

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Водяного Олександра Вікторовича
(прізвище, ім'я, по батькові здобувача освіти)

УДК 004.4/.5

КВАЛІФІКАЦІЙНА РОБОТА

Навчальна модель розумного будинку на базі технологій IoT
(тема роботи)

122 «Комп'ютерні науки»
(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

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

_____ Водяний О.В.
(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Лапін А. В
(прізвище, ім'я, по батькові)
к.е.н., доцент кафедри КТіМС
(науковий ступінь, вчене звання)

Висновок кафедри _____
за результатами попереднього захисту:

Протокол засідання кафедри

№ _____ від « _____ » _____ 20 _____ р.

Завідувач кафедри _____

_____ (науковий ступінь, вчене звання) _____ (підпис) _____ (прізвище, ім'я, по батькові)
« _____ » _____ 20 _____ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____
за шкалою ECTS _____
за національною шкалою _____

Секретар ЕК

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

АНОТАЦІЯ

Водяний О. В. Навчальна модель розумного будинку на базі технологій IoT – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 122 – комп'ютерні науки. – Поліський національний університет, Житомир, 202.

Зміст анотації

У кваліфікаційній роботі розглянуто процес проєктування, розробки та реалізації навчальної моделі системи розумного будинку з використанням технологій Інтернету речей (IoT). Головною метою дослідження було створення доступної фізичної моделі для використання в освітньому процесі, яка дозволяє студентам на практиці опанувати принципи побудови, програмування та тестування розумних систем. Як центральний контролер використано мікроконтролер ESP8266, до якого підключено сенсори та виконавчі пристрої. Конструктивна частина моделі виконана за допомогою друку на 3D-принтері із подальшим монтажем компонентів у корпус будинку. Програмна частина реалізована в середовищі Arduino IDE з підтримкою Wi-Fi та вебінтерфейсу для взаємодії з користувачем. Практична цінність роботи полягає в створенні універсального, змінного та масштабованого навчального макета, що дозволяє студентам відпрацьовувати ключові навички мікроконтролерного програмування, побудови IoT-архітектур та розробки автоматизованих систем у побутовому середовищі.

Ключові слова:

мікроконтролер, практичне проєктування, інтерактивне навчання, інтеграція сенсорів, 3D-друк.

SUMMARY

Vodiany O. V. Educational model of a smart home based on IoT technologies – Qualification work in the form of a manuscript.

Qualification work for obtaining a bachelor's degree in specialty 122 – computer science. – Polesie National University, Zhytomyr, 2025.

This thesis presents the design, development, and implementation of an educational prototype of a smart environment based on Internet of Things (IoT) technologies. The primary goal of the project is to create an accessible and functional physical model that supports the practical training of students in the fields of IoT systems, and automation. The system is built around the ESP8266 microcontroller, which manages data from environmental sensors and controls output devices. The physical structure of the model was developed using Blender and manufactured using 3D printing, with components integrated directly into scaled room mock-ups. The software part is implemented in the Arduino IDE environment using the C++ language, with integrated support for Wi-Fi connectivity and a simple web interface for real-time monitoring and interaction. The practical value of the project lies in its flexibility, modularity, and ease of replication, making it an effective hands-on platform for learning the design, programming, and deployment of modern IoT-based systems.

Keywords:

microcontroller, practical design, interactive learning, sensor integration, 3D printing.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.2. Моделювання бізнес-процесів інформаційної системи розумного дому.....	10
1.2. Інформаційні потреби навчальної моделі розумного будинку.....	12
1.3. Функціональні вимоги до навчальної моделі розумного будинку.....	13
РОЗДІЛ 2 ПРОЕКТУВАННЯ НАВЧАЛЬНОЇ МОДЕЛІ РОЗУМНОГО ДОМУ.....	15
2.1 Узагальнена архітектура та структурна схема системи.....	15
2.2 Інфологічне моделювання системи.....	15
2.3 Реалізація функціональних блоків.....	25
РОЗДІЛ 3. РЕАЛІЗАЦІЯ, НАЛАШТУВАННЯ ТА ТЕСТУВАННЯ НАВЧАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ РОЗУМНОГО БУДИНКУ.....	28
3.1 Практична реалізація навчальної моделі системи розумного будинку.....	28
3.2 Середовище розробки Arduino IDE та налаштування плати ESP8266.....	31
ВИСНОВКИ.....	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	38
ДОДАТКИ.....	42

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IOT – Інтернет речей

IP – Інтернет-протокол

UML - Unified Modeling Language

TCP/IP - Transmission Control Protocol/Internet Protocol

ОС - Операційна система

ВСТУП

У наш час світ перебуває на етапі інтенсивного розвитку інформаційних технологій, де важливе місце займає концепція Інтернету речей (IoT, Internet of Things). Ця технологія полягає в об'єднанні фізичних пристроїв в єдину мережу для автоматизованої взаємодії між собою без необхідності постійного втручання людини. Однією з найперспективніших галузей практичного застосування IoT є розробка систем розумного будинку, які сприяють підвищенню рівня комфорту, безпеки, енергоефективності та загальної якості життя користувачів. Розвиток і впровадження подібних технологій сьогодні стає все більш популярним, що одночасно зумовлює потребу у підготовці кваліфікованих фахівців, здатних працювати з подібними системами.

Актуальність теми цієї дипломної роботи визначається зростаючим попитом на спеціалістів, які не лише теоретично знайомі з технологіями Інтернету речей, а й мають практичні навички розробки та впровадження систем розумного будинку. Водночас існує проблема доступності якісних навчальних матеріалів і макетів, які б дозволяли студентам опановувати ці технології в умовах максимально наближених до реальності. Комерційні рішення, які використовуються на ринку, як правило, є дорогими, складними та часто непристосованими для навчальних цілей, що ускладнює їх застосування в освітньому процесі. У зв'язку з цим розробка доступної та функціональної навчальної моделі розумного будинку, яка дозволяє студентам працювати з реальними компонентами IoT-систем, є важливою та своєчасною задачею.

Метою даної дипломної роботи є створення фізичної навчальної моделі розумного будинку з використанням технології 3D-друку для виготовлення окремих елементів та інтеграцією в неї сучасних сенсорів, мікроконтролерів ESP8266 та інших пристроїв автоматизації, що моделюватимуть основні функціональні можливості реального розумного будинку.

Для досягнення поставленої мети було проведено аналіз сучасного стану розвитку технологій розумного будинку та їх застосування в освітньому процесі, спроектовано архітектуру навчальної моделі з урахуванням необхідних

функціональних блоків, створено 3D-моделі кімнат за допомогою програмного забезпечення Blender, виготовлено фізичні компоненти будинку методом 3D-друку, підібрано відповідні сенсори та мікроконтролери для реалізації функціональності системи. Крім того, було розроблено програмне забезпечення на базі мікроконтролерів ESP8266 для забезпечення зв'язку між елементами системи та реалізації сценаріїв автоматизації, а також проведено тестування моделі та оцінку її ефективності для використання в навчальному процесі.

Об'єктом дослідження в даній роботі виступає процес професійної підготовки фахівців у галузі інформаційних технологій та автоматизації з використанням практичних моделей систем розумного будинку. Предметом дослідження є фізична навчальна модель розумного будинку, створена за допомогою технології 3D-друку та технологій Інтернету речей.

У процесі виконання роботи застосовувалися різні методи дослідження. Було проведено аналіз літературних джерел для вивчення сучасних тенденцій розвитку технологій IoT та систем розумного будинку. Метод проєктування використовувався для створення архітектури навчальної моделі, а 3D-моделювання дозволило сформувати фізичні об'єкти будинку, які були виготовлені на 3D-принтері. Експериментальні дослідження дали можливість протестувати роботу інтегрованих сенсорів та виконавчих пристроїв, оцінити функціональність системи в умовах, наближених до реальних. Програмування мікроконтролерів ESP8266 здійснювалося за допомогою мови програмування C++ у середовищі Arduino IDE, що дозволило реалізувати управління елементами розумного будинку.

Практичне значення роботи полягає у створенні прототипу навчальної моделі розумного будинку, яка може бути ефективно використана в навчальному процесі для підготовки спеціалістів у сфері Інтернету речей, автоматизації та інформаційних технологій. Розроблена модель надає студентам можливість отримати практичні навички зі створення, налаштування, програмування та тестування елементів розумного будинку, що

суттєво підвищує рівень їх професійної компетентності та готовності до роботи з сучасними технологіями.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Поняття та особливості систем розумного будинку

Розвиток технологій Інтернету речей (IoT) є одним із ключових трендів сучасного етапу цифрової трансформації. Концепція IoT передбачає створення середовища, в якому фізичні об'єкти забезпечені сенсорами, виконавчими механізмами, контролерами та можливістю обміну даними через мережі зв'язку. Вперше ідея об'єднання фізичних об'єктів в єдину інформаційну мережу була запропонована Кевіном Ештоном ще наприкінці 1990-х років, а термін "Інтернет речей" набув популярності після його використання в проєктах компанії MIT Auto-ID Center [1].

Згідно з дослідженнями компанії Cisco, вже у 2020 році кількість підключених пристроїв перевищила кількість населення планети в декілька разів [2]. Одним із найбільш яскравих прикладів застосування концепції IoT є розробка систем розумного будинку, які інтегрують різноманітні пристрої для автоматизації побутових процесів, підвищення рівня комфорту, безпеки та оптимізації енергоспоживання.

Система розумного будинку – це комплекс технічних і програмних рішень, що забезпечує автоматичне або дистанційне керування інженерними системами будівлі, такими як освітлення, опалення, вентиляція, кондиціонування, безпека, мультимедійні пристрої тощо [3]. Головна ідея розумного будинку полягає у створенні максимально комфортного, безпечного та енергоефективного середовища для мешканців.

До складу системи розумного будинку входять різні за призначенням функціональні елементи. Сенсори забезпечують збір інформації про стан навколишнього середовища, зокрема рівень температури, вологості, освітленості, наявність руху, витік газу чи води. Отримані дані передаються до контролерів, які виконують обробку сигналів і приймають рішення про подальші дії. Виконавчі механізми, такі як реле, електроприводи, клапани або

освітлювальні пристрої, втілюють ці рішення у вигляді фізичних змін у середовищі – наприклад, вмикають світло або перекривають подачу води. Комунікаційні модулі забезпечують передачу даних між усіма компонентами системи, а також дозволяють здійснювати зовнішнє керування через мережу Інтернет. Користувацькі інтерфейси, включаючи мобільні застосунки, веб-платформи або голосових помічників, забезпечують зручний доступ до функцій системи, дозволяють переглядати поточний стан приміщення, надсилати команди та створювати сценарії автоматизації [4]. Архітектура типового розумного будинку має ієрархічну структуру, що дозволяє забезпечити як локальну автономну роботу пристроїв, так і централізоване або віддалене керування через Інтернет.

1.2. Моделювання бізнес-процесів інформаційної системи розумного дому

Для побудови системи розумного будинку важливим є формальне моделювання її логіки та взаємодії між елементами. У межах даної роботи для цього використано методології IDEF0 та IDEF3, які дозволяють формалізувати діяльність системи, описати структуру функцій і виявити інформаційні потоки між компонентами. Контекстну IDEF0-діаграму представлено на рисунку 1.2.

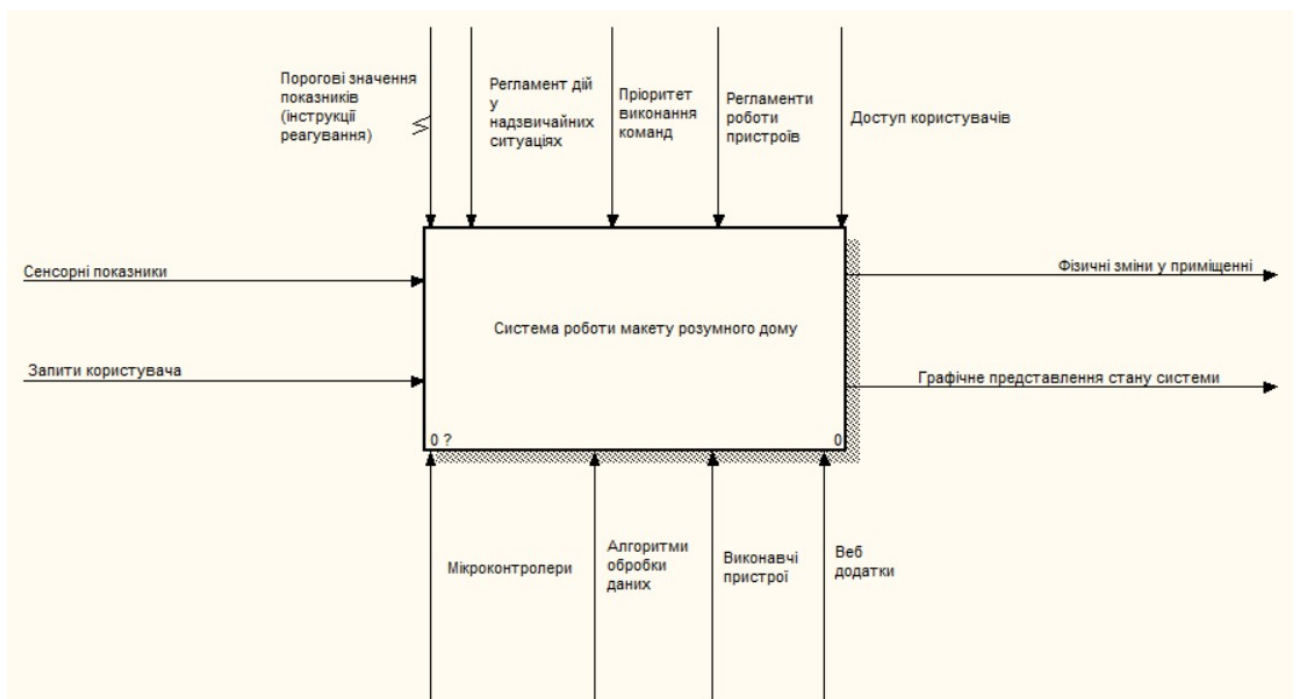


Рисунок 1.1 – Контекстна IDEF0-діаграма системи роботи розумного дому

Контекстна діаграма IDEF0 (рис. 1.1) ілюструє загальну функціональну модель системи. Вона демонструє основний процес – роботу макету розумного дому – разом з усіма вхідними даними, керуючими параметрами, механізмами реалізації та очікуваними виходами. На вході до системи надходять сенсорні показники та запити користувачів, які аналізуються відповідно до заданих порогів, правил дій у надзвичайних ситуаціях та пріоритетів. На виході система формує фізичні зміни в приміщенні, наприклад вмикання освітлення, а також графічне представлення стану на екрані. Розширену IDEF0-діаграму представлено на рисунку 1.3.

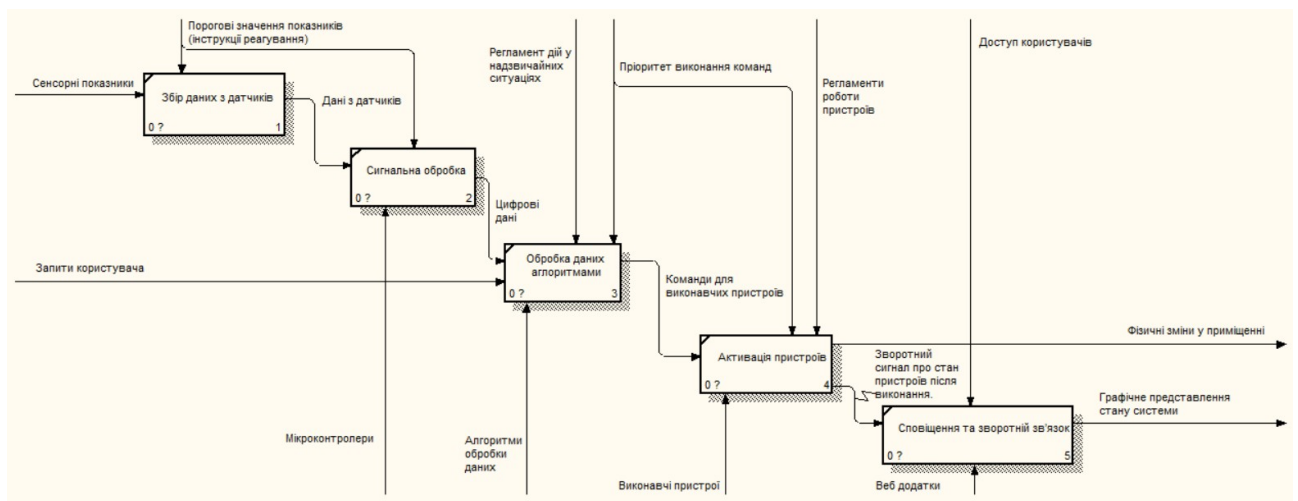


Рисунок 1.2 – Розширена IDEF0-діаграма системи роботи розумного дому

Розширена діаграма IDEF0 (рис. 1.2) деталізує основні функціональні блоки системи. На першому рівні показано етап збору даних з датчиків, подальшу обробку сигналів, алгоритмічний аналіз, генерацію команд для виконавчих пристроїв та візуалізацію результатів. Діаграма ілюструє, як саме інформація з навколишнього середовища проходить трансформацію і приводить до реальних змін у стані будинку. Цей рівень деталізації дозволяє краще зрозуміти логіку роботи системи і визначити критичні точки її налаштування. IDEF3-діаграму представлено на рисунку 1.3.

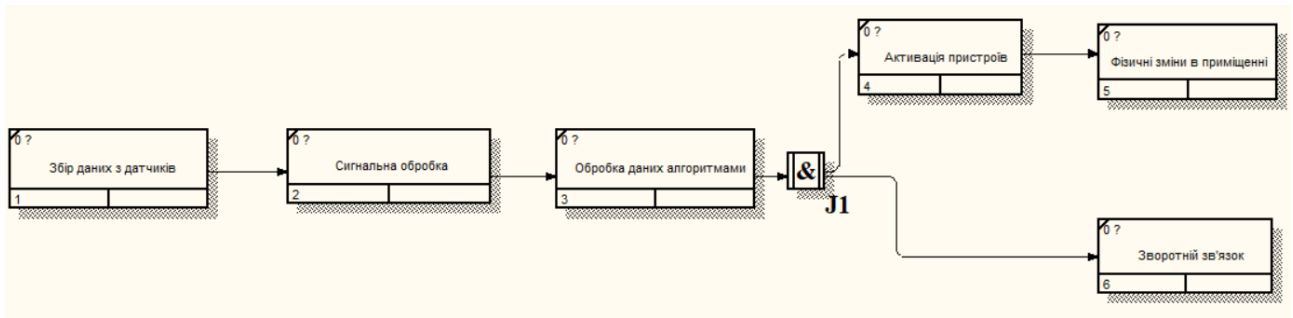


Рисунок 1.3 – IDEF3-діаграма системи роботи розумного дому

Для опису часових і логічних залежностей між подіями використано діаграму IDEF3 (рис. 1.3), яка моделює послідовність виконання дій. На діаграмі зображено послідовні кроки: від збору даних з датчиків, через обробку, аналіз та активацію пристроїв – до зміни стану середовища та формування зворотного зв'язку. Ця модель дозволяє чітко відслідковувати, які саме події ініціюють наступні, та які умови мають бути виконані для переходу до наступного етапу.

1.2. Інформаційні потреби навчальної моделі розумного будинку

У межах освітніх програм підготовки фахівців з інформаційних технологій та автоматизації виникає необхідність у створенні фізичних навчальних моделей, які б дозволяли студентам опановувати базові принципи побудови та функціонування систем розумного будинку.

Інформаційні потреби цієї моделі охоплюють цілу низку функціональних можливостей. Вона повинна забезпечувати збирання даних про фізичні параметри навколишнього середовища, такі як температура, вологість, рівень освітленості та наявність руху, тобто володіти засобами моніторингу ключових показників [5]. Не менш важливою є здатність моделі обробляти зібрані дані в режимі реального часу, використовуючи для цього мікроконтролери з відповідним програмним забезпеченням. Обробка має здійснюватися з урахуванням заздалегідь визначених порогів, логіки реагування та сценаріїв взаємодії між компонентами системи.

Навчальна модель повинна підтримувати можливість дистанційного керування виконавчими пристроями, серед яких можуть бути освітлювальні елементи, вентилятори, реле та інші елементи автоматизації. Це керування має

здійснюватися через бездротові мережі, зокрема з використанням Wi-Fi, що дозволяє реалізувати принцип доступу з будь-якої точки мережі. Важливо, щоб система мала можливість реалізації різноманітних сценаріїв автоматизації, які ґрунтуються на отриманих з сенсорів даних — наприклад, автоматичне вмикання освітлення при зниженні рівня освітленості або запуск вентилятора при перевищенні певного температурного порогу.

1.3. Функціональні вимоги до навчальної моделі розумного будинку

З огляду на поставлені інформаційні потреби, до навчальної моделі розумного будинку формуються наступні функціональні вимоги.

Модель має забезпечувати збір даних про температуру, вологість та освітленість за допомогою відповідних сенсорів, таких як DHT22 або аналогічних. Вона повинна мати можливість виявлення руху за допомогою PIR-датчиків, що є необхідним для створення системи безпеки.

Обробка даних повинна здійснюватися за допомогою мікроконтролерів ESP8266 з можливістю бездротової передачі інформації по мережі Wi-Fi. З метою забезпечення управління пристроями передбачено використання реле для вмикання та вимикання пристроїв. Контроль над пристроями повинен бути доступний як автоматично, так і через мобільний застосунок або веб-інтерфейс.

Модель має передбачати можливість створення сценаріїв автоматизації, наприклад, автоматичне вмикання світла при зниженні рівня освітлення або увімкнення вентилятора при перевищенні температурного порогу. Збір, обробка і візуалізація даних повинні відбуватися в режимі реального часу [6].

Оскільки модель призначена для навчання, її конструкція повинна бути модульною та легкою у збірці і розборці. Всі компоненти мають бути доступними для повторного використання і модернізації. 3D-моделі приміщень будинку повинні бути розроблені у програмному забезпеченні Blender та виготовлені за допомогою технології FDM-друку (Fused Deposition Modeling), що дозволяє забезпечити достатню деталізацію і реалістичність макета.

Програмне забезпечення має бути розроблене на мові програмування C++ в середовищі Arduino IDE, оскільки цей інструмент забезпечує зручність

розробки, велику кількість бібліотек для роботи з різними сенсорами та пристроями і є загальнодоступним у навчальних цілях.

Висновки до першого розділу

У цьому розділі було проведено аналіз предметної області дослідження, розглянуто основні концепції, а також встановлено інформаційні потреби навчальної моделі. Визначено функціональні вимоги до системи, серед яких: збір та обробка даних із сенсорів, бездротове керування виконавчими пристроями, можливість створення сценаріїв автоматизації, модульність конструкції та використання доступних інструментів для розробки 3D-моделей і програмного забезпечення. Сформульовані вимоги є основою для подальшої розробки навчальної моделі розумного будинку на базі технологій Інтернету речей.

РОЗДІЛ 2 ПРОЕКТУВАННЯ НАВЧАЛЬНОЇ МОДЕЛІ РОЗУМНОГО ДОМУ

2.1 Узагальнена архітектура та структурна схема системи

Архітектура розробленої системи є багаторівневою. В її основі – мікроконтролер ESP8266, який виконує роль центрального керуючого вузла. Він приймає інформацію з сенсорів, опрацьовує її за закладеною логікою, і, за потреби, активує відповідні виконавчі пристрої або надсилає оновлену інформацію до користувача через веб-інтерфейс. Уся система функціонує на базі локального Wi-Fi-з'єднання, без необхідності постійного підключення до зовнішніх серверів.

Сенсорний рівень включає датчики температури, вологості, освітленості та руху, розміщені в окремих кімнатах фізичної моделі будинку. Зчитування даних із цих сенсорів відбувається циклічно, із заданим часовим інтервалом. Контрольний рівень, який представляє ESP8266, обробляє отримані значення, порівнює їх з наперед визначеними порогами та генерує відповідні події. Виконавчий рівень складається з таких пристроїв, як світлодіодні лампи, реле для підключення навантаження, вентилятори або моделі замків.

На рівні інтерфейсу реалізовано веб-додаток, який дозволяє переглядати поточні показники, вмикати або вимикати пристрої, а також змінювати параметри спрацювання. Структурна схема системи передбачає взаємодію між усіма рівнями – сенсори надсилають дані, контролер реагує, інтерфейс повідомляє користувача або приймає команди. Ця схема є гнучкою, дозволяє масштабування та додає можливість легко інтегрувати нові пристрої без повного перезавантаження логіки системи.

2.2 Інфологічне моделювання системи

Уявіть собі будинок, який сам піклується про ваш комфорт. Це не просто ідея, а результат роботи складної інформаційної системи, яка об'єднує десятки сенсорів, контролерів і виконавчих пристроїв. Для створення навчальної моделі такої системи, яка допоможе студентам опанувати основи IoT та зрозуміти

принципи автоматизації житлових приміщень, важливо спочатку детально описати її структуру та функціонування. UML-діаграму прецедентів представлено на рисунку 2.1.

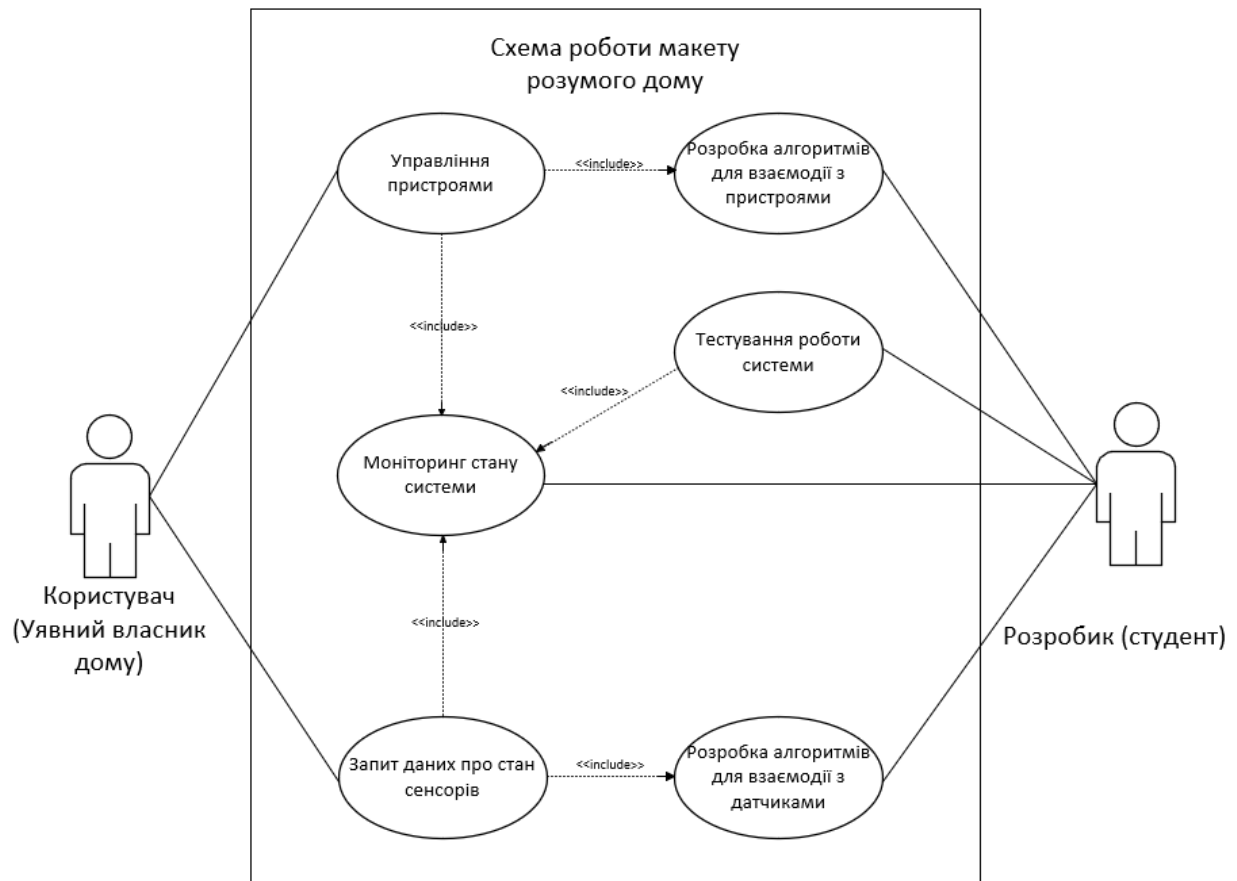


Рисунок 2.1 – UML-діаграма прецедентів макету розумного дому

Головна мета проєкту – не просто створити автоматизовану систему для керування умовами в будинку, а надати студентам інструмент, за допомогою якого вони зможуть на практиці вивчати механізми збору, передачі, обробки інформації та прийняття рішень у середовищі Інтернету речей.

З одного боку, користувач цієї системи, що умовно представлений як уявний власник будинку, взаємодіє з інтерфейсом розумного дому як кінцевий споживач послуг автоматизації. Його участь полягає у тому, щоб здійснювати базові дії: керувати пристроями, ініціювати запити на отримання даних з сенсорів і контролювати поточний стан системи. При цьому всі функції взаємопов'язані з підсистемами, які реалізуються в системі розумного дому. Управління пристроями здійснюється не безпосередньо, а через алгоритми, які було попередньо закладено в систему. Тобто користувач лише ініціює

взаємодію, тоді як сам процес виконується відповідно до логіки, розробленої і протестованої студентом.

З іншого боку, розробник, роль якого виконує студент, працює на глибшому рівні. Його завдання включає розробку алгоритмів для взаємодії як з виконавчими пристроями (реле, освітлення, вентилятори тощо), так і з сенсорними модулями (датчики температури, вологості, освітленості, руху). Саме студент створює структуру логіки, яка буде активувати ті чи інші дії на основі вхідних даних. У межах навчального проєкту він також виконує тестування – це окрема й надзвичайно важлива діяльність, що дозволяє перевірити працездатність усіх функцій системи, виявити помилки в реалізації сценаріїв, удосконалити алгоритми реагування на критичні ситуації.

Цей поділ ролей у системі демонструє двосторонню взаємодію: з одного боку – кінцевий користувач, який не вникає у внутрішню логіку, але може використовувати її результати; з іншого – студент-розробник, який створює саму цю логіку та реалізує її фізично через програмування, підключення сенсорів, керування виконавчими модулями. Така побудова процесу навчання дозволяє студентам опанувати повний цикл розробки IoT-системи – від зчитування даних до автоматичного прийняття рішень на основі заданих умов.

UML-діаграму станів представлено на рисунку 2.2.

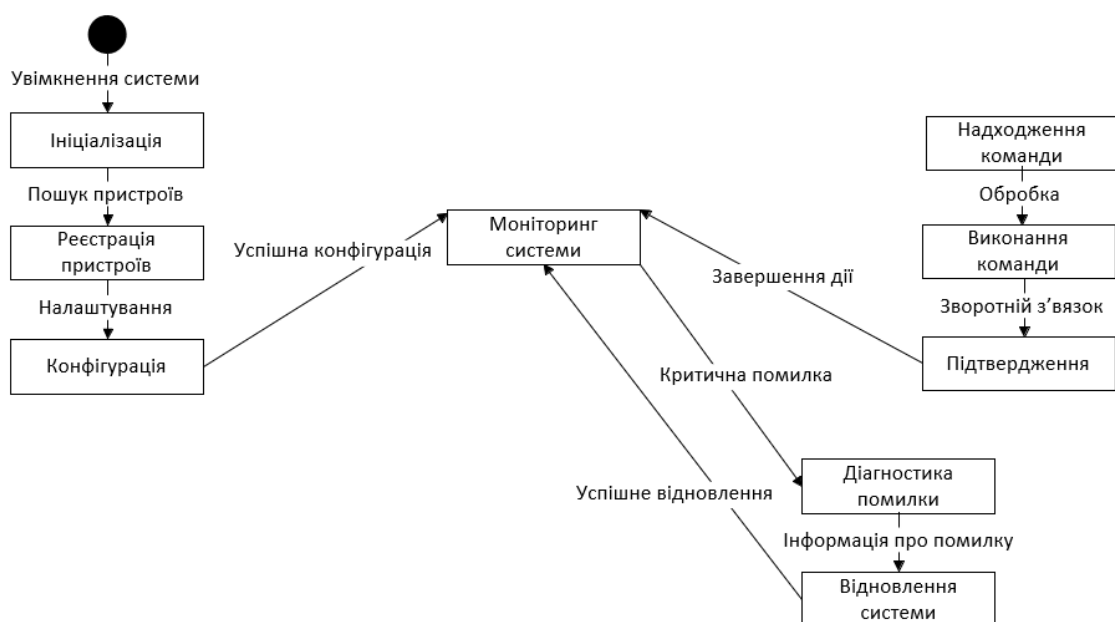


Рисунок 2.2 – UML-діаграма станів макету розумного дому

Система розумного будинку, реалізована як фізична навчальна модель, функціонує як послідовність логічних станів, які змінюються відповідно до внутрішніх подій або дій користувача. Саме це – логіка станів – є надзвичайно важливим об'єктом вивчення в рамках підготовки фахівців з інформаційних технологій.

Процес роботи системи починається зі стартової ініціалізації – моменту, коли вмикається електроживлення, і контролер отримує сигнал до запуску. На цьому етапі студенти мають змогу спостерігати за тим, як система автоматично проводить пошук доступних пристроїв: сенсорів, модулів реле, інтерфейсів виводу тощо. Знайдені пристрої реєструються в контролері, після чого відбувається процес їх налаштування. Кожен пристрій отримує параметри, необхідні для коректної роботи в поточному сценарії. Саме в цьому етапі майбутні фахівці навчаються базовим принципам автоматичного виявлення IoT-компонентів у мережі, роботі з протоколами та конфігурацією периферійних модулів.

Після завершення етапу конфігурації система переходить у режим моніторингу. У цьому режимі вона постійно відстежує стан усіх підключених пристроїв, зчитує значення сенсорів, аналізує параметри навколишнього середовища та очікує на зовнішні команди. Це дає студентам можливість дослідити, як функціонує основна логіка IoT-платформи: збір даних у реальному часі, обробка інформації на мікроконтролері, визначення відхилень від нормативів та ініціація сценаріїв реагування.

Коли до системи надходить команда від користувача (наприклад, увімкнути світло або змінити температурний режим), вона виконує алгоритм її обробки, реалізує необхідну дію через виконавчий пристрій та повертає зворотний зв'язок – повідомлення про успішність виконання або про помилку. Цей етап дозволяє студентам освоїти принципи керування пристроями через бездротові протоколи, налаштування логіки подій і підтвердження результатів операцій.

Окрему увагу в навчанні заслуговує поведінка системи у випадку виникнення критичних помилок – наприклад, втрати зв'язку з пристроєм або конфлікту даних. У таких ситуаціях система переходить до блоку діагностики, який дозволяє виявити тип помилки, проаналізувати причину та ініціювати процедуру відновлення. Після завершення відновлення, якщо система вважає його успішним, вона повертається до режиму моніторингу. Завдяки цьому студенти вчаться реалізовувати базові принципи fault tolerance – стійкості до збоїв, що є критично важливим при створенні надійних систем Інтернету речей [7].

UML-діаграму послідовностей (вмикання приладів) представлено на рисунку 2.3.

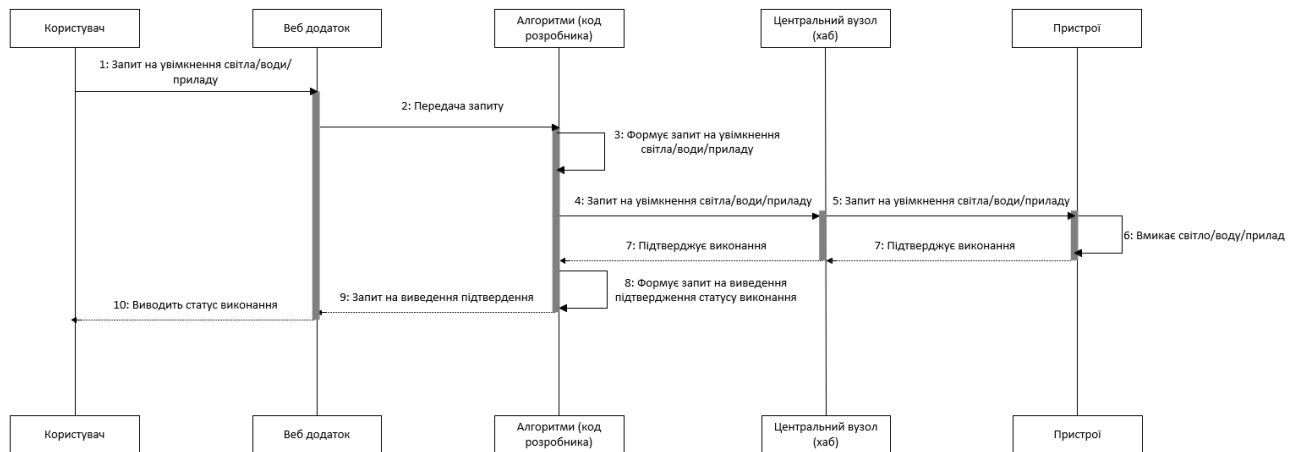


Рисунок 2.3 – UML-діаграма послідовностей (вмикання приладів) макету розумного дому

У навчальній моделі розумного будинку, створеній з метою підготовки студентів до професійної діяльності в галузі IoT, ключове значення має глибоке розуміння логіки взаємодії між усіма елементами системи. Цей процес охоплює кілька етапів – від моменту, коли користувач ініціює запит, до того, як конкретний пристрій у будинку виконує дію. Вивчення цього механізму є особливо важливим для студентів, оскільки дозволяє їм зрозуміти, як побудований повний цикл обміну даними, комунікації між компонентами системи та зворотного зв'язку.

Коли користувач взаємодіє з системою через вебінтерфейс або мобільний застосунок, він фактично створює запит на виконання певної дії: наприклад,

увімкнення світла, запуск вентиляції або відкриття замка. Цей запит обробляється на рівні користувацького інтерфейсу, після чого передається далі до логіки керування, яка була розроблена студентом. Саме в цьому моменті реалізується навчальний підхід до роботи з архітектурою клієнт-сервер у межах IoT-систем.

Програмний код, написаний студентом, обробляє запит користувача, формує відповідну команду та передає її до центрального вузла, який у навчальній моделі виступає в ролі хаба. Хаб виконує функцію комунікаційного посередника між логікою керування та фізичними пристроями. Він приймає запит, перекодовує його у формат, що розуміють пристрої, та надсилає відповідний сигнал на виконання – наприклад, подає напругу на реле для вмикання світильника або активує електронний клапан для подачі води.

Пристрій, який отримав сигнал, виконує дію відповідно до запиту. Після цього він надсилає підтвердження виконання назад до хаба. Центральний вузол, у свою чергу, передає це підтвердження у вигляді статусу виконання до логіки керування, а звідти – у вебінтерфейс користувача. На останньому етапі користувач отримує на екрані інформацію про те, чи була команда виконана, і який результат дії. Таким чином, повний цикл закривається, формуючи завершену транзакцію між ініціатором події та пристроєм-виконавцем.

Для студента ця логіка є практичним відображенням принципів побудови IoT-системи: розуміння, як дані переміщуються між рівнями, як формується запит, яким чином обробляються команди в мікроконтролері, та як забезпечується зворотній зв'язок. Робота з таким сценарієм дозволяє не лише вивчити структуру протоколів (HTTP, MQTT тощо), а й навчитися забезпечувати надійність комунікації, оптимізувати затримки, реалізовувати перевірку виконання та модифікувати логіку взаємодії з пристроями.

З педагогічної точки зору, це один з найважливіших фрагментів навчального процесу, адже він інтегрує одразу кілька аспектів: клієнтську логіку, серверну обробку, апаратну реалізацію та комунікаційну взаємодію. Завдяки роботі з цим сценарієм, студенти отримують унікальний досвід

створення систем, які не просто передають дані, а ще й керують фізичними об'єктами у просторі – що і є сутністю Інтернету речей.

UML-діаграму послідовностей (моніторинг показників із сенсорів) представлено на рисунку 2.4.

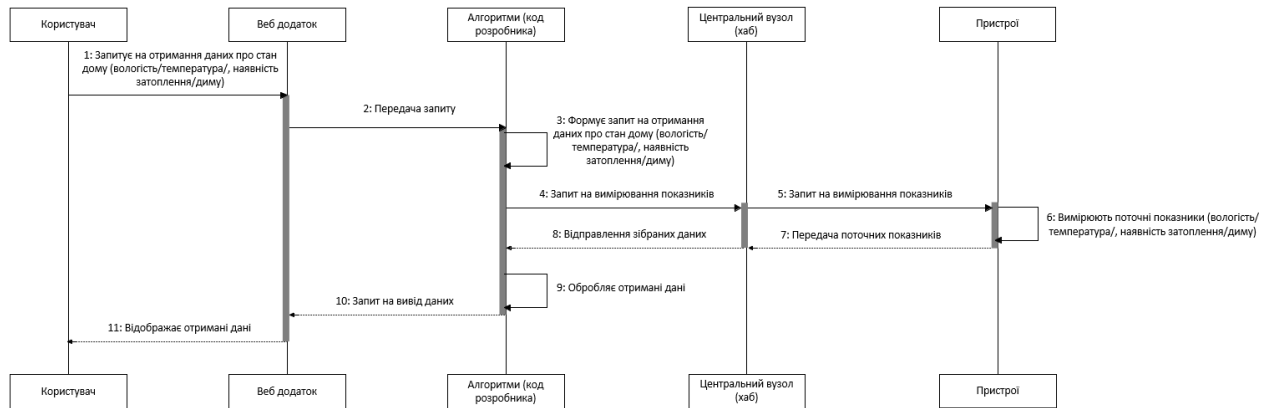


Рисунок 2.4 – UML-діаграма послідовностей (моніторинг показників із сенсорів) макету розумного дому

У навчальній моделі системи розумного будинку важливим аспектом роботи є не лише керування пристроями, а й регулярний збір інформації про поточний стан навколишнього середовища в приміщеннях. Саме можливість отримання сенсорних даних у реальному часі дозволяє системі не лише реагувати на події, а й працювати проактивно – аналізуючи ситуацію, прогножуючи зміни та запобігаючи загрозам. Для студентів це є однією з ключових ділянок навчання: вони повинні зрозуміти, як формується запит на зчитування даних, як працює зворотна передача інформації, як інформація виводиться на інтерфейс, і як її надалі можна інтерпретувати чи використовувати в автоматизованих сценаріях.

Типовий процес взаємодії користувача з навчальною системою починається з ініціації запиту на отримання даних. Це може бути інформація про рівень вологості, температуру повітря, освітленість або навіть сигнали, що вказують на потенційні загрози – наприклад, виявлення затоплення чи диму. Користувач через вебінтерфейс надсилає запит до системи, не занурюючись у технічні деталі реалізації. Саме тому наступний етап – реалізація логіки обробки цього запиту – покладається на розробника, яким у рамках навчального проєкту виступає студент.

Отримавши запит, система передає його до програмної частини, створеної студентом, де формується команда для опитування відповідних сенсорів. Код контролера звертається до центрального вузла – хаба – який далі надсилає запит на вимірювання відповідним пристроям. Сенсори, які підключено до макету, виконують вимірювання за заданими параметрами та повертають актуальні значення. Це можуть бути значення у відсотках вологості, градусах Цельсія або бінарні сигнали у випадку виявлення диму чи води.

Далі відбувається зворотна передача зібраної інформації – спочатку на хаб, потім у модуль обробки, де студентський код перевіряє цілісність і актуальність даних, і лише після цього надсилає їх до вебінтерфейсу. Після отримання даних, інтерфейс виводить їх у зручному вигляді для користувача – це може бути цифрове значення, графік або сигнал попередження. На цьому етапі студенти навчаються не лише технічним аспектам передачі інформації, а й візуалізації, що є важливою складовою користувацького досвіду в IoT-системах.

З технічної точки зору, така послідовність взаємодій навчає студентів працювати з багаторівневою архітектурою IoT-системи. Вони освоюють принципи формування запитів, асинхронного зчитування даних із сенсорів, обробки отриманих результатів, розробки серверної логіки та реалізації зручного інтерфейсу виводу. Це практичне завдання дозволяє не лише краще зрозуміти структуру Інтернету речей, а й дає змогу побудувати навички, необхідні для проєктування повноцінних автоматизованих систем контролю середовища – у будинку, на підприємстві чи у міській інфраструктурі.

UML-діаграму активності (приладів) представлено на рисунку 2.5.

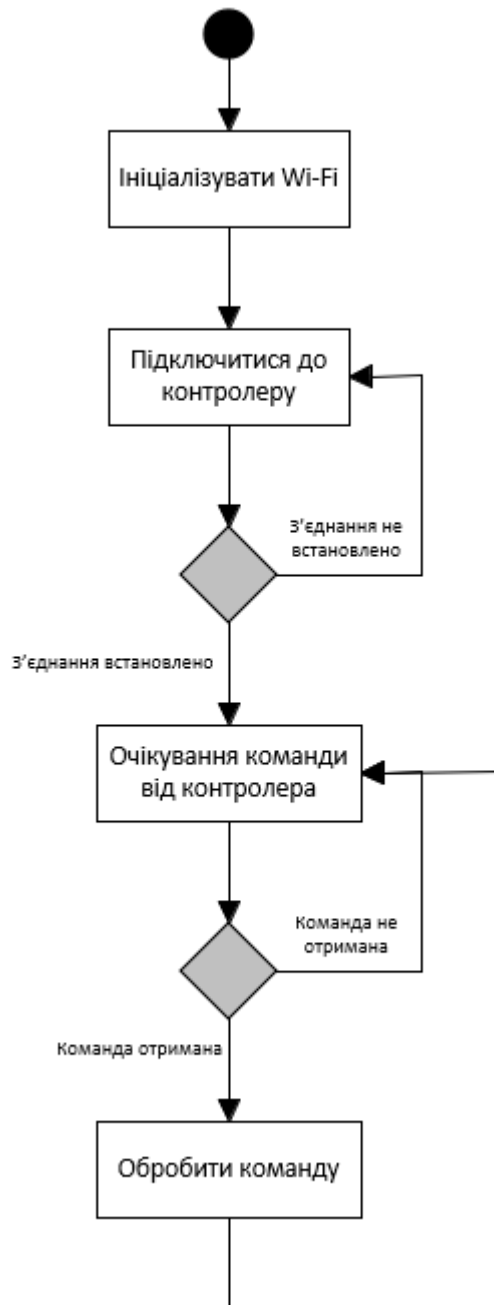


Рисунок 2.5 – UML-діаграма активності (приладів) макету розумного дому

В основі будь-якої системи розумного будинку лежить безперервний процес очікування, прийому та обробки команд. Для того щоб студенти могли на практиці засвоїти ці механізми, навчальна модель IoT-системи має наочно демонструвати, як працює пристрій-виконавець, що підключений до центрального контролера. Такий пристрій не просто пасивно реагує на зовнішній сигнал – він перебуває у стані постійної готовності, очікує команду, перевіряє з'єднання і тільки після цього виконує інструкції. Це типова логіка роботи IoT-модуля, яку студенти повинні вивчити, проаналізувати і навчитися реалізовувати у своїх проєктах.

Першим кроком у роботі пристрою є підключення до мережі. У рамках навчального середовища це зазвичай Wi-Fi-з'єднання, яке встановлюється автоматично після запуску модуля. Студенти мають змогу реалізувати ініціалізацію мережі, прописати параметри підключення, а також створити механізм повторної спроби у разі невдалого з'єднання. Це дозволяє їм засвоїти важливі практики розробки стабільного мережевого коду для IoT-середовища.

Після того як пристрій успішно підключився до локальної мережі та знайшов контролер, він переходить до наступного етапу – очікування команд. Тут важливо, щоб студенти розуміли різницю між активною ініціативою пристрою та пасивним режимом очікування. Більшість IoT-пристроїв працюють саме у другому режимі: вони слухають мережу, аналізують трафік, перевіряють, чи адресовано повідомлення саме їм. Це дозволяє економити ресурси, зменшувати навантаження на мікроконтролер і забезпечувати масштабованість системи.

Коли контролер надсилає команду, пристрій повинен розпізнати її, перевірити валідність (коректність структури, формат, авторизацію) і лише після цього перейти до її виконання. У навчальному проєкті ця команда може стосуватися, наприклад, вмикання світлодіода, запуску вентилятора, активації електромагнітного замка чи зміни порогу спрацювання сенсора. Саме реалізація цього етапу – обробки команди – є найважливішим завданням для студента-розробника, адже вона охоплює роботу з апаратною частиною, маніпулювання GPIO-портами, таймерами, логікою реагування на події та обробкою можливих помилок.

Крім цього, студенти мають можливість реалізувати безкінечний цикл перевірки: якщо команда не отримана – пристрій повертається до стану очікування. Це дозволяє розробити оптимальну логіку взаємодії, яка не перевантажує систему зайвими запитами, але водночас гарантує оперативність реагування. Саме це і є основою побудови ефективної IoT-системи – постійна перевірка стану з'єднання, циклічне очікування подій та умовне виконання дій за заданими сценаріями.

З освітньої точки зору, така логіка дозволяє студентам засвоїти кілька ключових тем одночасно: основи роботи з мережами, взаємодію з контролерами, обробку вхідних подій та програмування пристроїв. Вони отримують змогу навчитися створювати енергоефективні, стабільні та адаптивні рішення, які відповідають сучасним вимогам до IoT-продуктів. Таким чином, простий алгоритм роботи виконавчого пристрою у навчальній моделі стає потужним інструментом формування прикладних навичок для майбутніх спеціалістів цифрової інженерії.

2.3 Реалізація функціональних блоків

Практична реалізація навчальної моделі розумного будинку є центральною частиною проєкту, адже саме в ній студенти мають змогу безпосередньо працювати з фізичними елементами IoT-системи. На зображенні нижче представлено структурну схему, змодельовану у середовищі WokWi – віртуальному симуляторі електронних схем, який широко використовується в освітньому процесі [8].

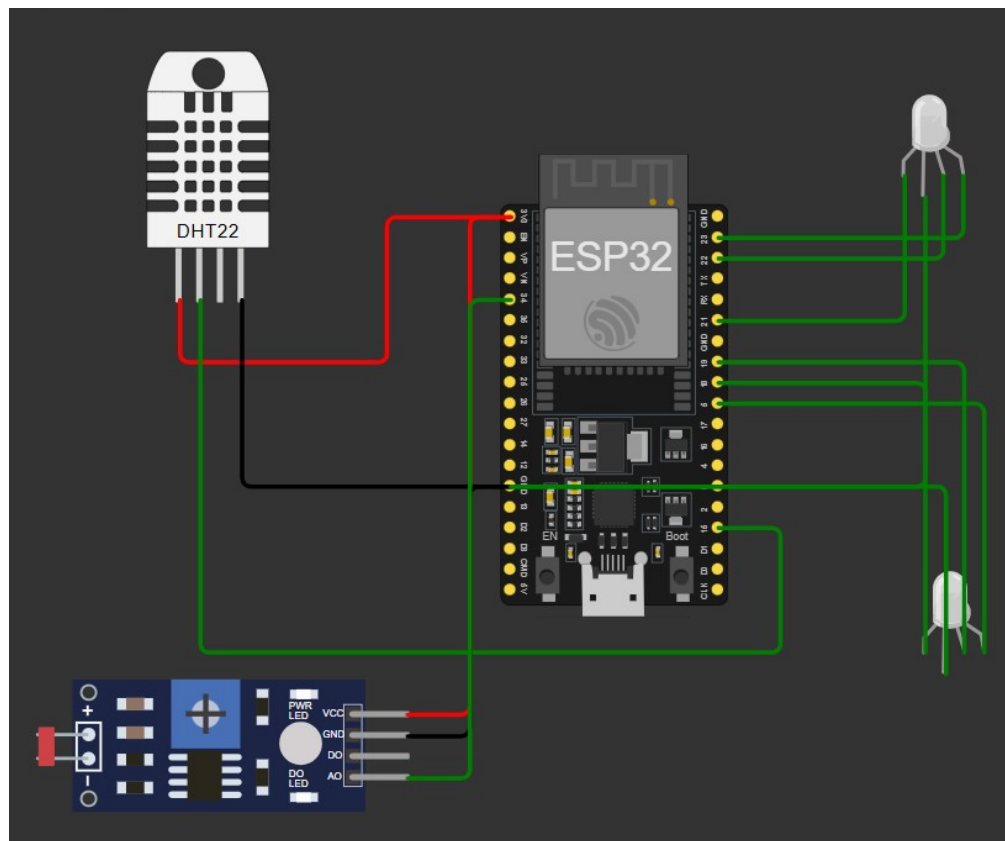


Рисунок 2.6 – Фізична схема навчальної моделі в середовищі WokWi

У центрі схеми знаходиться плата ESP32 – сучасний мікроконтролер з інтегрованим Wi-Fi-модулем, який використовується як головний обчислювальний вузол. Його функціональність дозволяє зчитувати дані з датчиків, аналізувати їх, обробляти події, надсилати команди на виконавчі пристрої та забезпечувати зв'язок з вебінтерфейсом через мережу Wi-Fi. У цій схемі ESP32 поєднується з кількома зовнішніми елементами, які виконують ключові функції у рамках макета.

До мікроконтролера підключено датчик температури та вологості DHT22. Він має три контакти: живлення (VCC), заземлення (GND) і цифровий вихід (DATA). Передача інформації з цього сенсора здійснюється по цифровому протоколу, що дозволяє точно та швидко зчитувати показники температури повітря в °C та відносної вологості в %. Дані використовуються для автоматичного реагування – наприклад, вмикання охолоджувального пристрою, якщо температура перевищує встановлений поріг.

Ще одним елементом у системі є датчик освітленості. Модуль має чотири виходи: живлення (VCC), заземлення (GND), цифровий вихід (DO) та аналоговий вихід (AO). У цьому проєкті використовується цифровий вихід DO, який надсилає логічний сигнал до ESP32 при падінні освітленості нижче встановленого рівня. Це дозволяє автоматично вмикати підсвічування або світлодіоди, імітуючи функцію вечірнього освітлення.

У якості виконавчих пристроїв виступають два світлодіоди (LED), підключені до цифрових виходів ESP32. Кожен з них програмно пов'язаний із певною подією – зміною температури, вологості чи рівня освітлення. Вони вмикаються або вимикаються залежно від результатів обробки даних сенсорів. Студенти можуть змінювати логіку роботи цих індикаторів, експериментуючи з пороговими значеннями та сценаріями автоматизації.

Живлення усіх елементів здійснюється від живлення плати ESP32 (3.3 V), а заземлення з'єднується через загальний GND. Таким чином, уся система утворює єдиний замкнений електричний та логічний ланцюг, у якому кожен компонент виконує своє завдання [9]. Завдяки цьому студенти мають змогу

зрозуміти основи схемотехніки, правила підключення сенсорів та виконавчих пристроїв, роботу з цифровими входами/виходами та способи забезпечення енергетичної стабільності схеми.

Уся логіка системи запрограмована у середовищі Arduino IDE з використанням мови C++. Програмний код містить бібліотеки для роботи з сенсорами (наприклад, DHT.h для DHT22), цикл зчитування даних, обробку порівнянь, а також функції, що керують цифровими виходами для керування світлодіодами [10]. Приклад коду наведено в додатку (додаток А) Особлива увага приділяється обробці помилок – наприклад, ситуацій, коли сенсор не передає дані або коли показники не оновлюються понад заданий інтервал часу.

Висновки до другого розділу

У другому розділі було сформовано цілісне уявлення про архітектуру, алгоритми та реалізацію навчальної моделі системи розумного будинку. Представлено структурну схему, описано логіку функціонування, розроблено п'ять UML-діаграм, що відображають функціональні та інформаційні процеси в системі, а також реалізовано фізичну модель на базі ESP32 з використанням сенсорів температури, вологості та освітленості. Розроблена система повністю відповідає освітнім завданням – вона забезпечує практичне навчання студентів технологіям IoT, сприяє розвитку інженерного мислення та навичок програмування вбудованих систем.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ, НАЛАШТУВАННЯ ТА ТЕСТУВАННЯ НАВЧАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ РОЗУМНОГО БУДИНКУ

3.1 Практична реалізація навчальної моделі системи розумного будинку

Для створення корпусу навчального макету було змодельовано кілька кімнат із базовими меблями, які адаптовані для подальшого вмонтовування сенсорів і виконавчих пристроїв. Моделі розроблялись у програмному забезпеченні Blender, з урахуванням масштабування під компоненти, що використовуються. Особливу увагу приділено функціональності: кожна кімната має технічні отвори для прокладання проводів, знімні панелі або відкриті стінки для доступу до внутрішніх елементів під час навчального процесу.

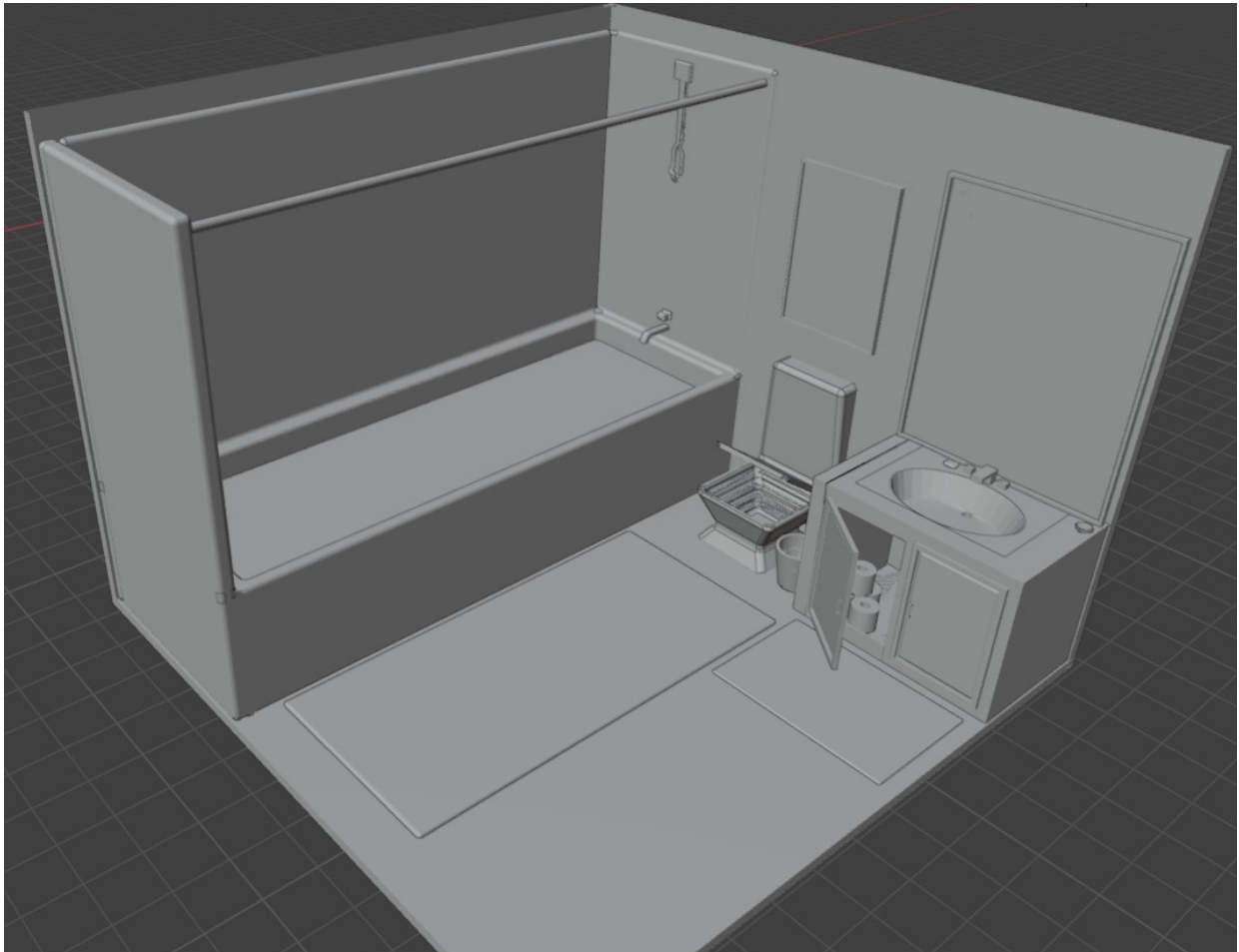


Рисунок 3.1 – 3D-модель кімнати (ванна)

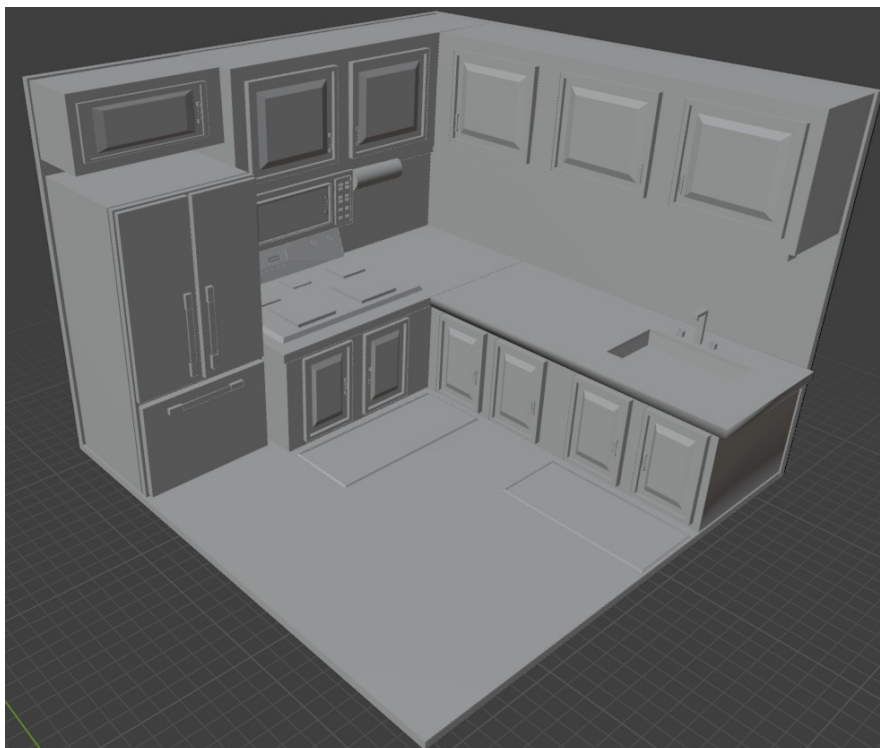


Рисунок 3.2 – 3D-модель кімнати (кухня)

Для друку 3D-моделей використовувався 3D-принтер Creality Ender-3 V3 KE. Цей принтер забезпечує високу якість друку завдяки покращеній стабільності платформи, автоматичному вирівнюванню стола, а також можливості точно відтворювати складні елементи за низьких витрат матеріалу. Як друкарський матеріал обрано PLA-пластик.

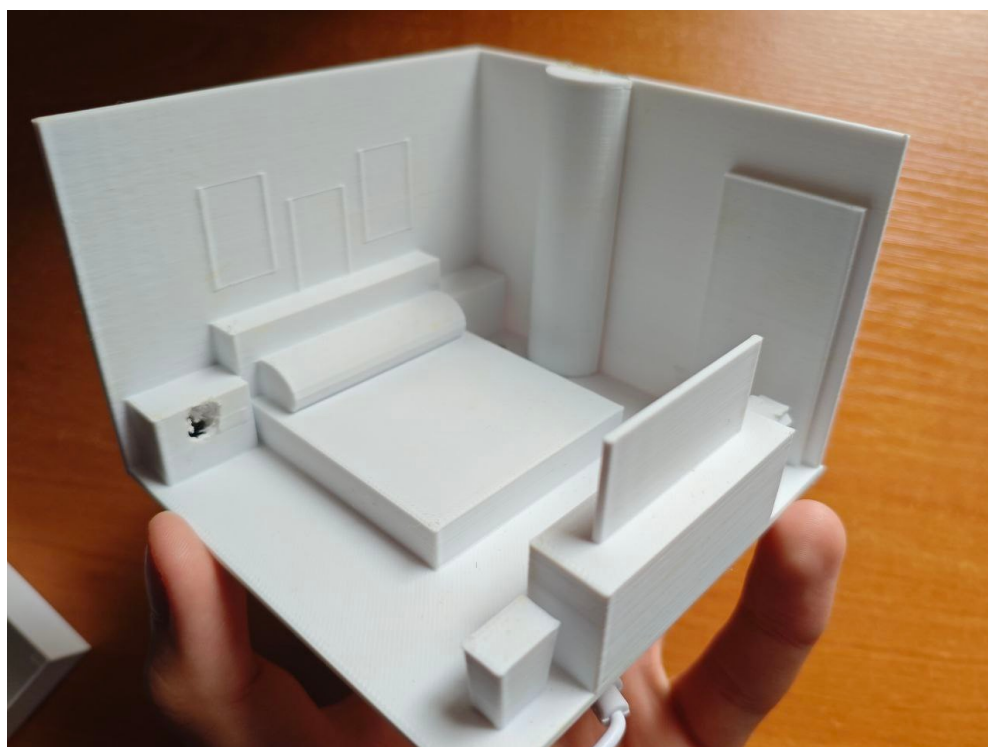


Рисунок 3.3 – надрукована 3D-модель кімнати (спальня)

Після виготовлення корпусу було проведено монтаж електроніки. Усі сенсори та виконавчі елементи підключаються до мікроконтролера ESP8266. З'єднання виконувалось через пайку проводів безпосередньо до контактів або з використанням макетної плати, залежно від компонента. Усі елементи електроніки розміщуються в корпусі будинку. Для надійної фіксації використовувалися монтажні отвори, закладені в моделі ще на етапі проєктування.

Щоб організувати внутрішній простір і приховати кабелі, була змодельована та надрукована 3D-підставка у вигляді коробки без однієї стінки, що виконує роль технічного відсіку. У неї можна помістити мікроконтролер, макетну плату, запас дротів і кабель живлення. Підставка має вихід для USB-з'єднання. Також підставка виконує функцію організації простору – вона унеможливорює пошкодження проводки і сприяє загальній ергономіці моделі.

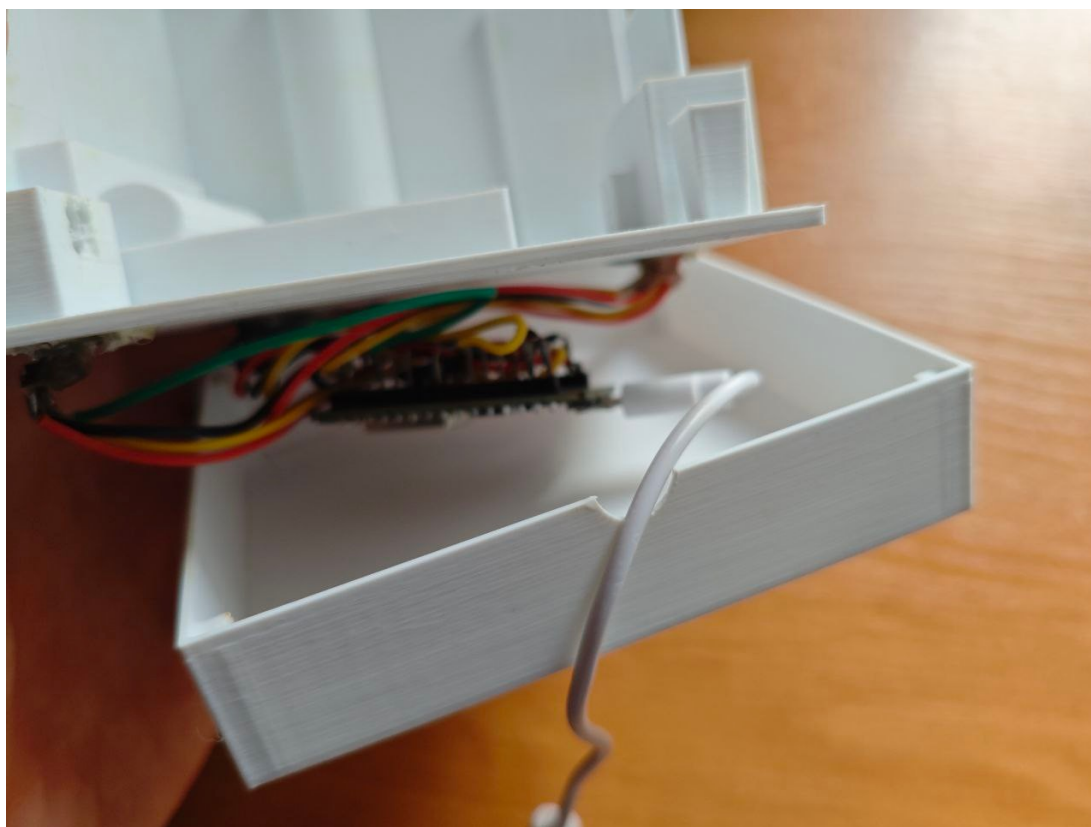


Рисунок 3.4 – 3D-модель підставки для контролера ESP8266

3.2 Середовище розробки Arduino IDE та налаштування плати ESP8266

Розробка навчальної моделі розумного будинку базується на використанні апаратного забезпечення в поєднанні з інструментами програмного забезпечення, які дозволяють створювати та налагоджувати інтерактивну систему Інтернету речей. Основним інструментом для взаємодії студентів з цією системою виступає Arduino IDE – відкрите, кросплатформне середовище для написання, компіляції та завантаження програмного коду на мікроконтролери. В рамках цього проєкту використовується модуль ESP8266, який завдяки вбудованому Wi-Fi-модулю і високій енергоефективності є ідеальним варіантом для побудови недорогих, але функціональних IoT-систем.

Arduino IDE забезпечує всі необхідні інструменти для ефективного навчання: зручний редактор коду, доступ до бібліотек, монітор послідовного порту для налагодження, простий механізм завантаження скетчів на пристрій [11]. Інтерфейс програми максимально адаптований для початківців: студенти легко орієнтуються у вікні середовища, яке включає редактор скетчу (основного програмного коду), панель повідомлень компіляції, панель керування (перевірка, компіляція, завантаження, серійний монітор) та файлове меню для створення і відкриття проєктів.

Щоб почати роботу з ESP8266, необхідно виконати початкове налаштування середовища. Завантаження Arduino IDE здійснюється з офіційного сайту <https://www.arduino.cc/en/software>. Після встановлення потрібно додати підтримку плати ESP8266. Для цього відкривається меню File - Preferences (Файл - Налаштування), у якому в полі Additional Boards Manager URLs вводиться посилання: http://arduino.esp8266.com/stable/package_esp8266com_index.json [12].

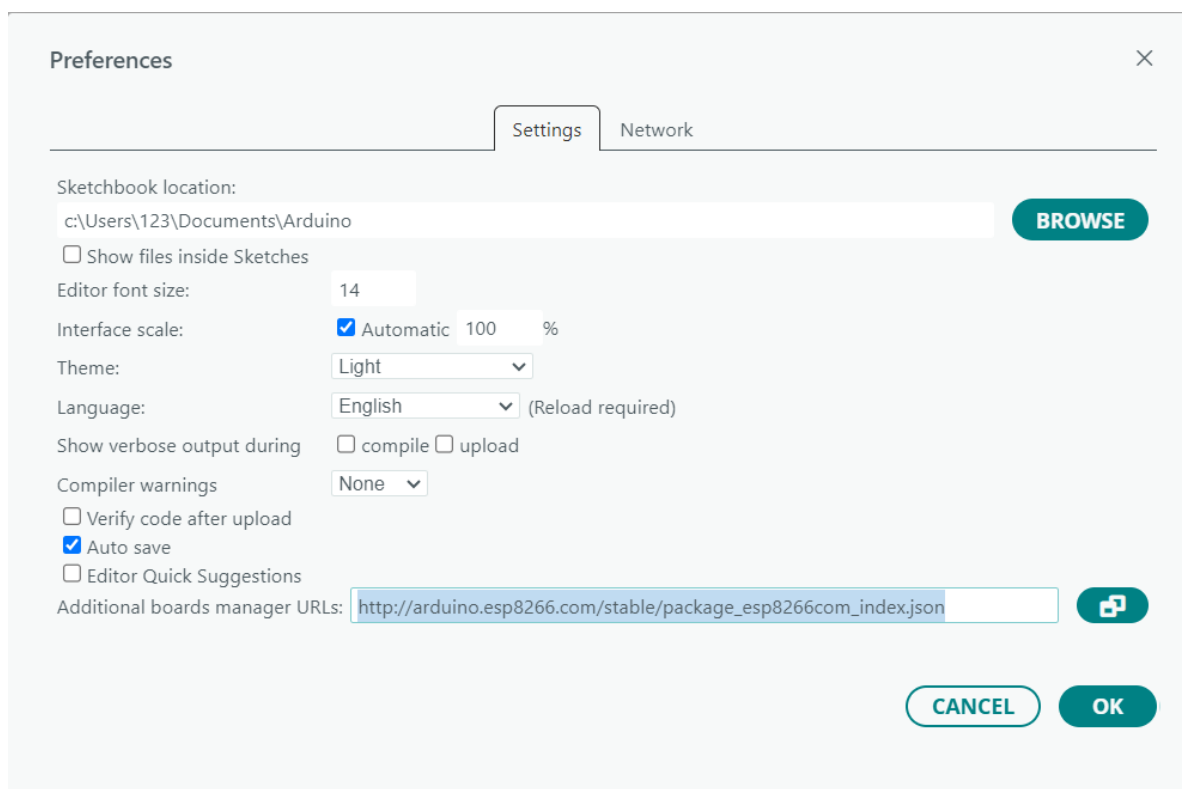


Рисунок 3.5 – Налаштування середовища Arduino IDE (Additional Boards Manager URL)

Це джерело дозволяє менеджеру плат завантажити опис ESP8266 і зробити його доступним у середовищі. Після цього в меню Tools - Board - Boards Manager необхідно знайти «esp8266 by ESP8266 Community» і натиснути Install. Після встановлення у списку плат з'явиться перелік моделей ESP8266, серед яких для цього проєкту зазвичай обирається NodeMCU 1.0 (ESP-12E Module) або аналогічна плата.

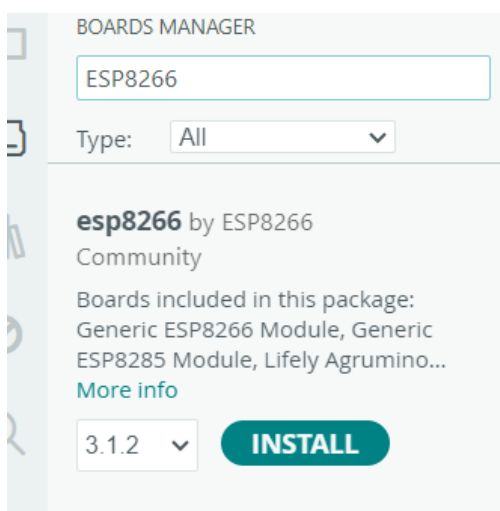


Рисунок 3.6 – Налаштування середовища Arduino IDE (завантаження модулю ESP8266)

Далі слід підключити ESP8266 до комп'ютера через USB-кабель. Arduino IDE автоматично розпізнає порт, але за потреби його можна обрати вручну через меню Tools → Port.

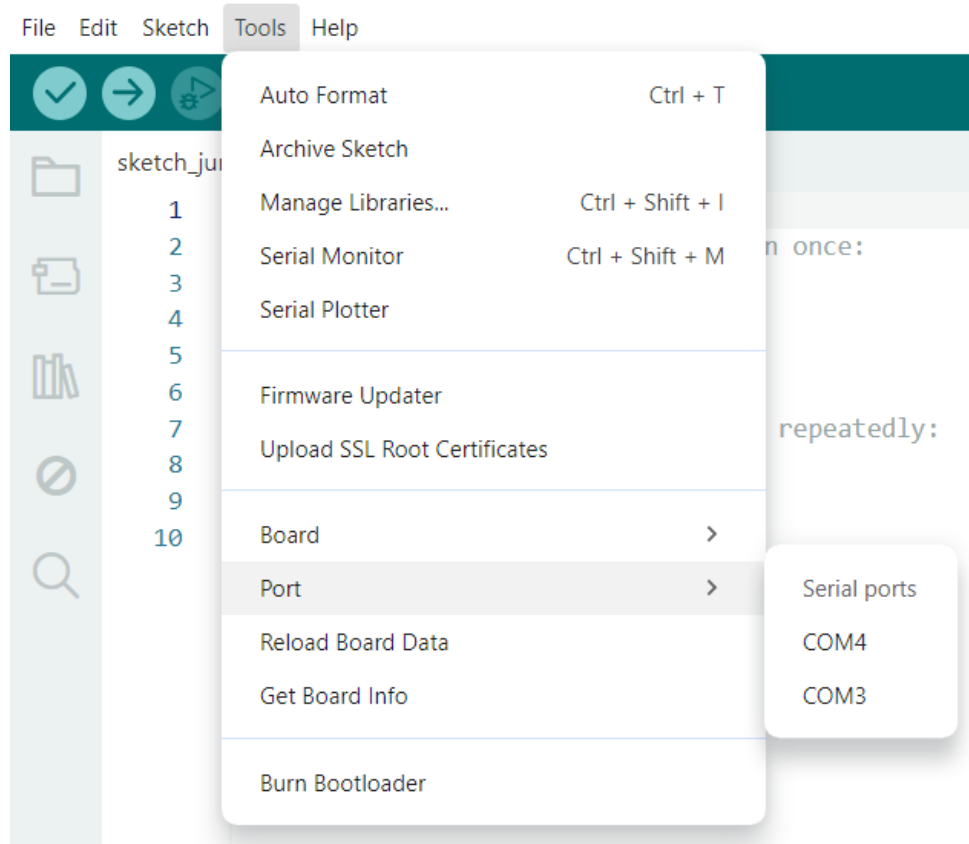


Рисунок 3.6 – Налаштування середовища Arduino IDE (вибір порту)

У випадку, якщо система не розпізнає пристрій, необхідно встановити драйвер для USB-UART перетворювача, який найчастіше базується на чипі CH340G або CP2102. Драйвери доступні на офіційних сайтах виробників або в архівах спільноти Arduino [12].

Після встановлення плати та драйверів студент може перейти до першого запуску. Для тестування правильності підключення створюється простий скетч, наприклад, виведення тексту в послідовний порт. Код виглядає так:

```
void setup() {
  Serial.begin(9600);
  Serial.println("ESP8266 готовий до роботи!");
}

void loop() {
}
```

Після компіляції та завантаження скетчу на плату (через кнопку Upload) відкривається Serial Monitor (інструмент для перегляду тексту, що виводиться з плати в реальному часі), де має з'явитись відповідне повідомлення. Цей етап є першим кроком до взаємодії з системою і дозволяє переконатися, що середовище правильно налаштоване.

Наступні кроки студент виконує самостійно або під керівництвом викладача: підключення сенсорів (наприклад, DHT22 або фоторезистора), зчитування значень з них через відповідні бібліотеки (DHT.h, ESP8266WiFi.h, ESP8266WebServer.h), обробка даних, програмування логіки автоматизації (реакції на умови) та керування пристроями (вивід сигналу на реле або LED).

Ключовим елементом є налаштування Wi-Fi-підключення, через яке система надсилає дані на вебінтерфейс. Студенти вписують параметри мережі (SSID і пароль) безпосередньо у код, після чого плата підключається до локальної мережі й запускає вбудований вебсервер, що дає змогу отримати доступ до керування системою через звичайний браузер. Приклад коду з налаштованими параметрами локальної мережі наведено в додатку (додаток А). У серійному моніторі відображається IP-адреса пристрою, яку вводять у браузер для взаємодії з моделлю.

Після завершення програмування та підключення всіх компонентів необхідно провести повне тестування функціональності. Студенти здійснюють покрокову перевірку кожного сенсора та відповідної реакції системи. Наприклад, для перевірки температурного сенсора необхідно змінити температуру навколишнього середовища, і переконатися, що після досягнення порогу вмикається відповідний індикатор. Тестування освітленості виконується шляхом затемнення датчика — наприклад, накривання пальцем, після чого система повинна активувати світлодіод або подати сигнал про низьке освітлення. Усі ці дії фіксуються в серійному моніторі, а також у вебінтерфейсі, що дозволяє перевірити логіку реагування системи.

Також тестується взаємодія з користувачем: надсилання HTTP-запитів через браузер, отримання відповідей від контролера, оновлення сторінки зі

станом системи. Важливо, що всі дії студентів супроводжуються вивченням структури програмного коду: як написано обробник запиту, які змінні зчитуються, як формується відповідь.

Висновки до третього розділу

У третьому розділі було детально описано процес практичної реалізації навчальної моделі системи розумного будинку, починаючи з налаштування середовища Arduino IDE для роботи з платою ESP8266, встановлення бібліотек і драйверів, програмування логіки керування сенсорами та пристроями, і завершуючи розробкою вебінтерфейсу для взаємодії з користувачем. Описано процес створення фізичної моделі будинку за допомогою 3D-друку на принтері, запаюванню компонентів, монтуванню сенсорів у кімнати та організації внутрішнього простору моделі.

ВИСНОВКИ

У ході роботи було розроблено, змодельовано та реалізовано функціональну фізичну систему, призначену для використання в освітньому процесі з метою формування у студентів практичних навичок побудови, налаштування та програмування IoT-рішень.

У першому розділі роботи було здійснено аналіз предметної області та обґрунтовано актуальність використання розумних систем у контексті сучасного освітнього середовища. Розглянуто принципи функціонування систем розумного будинку, проаналізовано інформаційні потреби навчальної моделі та сформульовано функціональні вимоги до її реалізації. Це дало змогу окреслити основні технічні та навчальні цілі, яких слід досягти під час побудови системи.

У другому розділі було розроблено архітектуру системи, описано логіку функціонування, виконано інфологічне моделювання та представлено фізичну схему у симуляторі яка демонструє, сенсори, та виконавчі пристрої взаємодіють у межах макету.

У третьому розділі було детально описано взаємодію студентів з системою через середовище Arduino IDE. Описано порядок встановлення платформи, підключення необхідних бібліотек, написання програмного коду, налагодження логіки роботи з сенсорами та пристроями. Наведено інструкцію користувача, типові помилки та шляхи їх усунення. Також було реалізовано 3D-моделі кімнат, які надруковано на принтері, виконано монтаж сенсорів і організацію внутрішнього простору макета. Система стабільно виконує функції збору даних, аналізу та прийняття рішень у реальному часі, а також забезпечує зворотний зв'язок із користувачем через вебінтерфейс.

Найважливішим результатом виконання цієї кваліфікаційної роботи є створення доступного, компактного та гнучкого навчального засобу для вивчення основ Інтернету речей.

Результати цієї роботи можуть бути застосовані для вирішення таких проблем: підвищення якості практичної підготовки студентів у галузі

інформаційних технологій та автоматизації; впровадження прикладних моделей у лабораторні заняття з IoT, електроніки, комп'ютерної інженерії; створення умов для реалізації міждисциплінарних освітніх проєктів. Модель є відкритою до модифікацій та може бути використана як основа для подальшого розвитку – наприклад, додавання нових типів сенсорів, впровадження мобільного застосунку чи підключення до хмарних сервісів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ashton, K That “Internet of Things” Thing: In the Real World Things Matter More than Ideas. RFID Journal. 2009 URL:
<http://www.rfidjournal.com/articles/view?4986> (дата звернення: 13.05. 2025)
2. Evans D. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. Cisco Internet Business Solutions Group (IBSG). 2011.
3. Aldrich F. Smart Homes: Past, Present and Future // Inside the Smart Home. 2003.
4. Ayo B. S. Development of a Home Automation System Using Wireless Sensor/Actuator Nodes. 2018. URL: <https://www.academia.edu/38707094> (дата звернення: 16.05.2025).
5. Розумне місто: використання IoT-датчиків. ProIT : веб-сайт. URL:
<https://proit.com.ua/news/rozumnyj-gorod-vykorystannya-iot-datchykyiv/> (дата звернення: 02.05.2025).
6. Мусіченко О.Є. Розробка програмного забезпечення мікроконтролерів на основі ESP8266 з підтримкою бездротової мережі Wi-Fi: бакалаврська робота, спеціальність 123 «Комп'ютерна інженерія» / О.Є. Мусіченко. Запоріжжя : НУ «Запорізька політехніка», 2024. 70 с.
7. Конфігурація проєкту ESP8266. Tseivo : веб-сайт. URL:
<https://tseivo.com/b/memecode/t/bgpqoanv0r> (дата звернення: 02.06.2025).
8. WokWi ESP8266 Simulation Project. WokWi : веб-сайт. URL:
<https://wokwi.com/projects/427685981074897921> (дата звернення: 03.06.2025).
9. ESP32 Microcontroller Guide. Ariat Tech : веб-сайт. URL: <https://ua.ariat-tech.com/blog/ESP32-Microcontroller-Guide.html> (дата звернення: 08.06.2025).
10. Arduino бібліотеки: налаштування і використання. ITMaster : веб-сайт. URL: <https://itmaster.biz.ua/electronics/arduino/arduino-lib.html> (дата звернення: 12.05.2025).

11. Програмування ESP32 через Arduino. ITMaster : веб-сайт. URL: <https://itmaster.biz.ua/electronics/esp32/esp32-arduino.html> (дата звернення: 08.06.2025).
12. Програмування ESP8266 у середовищі Arduino IDE. ITMaster : веб-сайт. URL: <https://itmaster.biz.ua/electronics/esp8266/esp8266-arduino.html> (дата звернення: 08.06.2025).
13. Balyk N., Leshchuk S., Yatsenyak D. Design and implementation of an IoT-based educational model for smart homes: a STEM approach. // Journal of Edge Computing. – 2023. – Vol. 2, № 2. – С. 148–162.
14. Kumar S., Tiwari P., Zymbler M. Internet of Things is a revolutionary approach for future technology enhancement: a review. // Journal of Big Data. – 2019. – Vol. 6, Article 111.
15. Dionysios G. et al. A Comprehensive Review of IoT Networking Technologies for Smart Home Automation Applications. // Sensors. – 2022. – Vol. 12, № 2. – Article 30.
16. Yang Y. et al. Visualizing the Landscape of Home IoT Research: A Bibliometric Analysis Using VOSviewer. // Sensors. – 2023. – Vol. 23, № 6. – Article 3086.
17. Burroughs J. Unleashing the Power of IoT: A Comprehensive Review. // Sensors. – 2023. – Vol. 23, № 16. – Article 7194.
18. Jacobsson A., Boldt M., Carlsson B. A risk analysis of smart home automation systems. // Future Generation Computer Systems. – 2016. – Vol. 56. – С. 719–733.
19. Kim H. et al. IoT security: review, blockchain solutions, and open challenges. // Future Generation Computer Systems. – 2018. – Vol. 82. – С. 395–411.
20. Marikyan D., Papagiannidis S., Alamanos E. A systematic review of the smart energy conservation system: from smart homes to sustainable smart cities. // Renewable and Sustainable Energy Reviews. – 2021. – Vol. 134. – Article 110217.

21. Moyle W. et al. A systematic review of the smart home literature: a user perspective. // *Technological Forecasting and Social Change*. – 2019. – Vol. 138. – C. 139–154.
22. Nikou S. The effectiveness of smart home technologies to support health outcomes. // *International Journal of Medical Informatics*. – 2021. – Vol. 151. – Article 104474.
23. Rahimi M. et al. Factors driving the adoption of smart home technology: an empirical assessment. // *Telematics and Informatics*. – 2019. – Vol. 38. – C. 118–130.
24. Sovacool B. K. et al. Fog-based smart homes: a systematic review. // *Journal of Network and Computer Applications*. – 2020. – Vol. 172. – Article 102736.
25. Wilson G. et al. Smart home technologies in Europe: a critical review of challenges. // *Renewable and Sustainable Energy Reviews*. – 2020. – Vol. 120. – Article 109660.
26. Wilson C. et al. Robot-enabled support of daily activities in smart home environments. // *Cognitive Systems Research*. – 2019. – Vol. 55. – C. 229–241.
27. Büyük M. et al. Benefits and risks of smart home technologies. // *Energy Policy*. – 2017. – Vol. 110. – C. 12–18.
28. Chen L. et al. Overview of smart home concepts through energy management systems. // *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*. – 2022. – Vol. 44, № 15. – C. 2905–2917.
29. Afroz A., Khamari S. S., Behera R. K. Solar Powered Smart Home Automation and Smart Health Monitoring with IoT. // In: Udgata S. K. et al. (eds). *Intelligent Systems*. – Springer, 2024. – Lecture Notes in Networks and Systems, vol. 826.
30. Hall F., Maglaras L., Aivaliotis T. et al. Smart Homes: Security Challenges and Privacy Concerns. // arXiv preprint. – 2020. URL: <https://arxiv.org/abs/2005.09985> (дата звернення: 01.06.2025).

31. Bouchabou D., Nguyen S. M. et al. A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors. // arXiv preprint. – 2021. URL: <https://arxiv.org/abs/2106.07775> (дата звернення: 01.06.2025)
32. Jacobsson A., Boldt M., Carlsson B. “Home is where the smart is”? Evaluating smart home research from a user perspective. // Energy Research & Social Science. – 2018. – Vol. 37. – С. 94–101.
33. Діаграма активності (Activity diagram): веб-сайт. URL: <http://um.co.ua/1/1-7/1-73653.html> (дата звернення: 01.06.2025)
34. Діаграма станів : веб-сайт. URL: <http://surl.li/ybhxws> (дата звернення: 01.06.2025)
35. Діаграма станів: веб-сайт. URL: www.wikidata.uk-ua.nina.az (дата звернення: 01.06.2025)
36. Лекція 6. Нотація IDEF0: веб-сайт. URL: <http://surl.li/njvera> (дата звернення: 01.06.2025)
37. Еталонна архітектура Microsoft Azure IoT : веб-сайт. URL: https://learn.ztu.edu.ua/pluginfile.php/204963/mod_resource/content/1/201.pdf (дата звернення: 01.06.2025)
38. «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» : веб-сайт. URL: <https://dspace.nuft.edu.ua/server/api/core/bitstreams/4c927394-ae5d-4c7e-adc8-063d6a23a3c7/content> (дата звернення: 01.06.2025)
39. Навчальний посібник для студентів вищих навчальних закладів: веб сайт. URL: <https://vseosvita.ua/library/embed/01003qnw-c7a5.doc.html> (дата звернення: 01.06.2025)
40. Діаграми Прецедентів (Use Case UML Diagram) : веб-сайт. URL: <https://lvivqaclub.blogspot.com/2008/10/use-case-uml-diagram.html> (дата звернення: 01.06.2025)

ДОДАТКИ

ДОДАТОК А

```
#include <ESP8266WiFi.h>
#include <ESPAsyncWebServer.h>
#include <Adafruit_NeoPixel.h>
#include <Adafruit_HTU21DF.h>

#define LDR_PIN 14 // D5
#define LED_PIN 12 // D6
#define NUM_LEDS 2

Adafruit_NeoPixel strip(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);
Adafruit_HTU21DF htu = Adafruit_HTU21DF();
AsyncWebServer server(80);

bool useLDR = false;
int tempThreshold = 22;
int led1ColorR = 255, led1ColorG = 255, led1ColorB = 255;
int brightness = 255;

float tempCache = 0.0;
float humCache = 0.0;
unsigned long lastSensorRead = 0;

void setup() {
  Serial.begin(115200);
  pinMode(LDR_PIN, INPUT);
  strip.begin();
  strip.setBrightness(brightness);
  strip.show();

  if (!htu.begin()) {
    Serial.println("HTU21 not found!");
  }

  WiFi.softAPConfig(IPAddress(192,168,4,1), IPAddress(192,168,4,1), IPAddress(255,255,255,0));
  WiFi.softAP("ESP-Освітлення", "12345678");
  Serial.print("AP IP address: ");
  Serial.println(WiFi.softAPIP());

  // ... (весь код той самий до server.on("/"), а далі оновлена HTML частина)
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    String html = R"rawliteral(
    <html><head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Розумна Спальня</title>
    <style>
    body {
      font-family: Arial, sans-serif;
      background: #f5f5f5;
      color: #333;
      padding: 20px;
      max-width: 500px;
      margin: auto;
    }
    h1 {
```

```

    text-align: center;
    color: #444;
  }
  .card {
    background: white;
    border-radius: 10px;
    padding: 20px;
    box-shadow: 0 4px 10px rgba(0,0,0,0.1);
    margin-bottom: 20px;
  }
  label, input, span {
    font-size: 1.1em;
  }
  input[type="color"],
  input[type="range"],
  input[type="number"] {
    width: 100%;
    margin-top: 5px;
  }
</style>
</head><body>
<h1>Розумна Спальня</h1>

<div class="card">
  <p>🌡 Температура: <span id="temp">...</span> °C</p>
  <p>💧 Вологість: <span id="hum">...</span> %</p>
</div>

<div class="card">
  <label><input type='checkbox' id='useLDR' onchange='toggleLDR()'> Використовувати датчик
освітлення</label>
  <label>😊 Колір освітлення</label>
  <input type='color' id='ledColor' value='#ffffff' onchange='setColor()'>
  <label>👁 Яскравість</label>
  <input type='range' min='0' max='255' id='brightness' value='255' onchange='setBrightness()'>
</div>

<div class="card">
  <h3>🕒 Керування обігрівачем</h3>
  <label>Температура вмикання: <input type='number' id='tempVal' value='22' onchange='setTemp()'>
°C</label>
</div>

<script>
function toggleLDR() {
  fetch('/useLDR?val=' + document.getElementById('useLDR').checked);
}
function setColor() {
  let c = document.getElementById('ledColor').value.substring(1);
  fetch('/color?r=' + parseInt(c.substring(0,2),16) + '&g=' + parseInt(c.substring(2,4),16) + '&b=' +
parseInt(c.substring(4,6),16));
}
function setBrightness() {
  fetch('/brightness?val=' + document.getElementById('brightness').value);
}
function setTemp() {
  fetch('/temp?val=' + document.getElementById('tempVal').value);
}
function updateSensorData() {
  fetch('/sensors').then(r => r.json()).then(d => {

```

```

        document.getElementById('temp').innerText = d.temp;
        document.getElementById('hum').innerText = d.hum;
    });
}
setInterval(updateSensorData, 2000);
updateSensorData();
</script>
</body></html>
)rawliteral";
request->send(200, "text/html", html);
});

// Додай повернення хендлера tempThreshold:
server.on("/temp", HTTP_GET, [](AsyncWebServerRequest *request){
    tempThreshold = request->getParam("val")->value().toInt();
    request->send(200, "text/plain", "Threshold set");
});

server.on("/useLDR", HTTP_GET, [](AsyncWebServerRequest *request){
    useLDR = request->getParam("val")->value() == "true";
    request->send(200, "text/plain", "OK");
});

server.on("/color", HTTP_GET, [](AsyncWebServerRequest *request){
    led1ColorR = request->getParam("r")->value().toInt();
    led1ColorG = request->getParam("g")->value().toInt();
    led1ColorB = request->getParam("b")->value().toInt();
    request->send(200, "text/plain", "Color updated");
});

server.on("/brightness", HTTP_GET, [](AsyncWebServerRequest *request){
    brightness = request->getParam("val")->value().toInt();
    strip.setBrightness(brightness);
    request->send(200, "text/plain", "Brightness updated");
});

server.on("/sensors", HTTP_GET, [](AsyncWebServerRequest *request){
    String json = "{\"temp\":\"" + String(tempCache, 1) + "\", \"hum\":\"" + String(humCache, 1) + "\"}";
    request->send(200, "application/json", json);
});

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

void loop() {
    // Оновлення сенсора кожну секунду
    if (millis() - lastSensorRead > 1000) {
        tempCache = htu.readTemperature();
        humCache = htu.readHumidity();
        lastSensorRead = millis();
    }

    // 1-й світлодіод — освітлення
    if (useLDR && digitalRead(LDR_PIN) == LOW) {
        strip.setPixelColor(1, 0, 0, 0); // темно — вимк
    } else {
        strip.setPixelColor(1, led1ColorR, led1ColorG, led1ColorB); // включено
    }

    // 2-й світлодіод — обігрівач

```

```
if (tempCache < tempThreshold) {  
    strip.setPixelColor(0, 255, 0, 0); // червоний  
} else {  
    strip.setPixelColor(0, 0, 0, 255); // синій  
}  
  
strip.show();  
delay(50);  
}
```