні протоколи та стандарти. Важливо переконатися, що ці пристрої можуть спілкуватися один з одним і легко обмінюватися даними.

РОЗДІЛ 2 МЕТОДИКА І ТЕХНІКА ЕКСПЕРИМЕНТУ

2.1 Апаратне забезпечення для проектування вузла однорангової IoT мережі

Однорангові мережі застосовуються у комунікації коли апаратні можливості вузлів IoT мережі, а саме обчислювальна потужність, потужність передатчиків і приймачів та ін. обмежені в наслідок особливостей архітектури чи енергетичних режимів роботи. Загалом при проектуванні елементів однорангової мережі слід опиратись на частотні діапазони роботи та інтенсивність обміну даними між вузлами. Немало важливим є врахування динамічних факторів, як то зміна топології мережі, перерозподіл трафіку між вузлами, раптові перешкоди в робочому діапазоні частот та ін. [11].

Як платформу для розробки вузла однорангової мережі було обрано апаратну платформу, що базується на мікроконтролері ESP-32 (рис. 2.1).

ESP32 – мікроконтролер, розроблений компанією Espressif Systems, який має вбудований модуль Wi-Fi та Bluetooth, що робить його ідеальним вибором для проектів Інтернету речей (IoT). Він має двоядерний процесор Xtensa LX6, що базується на архітектурі Tensilica. Однак його основна перевага полягає в можливості безпосереднього підключення до бездротових мереж Wi-Fi та взаємодії з іншими пристроями через Bluetooth.

Однією з важливих особливостей ESP32 є велика кількість виводів введення/виведення (I/O), які дозволяють підключати різноманітні датчики та пристрої для збору даних. Завдяки двом ядрам мікроконтролера та можливості роботи на високих частотах, ESP32 здатний ефективно обробляти обчислення та взаємодіяти з іншими пристроями у реальному часі [11, 12].

Інтеграція Wi-Fi дозволяє ESP32 легко підключатися до Інтернету, надсилати дані на сервери або отримувати інформацію з хмарних джерел, що

робить його ідеальним для розробки проектів IoT, які потребують бездротового зв'язку.

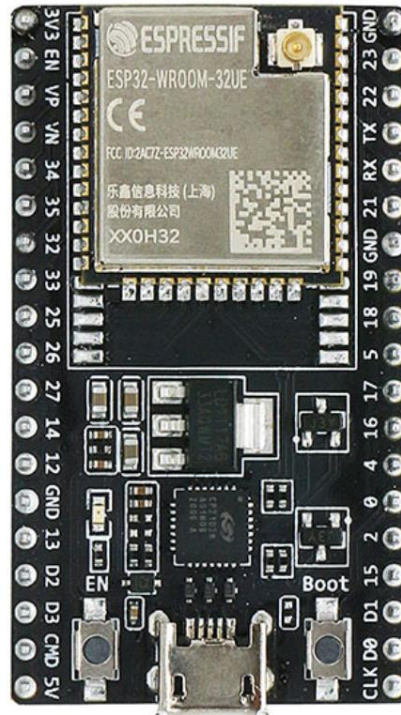


Рисунок 2.1 – Зовнішній вигляд контролера на базі ESP32 [12]

Цей мікроконтролер підтримує різні інтерфейси, такі як I2C, SPI, UART, GPIO, PWM, що дозволяє підключати до нього різноманітні сенсори та пристрої. Застосування ESP32 може бути різноманітним, включаючи створення вузлів IoT для контролю умов у виробничих приміщеннях та надсилання даних до центральних систем, а також інтеграцію IoT-функціональності у системи інтелектуального керування [11, 12].

2.2 Програмне забезпечення для проектування вузла однорангової IoT мережі

Як програмне середовище для розробки вузла однорангової мережі було обрано онлайн середовище автоматизованого проектування <https://wokwi.com/>, що має підтримку контролера ESP32 і дозволяє в режимі реального часу моделювати роботу електричних схем [13, 14].

Wokwi (рис. 2.2), як інноваційний онлайн-інструмент для моделювання та проектування, вносить значний внесок у сферу освіти та хобі в галузі електроніки та програмування. Цей ресурс відрізняється своєю здатністю точно моделювати електронні схеми та програмні коди, що робить його цінним інструментом для студентів, викладачів та ентузіастів електроніки.

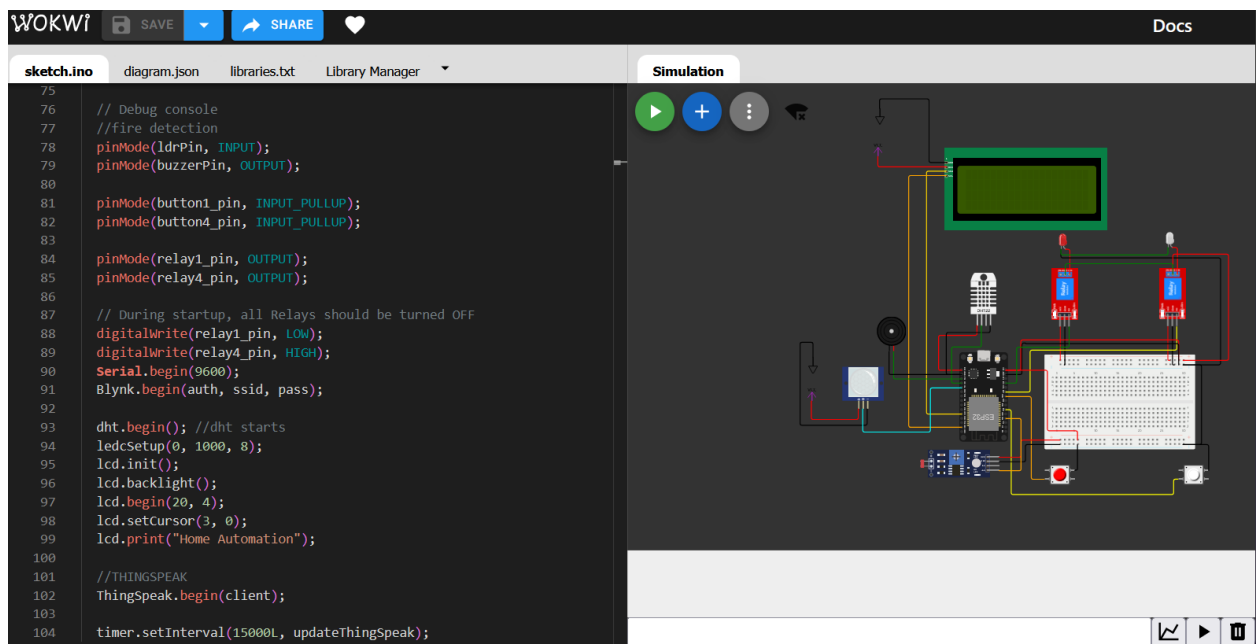


Рисунок 2.2 – Вікно програми Wokwi [13]

Wokwi дозволяє інтегрувати різноманітні компоненти, такі як датчики, реле та інші, в емульовані схеми, що спрощує процес розробки. Основною перевагою є гнучкість у виборі мікроконтролера для симуляції, включаючи такі платформи, як Arduino, ESP32, Raspberry Pi Pico. Користувачі можуть моделювати різні сценарії та експериментувати з різними мікроконтролерами, визначаючи найбільш ефективні рішення для своїх проектів [13].

Особливістю Wokwi є його здатність ефективно моделювати та тестувати мікроконтролери та їх взаємодію з різними електронними компонентами. Це дозволяє користувачам проводити віртуальні експерименти та випробування, маючи можливість візуально прослідкувати поведінку схеми та виводів мікроконтролерів у реальному часі.

Wokwi надає інтуїтивно зрозумілі інструменти для створення та редагування електронних схем, сприяючи легкому засвоєнню основ електроніки та мікроконтролерних систем. Користувачі мають доступ до різноманітних компонентів та модулів, що дозволяє їм експериментувати з широким спектром проектів, від базових до складніших.

Серед основних переваг симулятора Wokwi можна виділити наступні: Форматування та автозаповнення коду; Демонстраційні приклади стандартної бібліотеки; Можливість створення власних компонентів і бібліотек; Функції моніторингу LED і FPS; Робота проекту у мобільних браузерях; Орієнтація на реальні умови - наприклад, функція відскоку кнопки; Підтримка низькорівневого програмування, наприклад асемблерних інструкцій.

Крім того, Wokwi дозволяє програмувати мікроконтролери за допомогою популярних мов програмування, таких як C++ та Python. Це дозволяє розробляти, тестувати та вдосконалювати програмні коди у віртуальному середовищі, що значно підвищує ефективність навчання та розробки [13, 14].

Додатково, Wokwi створює сприятливе середовище для спільної роботи та обміну ідеями. За допомогою цього інструменту користувачі можуть ділитися своїми проектами, обговорювати рішення та надавати поради один одному, що сприяє колективному розвитку в галузі електроніки.

Здатність Wokwi до віртуального моделювання та програмування робить його цінним інструментом для експериментування та вивчення, хоча його обмеження у комплексності можуть вимагати додаткового використання фізичних компонентів для більш глибокого розуміння електронних систем.

Варто відзначити обмежені можливості реалістичності Wokwi. Хоча він емулює електронні схеми, йому бракує повноцінності у передачі фізичних аспектів проектів, особливо у складних сценаріях. Наприклад, можуть виникнути випадки, коли певні аспекти реального проекту, такі як електромагнітні взаємодії або точна поведінка компонентів у різних умовах, не відтворюються достатньо точно. Це може призвести до невідповідності між прогнозованим поведінкою схеми та реальним результатом.

Крім того, обмежена бібліотека компонентів також може стати ще однією перешкодою. Навіть при можливості додавати власні компоненти, відсутність повноцінної бібліотеки може викликати дискомфорт у роботі з проектами, особливо якщо необхідні компоненти відсутні або не підтримуються. Це може обмежувати можливості розробників у створенні більш складних і реалістичних проектів [13, 14].

РОЗДІЛ 3. КОМПОНУВАННЯ ТА ПРОГРАМУВАННЯ КОМПОНЕНТА ОДНОРАНГОВОЇ ІОТ МЕРЕЖІ

В рамках експериментальної частини роботи було створено прототип вузла однорангової мережі у рамках концепції Інтернет речей. Концепція розробленої системи полягає у тому, що розроблюваний пристрій буде комунікувати з іншими за посередництва MQTT брокера, тобто використовується гібридна модель однорангової мережі.

Розроблення пристрою проводиться в онлайн середовищі моделювання та програмування Wokwi.com, яке дозволяє в режимі реального часу моделювати роботу електронної схеми, змінювати вхідні параметри датчиків і, найважливіше, підключатись до мережі. Схема у Wokwi, яка приєднана до мережі, може комунікувати із зовнішніми додатками, що робить її повноцінним вузлом однорангової мережі. У нашому випадку роль додатка виконує MQTT брокер, який за своєю суттю є хмарним сервісом перенаправлення потоків даних між окремими вузлами.

Першим етапом створення вузла однорангової мережі є розміщення на робочому полі Wokwi.com електронних компонентів (Рис. 3.1). Як керуючий пристрій було обрано платформу розробника на основі контролера ESP32, також схема містить датчик температури і вологості DHT-22, інфрачервоний датчик руху, фоторезистор, дві кнопки керування та два реле, до яких може бути підключене силове навантаження. Для візуалізації параметрів на локальному пристрої використовується символічний екран розмірністю 4 стрічки по 20 символів у кожній, чого цілком достатньо для відображення всіх вимірюваних параметрів та службової інформації. Далі всі елементи необхідно об'єднати таким чином, щоб всі вони мали спільне живлення, а сигнальні дроти кожного із датчиків були приєднані до відповідних входів контролера. Також слід врахувати, що для підключення дисплея використовується апаратний протокол I2C, і зарезервовані під нього пінні контролера не можуть бути використані для підключення інших пристроїв.

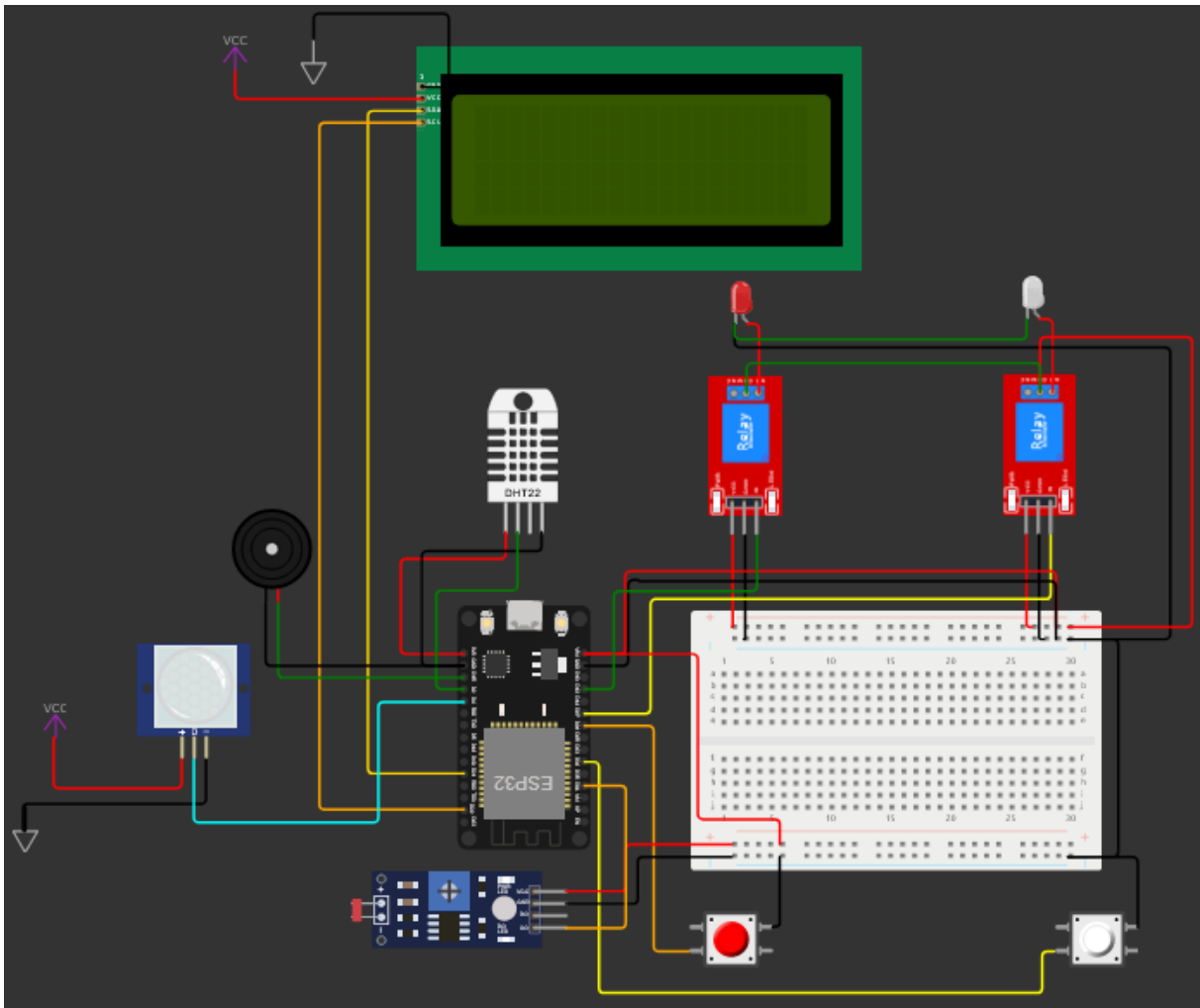


Рисунок 3.1 – Схема вузла однорангової мережі в середовищі Wokwi.com

Середовище Wokwi.com дозволяє виконувати програмування контролера безпосередньо у вікні браузера, без застосування спеціалізованих інтегрованих середовищ розробника. Як мова програмування підтримуються Python, C/C++, Asembler та інші. Нами для розробки було обрано мову C++ з підключенням спеціалізованих бібліотек для мікроконтролерів. Саме підключення бібліотек необхідне для того, щоб компілятор міг обробити спеціальні функції, відсутні у базовій версії мови програмування, цей процес проілюстровано у лістингу 3.1.

Лістинг 3.1 – Підключення бібліотек до проекту

```
#include <WiFi.h>
#include <WiFiClient.h>
```

```
#include <BlynkSimpleEsp32.h>
#include "DHT.h"
#include "ThingSpeak.h"
#include <LiquidCrystal_I2C.h>
```

Програмний код додатку більш структурований і зрозумілий, коли робочі параметри задані у вигляді змінних, особливо актуально це при підключенні пристроїв до портів введення/виведення контролера, оскільки за замовчуванням вони мають цифрову нумерацію, а структурування задається способом найменування цих портів. Таку операцію можна здійснити двома способами, а саме використанням констант або ж директиви `#define`. Ця операція показана у лістингу 3.2.

Лістинг 3.2 – Задання констант та наперед визначених значень

```
#define ldrPin 34
#define buzzerPin 15
#define button1_pin 26
#define button4_pin 32
#define relay1_pin 12
#define relay4_pin 27
int relay1_state = 0;
int relay4_state = 0;
#define button1_vpin V1
#define button4_vpin V4
#define DHTPIN 2
```

Розроблюваний пристрій для комунікації використовує MQTT брокера, як посередника. За своєю суттю він є хмарним сервісом з окремим програмним кодом, та у нашому проєкті використовується сервіс Blynk, який можна налаштовувати під власні проєкти. Лістинг 3.3 описує частину коду програми, яка відповідає за налаштування брокера та підключення до нього.

Лістинг 3.3 – Налаштування MQTT брокера

```
BlynkTimer timer;
const int myChannelNumber = 2383821;
```

```

const char* myApiKey = "09308S2L7TZVR14J";
const char* server = "api.thingspeak.com";
WiFiClient client;

BLYNK_CONNECTED() {
  Blynk.syncVirtual(button1_vpin);
  Blynk.syncVirtual(button4_vpin);
}

BLYNK_WRITE(button4_vpin) {
  relay4_state = param.asInt();
  digitalWrite(relay4_pin, relay4_state);
}

```

За передавання даних від вузла однорангової мережі до брокера відповідає частина програмного коду, що показана в лістингу 3.4. Цей код оформлений у вигляді окремої функції і, у разі необхідності, може бути викликаний у довільний момент при виконанні основної програми контролера. У коді функції по чергово виконуються операції зчитування даних із фізичних датчиків, після чого вони передаються мережею до MQTT брокера. В кінці виконується перевірка коректності передавання даних і, у разі потреби, виведення повідомлень про помилку.

Лістинг 3.4 – Програмний код функції передачі даних на MQTT сервер

```

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

  if (digitalRead(PIR_SENSOR) == HIGH &&
  digitalRead(button1_pin) == HIGH) {
    ThingSpeak.setField(3,1);
  }
  else{
    ThingSpeak.setField(3,0);
  }
  int sensorValue = analogRead(ldrPin);

  if ((sensorValue < 100) && (temperature > 50)) {

```

```

ThingSpeak.setField(5,1);
} else {
ThingSpeak.setField(5,0);
}
ThingSpeak.setField(1,temperature);
ThingSpeak.setField(2,humidity);
ThingSpeak.setField(4,relay4_state);

int x = ThingSpeak.writeFields(myChannelNumber,myApiKey);
if(x == 200){
Serial.println("Data pushed successfully");
}else{
Serial.println("Push error" + String(x));
}
Serial.println("---");
}
}

```

Пристрій також має функціонал відображення поточних параметрів на локальному символічному дисплеї. Частина програмного коду, яка відповідає за цей функціонал приведена у лістингу 3.5, а результат роботи показаний на рис. 3.2.

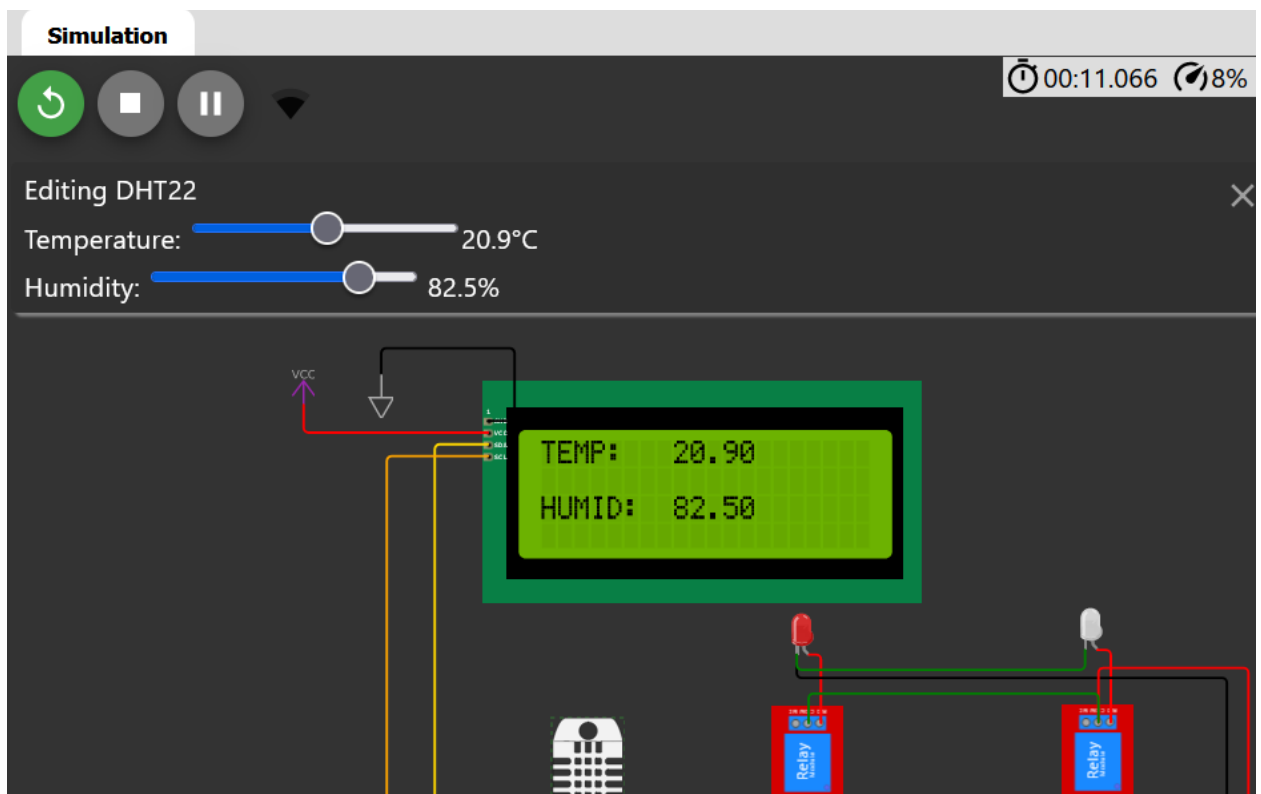


Рисунок 3.2 – Результат відображення даних на дисплеї

Лістинг 3.5 – Відображення даних на локальному дисплеї

```
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("TEMP:");  
lcd.setCursor(8, 0);  
lcd.print(temperature);  
lcd.setCursor(0, 2);  
lcd.print("HUMID:");  
lcd.setCursor(8, 2);  
lcd.print(humidity);
```

Загальна логіка роботи програмної частини вузла однорангової мережі може бути описана наступним чином:

- виконання підготовчих програмних операцій – об’явлення констант, визначення точок підключення датчиків;
- налаштування режимів роботи датчиків та форматів передаваних даних;
- налаштування параметрів MQTT брокера та способів його взаємодії з вузлом мережі;
- підключення до мережі та MQTT брокера;
- зчитування інформації з датчиків;
- відображення поточних параметрів на локальному дисплеї;
- передавання даних до MQTT брокера.

ВИСНОВКИ

1. Інтернет речей охоплює величезну мережу датчиків і хмарних ресурсів. У цій мережі датчики та пристрої можуть обмінюватися даними між собою та передавати свої дані на більш потужні пристрої для обробки. Універсальність IoT дозволяє застосовувати його в широкому діапазоні галузей, включаючи IT та мережі, промислові операції, охорону здоров'я, розумні будинки, виробництво енергії та багато інших.

2. Показано, що однорангові мережі добре підходять для комунікації всередині екосистеми Інтернет речей. Передача даних в таких системах здійснюється безпосередньо між окремими вузлами, без необхідності використання сервера, як комунікаційної ланки. Також в IoT можливе застосування гібридної моделі однорангової мережі, яка поєднує в собі елементи класичної та P2P архітектур.

3. Передача даних в однорангових мережах може здійснюватись з використанням багатьох протоколів, та для Інтернету речей. Найкраще для цього підходять протокол прикладного рівня DDS, спеціалізований протокол CoAP та відкриті протоколи MQTT і XMPP, оскільки враховують особливості комунікації в мережах, схильних до динамічних змін топології мереж.

4. Показано, що апаратно вузли однорангової мережі можуть бути побудовані на базі мікроконтролера та модуля зв'язку з підтримкою роботи в необхідному частотному діапазоні. Нами для розробки такої системи було обрано мікроконтролерну платформу ESP32, оскільки вона має вбудований модуль передатчика 2,4 ГГц а програмний код для роботи може бути написаний мовою програмування C++.

5. Розроблено схему та програмний код вузла однорангової мережі на основі мікроконтролера ESP32, який здійснює зчитування даних з датчиків, відображає їх на локальному символічному дисплеї. Пристрій здійснює також передачу цих даних до хмарного сервісу, представленого у вигляді MQTT брокера, чим забезпечує функціонал гібридної однорангової мережі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What Is Peer-To-Peer? Meaning, Features, Pros, and Cons [Електронний ресурс]. – Режим доступу: <https://www.spiceworks.com/tech/networking/articles/what-is-peer-to-peer/> 10.04.2024
2. Peer-to-Peer Energy Trading in a Micro-grid Using Internet of Things and Blockchain / M. Jabbar Aziz Baig et al. *Electronics ETF*. 2021. Vol. 25, no. 2. P. 39–49.
3. Bin Sulaiman R. Future Threats to Internet of Things (IoT) Security & Privacy: A Survey. *SSRN Electronic Journal*. 2019.
4. Bhavana C B., C T Vathsala Gowda., B H Rakshitha. A Survey: Internet of Things (IOT) Technologies, Applications. *International Journal for Research in Applied Science and Engineering Technology*. 2022. Vol. 10, no. 8. P. 640–644
5. IoT Explained — How Does an IoT System Actually Work? [Електронний ресурс]. – Режим доступу: <https://medium.com/iotforall/iot-explained-how-does-an-iot-system-actually-work-e90e2c435fe7> 10.04.2024
6. The advantages and disadvantages of Internet Of Things [Електронний ресурс]. – Режим доступу: <https://e27.co/advantages-disadvantages-internet-things-20160615/> 10.04.2024
7. Протоколи інтернету речей [Електронний ресурс]. – Режим доступу: https://e-tk.lntu.edu.ua/pluginfile.php/20331/mod_resource/content/0/%D0%A2%D0%B5%D0%BC%D0%B0%2014.%20%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D0%B8%20%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%D1%83%20%D1%80%D0%B5%D1%87%D0%B5%D0%B8%CC%86.pdf 10.04.2024
8. Internet of Things: A Comprehensive Overview on Protocols, Architectures, Technologies, Simulation Tools, and Future Directions / M. Mansour et al. *Energies*. 2023. Vol. 16, no. 8. P. 3465

9. Lakshminarayana S., Praseed A., Thilagam P. S. Securing the IoT Application Layer from an MQTT Protocol Perspective: Challenges and Research Prospects. *IEEE Communications Surveys & Tutorials*. 2024. P. 1.
10. Top P2P IoT Platforms in 2024 [Електронний ресурс]. – Режим доступу: <https://bytebeam.io/blog/top-p2p-iot-platforms/#some-popular-iot-platforms-in-2024> 10.04.2024
11. Chakraborty, Sudip and Aithal, P. S., Let Us Create Multiple IoT Device Controller Using AWS, ESP32 And C# (April 24, 2023). *International Journal of Applied Engineering and Management Letters (IJAEML)*, 7(2), 27-34.
12. Miniature Design of Liquid Filling Machine Automatically Using ESP32 Based IOT (Internet of Things) / A. Munandar et al. *Jurnal Komputer, Informasi dan Teknologi (JKOMITEK)*. 2023. Vol. 3, no. 1.
13. M. H. Widiyanto and V. H. Cahaya Putra, "Utilization of Wokwi Simulation Application in Supporting Internet of Things Learning (IoT)," *2023 International Conference on Information Management and Technology (ICIMTech)*, Malang, Indonesia, 2023, pp. 807-812
14. A. C. Bento, C. Vázquez-Hurtado, D. C. Gatti, C. M. Solis Garza, I. M. Amaya Contreras and S. Camacho-Leon, "Experimental Survey with IoT Students and Virtual Classes with Wokwi Circuits," *2023 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, Valdivia, Chile, 2023, pp. 1-5
15. Холоденко А.С., Немцов В.А., Тищенко К.В. Ідентифікація та моніторинг елементів IoT в однорангових мережах / Матеріали Міжнародної науково-технічної конференції студентів та молодих вчених «Фізика, електроніка, електротехніка ФЕЕ-2024». – Суми: СумДУ, 2024. – С.24.