Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Система керування внутрішнім середовищем розумного будинку на базі мікроконтролерів

Виконав: студент VI курсу, групи СНнм-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

_____ Зеленський К.К.
(підпис) (прізвище та ініціали)

Керівник _____ Литвиненко Я.В.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Никитюк В.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент _____ Тиш Є.В.
(підпис) (прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«____» 29 травня 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Зеленський Костянтин Костянтинович
(прізвище, ім'я, по батькові)

1. Тема роботи Система керування внутрішнім середовищем розумного будинку на базі мікроконтролерів

Керівник роботи Литвиненко Ярослав Володимирович, д.т.н., професор кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 24 » листопада 2023 року № 4/7-1100

2. Термін подання студентом завершеної роботи 30 травня 2024р.

3. Вихідні дані до роботи Наукові публікації на тему Давачі, які застосовують в розумному Будинку та Огляд мікро контролерів для побудови Розумного будинку

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1 Аналіз предметної області, огляд можливостей мікроконтролерів.
2 Розробка сайту Smart Home Dashboard
3 Розробка та опис роботи розумного будинку
4 Охорона праці та безпека в надзвичайних ситуаціях. Висновки. Перелік джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1.Вступ, 2. Мета дослідження, 3.Розумний будинок, 4. Огляд мікроконтролерів Arduino, 5. Огляд мікроконтролерів Raspberry Pi, 6. Огляд Raspberry Pi OS, 7. Таблиця відмінностей між Arduino і Raspberry Pi, 8 Схема системи «Розумний будинок», 9. Огляд плат DI/DO та ESP32, 10. Огляд початкогово вікна стайу Smart Home Dashboard, 11. Приклади датчиків, які використовуються в системах «Розумного будинку», 12. Висновки, 13. Кінцевий слайд

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викладач		

7. Дата видачі завдання 24 листопада 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	04.12.2023	*Виконано*
2.	Підбір наукових джерел на тему розумний будинок	15.12.2023-31.11.2023	*Виконано*
3.	Опрацювання наукових публікацій та збір даних по	15.01.2024-25.02.2024	*Виконано*
	темі роботи		
4.	Виконання дослідження згідно мети кваліфікаційної	26.02.2024-07.04.2024	*Виконано*
	роботи		
5.	Оформлення розділу «Аналіз предметної області,	15.04.2024-18.04.2024	*Виконано*
	огляд можливостей мікроконтролерів»		
6.	Оформлення розділу «Розробка сайту Smart Home	19.04.2024-25.04.2024	*Виконано*
	Dashboard»		
7.	Оформлення розділу «Розробка та опис роботи	26.04.2024-02.05.2024	*Виконано*
	розумного будинку		
8.	Виконання завдання до підрозділу «Охорона праці»	03.05.2024-07.05.2024	*Виконано*
9.	Виконання завдання до підрозділу «Безпека в	08.05.2024-10.05.2024	*Виконано*
	надзвичайних ситуаціях»		
10.	Оформлення кваліфікаційної роботи	11.05.2024-14.05.2024	*Виконано*
11.	Нормоконтроль	15.05.2024-16.05.2024	*Виконано*
12.	Перевірка на плагіат	17.05.2024	*Виконано*
13.	Попередній захист кваліфікаційної роботи	21.05.2024	*Виконано*
14.	Захист кваліфікаційної роботи	30.05.2024	

Студент _____ Зеленський К.К.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Литвиненко Я.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Система керування внутрішнім середовищем розумного будинку на базі мікроконтролерів // Кваліфікаційна робота освітнього рівня «Магістр» // Зеленський Костянтин Костянтинович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНнм-61 // Тернопіль, 2024 // С. 95, рис. – 36, табл. –2, додат. – 2, бібліогр. – 53.

Ключові слова: розумний будинок, сайт, мікроконтролери, автоматизація будинку, Raspberry Pi, Arduino, безпека розумного будинку, системи керування будинком.

Кваліфікаційна робота присв'ячена розробці Системі керування внутрішнім середовищем розумного будинку на базі мікроконтролерів.

У першому розділі кваліфікаційної роботи описані можливості мікроконтролерів, проаналізовано їх переваги та недоліки.

У другому розділі кваліфікаційної роботи розроблено сайт для моніторингу «Розумного будинку».

У третьому розділі кваліфікаційної роботи описано систему «Розумного будинку», наведено приклади підключення різних датчиків. Розглянуто значення «Розумних будинків».

ABSTRACT

Smart Home Indoor Environment Control System Based on Microcontrollers // Master's Degree Qualification Work // Konstantin Kostiantynovych Zelensky // Ternopil National Technical University named after Ivan Puluj, Faculty of Computer Information Systems and Software Engineering, Department of Computer Sciences, group CSnnm-61 // Ternopil, 2024 // p. 95, tables – 2, drawings – 36, appendices – 2, bibliography – 53.

***Keywords*:** smart home, website, microcontrollers, home automation, Raspberry Pi, Arduino, smart home security, home control systems.

The qualification paper is dedicated to the development of an Internal Environment Management System for a smart home based on microcontrollers.

In the first section of the qualification paper, the capabilities of microcontrollers are described, and their advantages and disadvantages are analyzed.

In the second section of the qualification paper, a website for monitoring the "Smart Home" has been developed.

In the third section of the qualification paper, the "Smart Home" system is described, examples of connecting various sensors are provided, and the significance of "Smart Homes" is discussed.

ЗМІСТ

Додаток А

Додаток Б

ВСТУП

Актуальність теми. У сучасному світі інтернету речей (IoT) концепція «Розумного будинку» стає дедалі популярнішою, пропонуючи користувачам підвищений контроль та зручність у щоденному житті. Реалізація таких систем вимагає комплексного підходу до інтеграції різноманітних технологій та пристроїв. В цьому контексті Raspberry Pi виступає як ідеальний кандидат для управління та координації розумних систем дому завдяки своїй гнучкості, доступності та великому співтовариству користувачів.

Мета і задачі дослідження. Метою цього проекту є створення веб-сайту для моніторингу параметрами «Розумного будинку», який буде працювати разом з Raspberry Pi. Цей сайт дозволяє користувачам відслідковувати стан різних систем будинку, таких як освітлення, температура, безпека та інші автоматизовані функції в режимі реального часу. Для досягнення поставленої мети потрібно викоанти ряд завдань, зокрема:

— Проаналізувати стан досліджень в області систем керування «Розумного будинку»;

— Дослідити існуючі на даний час методи реалізації подібних систем;

— Проаналізувати найпопулярніші методи втілення подібних систем;

— Виконати порівняння мікроконтролерів на базі Arduino і Raspberry Pi;

— Розробити сайт моніторингу системи «Розумний будинок» та безпосередньо саму систему.

Об’єкт дослідження. Процес керування «Розумним будинком».

Предмет дослідження. Мікроконтролери та системи для керування «Розумний будинок».

Практичне значення одержаних результатів. Розроблено сайт моніторингу системи «Розумний будинок» та саму систему на базі мікроконтролерів Raspberry Pi.

Наукове значення. Проведений аналіз мікроконтролерів, який показав оптимальний варіант для побудови систем керування внутрішнім середовищем розумного будинку.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на XI науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 2023 р.).

Публікації. Основні результати кваліфікованої роботи опубліковано у двох працях конференції (див. перілк джерел № 24 та № 25).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 53 найменувань та 2 додактів. Загальний обсяг кваліфікаційної роботи складає 97 сторінки, з них 73 сторінки основого тексту, який містить ?? рисунків та 2 таблиці.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, ОГЛЯД МОЖЛИВОСТЕЙ МІКРОКОНТРОЛЕРІВ

1.1 Огляд можливостей систем «Розумного будинку»

Управління енергією є запорукою повноцінного співіснування людини з навколишнім середовищем, а також безпеки на виробництві та в повсякденному житті. Серед аспектів даної концепції виділяється управління освітленням. Використовувана на освітлення частка енергії є досить суттєвою, і раціональна експлуатація приладів здатна значно зменшити ці витрати, без втрат всіх переваг. Вирішити це завдання допомагають системи управління освітленням, які також можна інтегрувати в комплекс автоматизації різних будівель.

Найважливішим завданням таких систем є підвищення ефективності, також вони забезпечують високий рівень побутового та виробничого комфорту шляхом підвищення експлуатаційних характеристик будь-яких будівель. Система управління освітленням може включати в себе і автономні рішення, здатні самостійно підлаштовуватися під ситуацію і налаштування, щоб своєчасно змінювати параметри функціонування освітлення. Автоматизовані рішення для управління освітленням забезпечують виконання таких функцій [1]:

- оцінка рівня природної освітленості в приміщеннях,
- облік часу доби і дня тижня,
- перевірка наявності людей в кімнаті.

Сучасні вимоги ринку освітлювальних приладів припускають повсюдну заміну аналогового управління цифровим. Ключова перевага цифрових моделей полягає в гнучкій комунікації та інтеграції з іншими пристроями, встановленні зв’язку між окремими приладами, що входять до складу системи. Часто для цього не потрібні окремі дроти, а передача здійснюється за допомогою силових кабелів. Також велике поширення сьогодні знаходять бездротові технології.

Система розумного освітлення в сучасному будинку складається з кількох компонентів: освітлювальних приладів та електронних систем керування. Для підвищення ефективності керування LED-освітленням у «Розумному будинку» можна рекомендувати наступні технології [2].

1) Стандартний RGB-контролер. Одноканальний пристрій для управління трьома кольорами, має ряд керуючих програм, контроль над якими здійснюється за допомогою клавіш на пристрої і пульті управління. Для підвищення числа адрес можна використовувати репітери [1].

2) Багатоканальний RGB-контролер. Три і більше каналів управління RGB-кольорами, з можливістю об'єднання в систему для виконання світлодинамічних сценаріїв. Програмування можна здійснювати через персональний комп'ютер (ПК).

3) DMX RGB-контролер. Пристрій для управління освітленням за допомогою протоколу DMX 512, підключається по USB до комп'ютера. Даний протокол дозволяє підключати до 512 каналів приладів, з якими потім можна реалізовувати певні сценарії освітлення.

4) DALI RGB-контролер. Рішення з використанням протоколу DALI і двонаправленим обміном цифровими даними з пристроями системи. Число адрес всередині системи може досягати 12800, об'єднаних в 200 ліній за допомогою спеціальних роутерів.

Розумне освітлення має багато корисних властивостей, наприклад, система зможе імітувати присутність, якщо господарі у відрядженні, автоматично знижувати яскравість світла, коли працює телевізор. Систему можна запрограмувати виконувати функції будильника, подавати сигнали про вхідні дзвінки та повідомлення. Лампи в подібній системі можуть вмикатися автоматично, коли власник повертається додому, та за допомогою голосових команд [3].

Комбінуючи систему освітлення та сенсори руху, можна регулювати роботу світильників у коридорі, ванній кімнаті. Подібна комбінація дозволить суттєво підвищити рівень комфорту та ефективність використання електроенергії,

знизивши затрати на оплату рахунків. Сенсори руху просто не допустять можливості забути вимкнути світло, адже автоматично вимкнуть його, якщо в кімнаті не зафіксовано рух. Окрім того, таке керування системою позбавить необхідності шукати в темряві вимикач – людина тільки заходить в кімнату або виходить на сходовий майданчик, а світло вже горить [1]

Поєднання бездротових технологій та енергоефективних ламп в одній системі, дозволяє змінювати не тільки наш звичний уклад життя, а й вигідно впливати на інтер'єр. Одна з ключових переваг системи розумного освітлення – можливість створення світлових сценаріїв. За допомогою світлових сценаріїв можна створювати оригінальні світлові гамми, тіньові переходи, варіювати яскравістю освітлення.

У цій системі можна використовувати популярні RGB світильники, в яких змінюється колір освітлення. Такі прилади, переважно, застосовуються для освітлення карнизів, шафок, можуть встановлюватися під ліжком, для створення оригінального ефекту. Один маленький вимикач дозволить плавно змінювати колір підсвічування.

Не менш важливий компонент системи розумного освітлення – світильники зі змінною колірною температурою. Такі прилади дуже поширені для створення ідеальних умов праці та відпочинку. Завдяки підключенню таких ламп до системи, можна ефективно працювати при яскравому освітленні, а відпочивати при м'якому, комфортному теплому світлі. З виникненням LED-технологій, управління світлодіодним освітленням взагалі стало легким і приємним процесом [3].

Використовувана на освітлення частка енергії є досить суттєвою, і раціональна експлуатація приладів здатна значно зменшити ці витрати, без втрат всіх переваг. Вирішити це завдання допомагають системи управління освітленням, які також можна інтегрувати в комплекс автоматизації різних будівель.

1.1.1. Система клімат-контроль

Система клімат-контролю працює на підставі закладених у неї алгоритмів, що дозволяють підтримувати встановлені параметри повітря серед різних кліматичних зон в приміщеннях при мінімальних затратах енергоресурсів. Система дозволяє забезпечувати виконання різних операцій. З її допомогою проводиться нагрів або охолодження. При цьому виключається одночасна робота кондиціонера і системи опалення. Винятком може бути наявність теплої підлоги, підтримуючого встановлену температуру в нижній частині кондиціонованого приміщення [2].

Клімат-контроль забезпечує зниження температури в нічний час в безлюдних приміщеннях і спальнях, що дозволяє створити комфортні умови для сну, а також економити енергоресурси. Крім того, він дає можливість мінімізувати роботу апаратури і обладнання під час відсутності господарів за допомогою використання режимів роботи «денна відсутність» і «відпустка». При включенні другого режиму проводиться повне відключення системи кондиціонування та вентиляції, а опалювальна система виводиться на мінімальний рівень потужності. Перед поверненням додому можна завчасно встановити в приміщеннях комфортний кліматичний режим шляхом активації системи клімат-контролю по телефону або через інтернет.

Система управління кліматом в приміщенні дає можливість коригувати рівень температури, вологості, величину притоку свіжого повітря індивідуально для кожного приміщення, управляти роботою системи фільтрації повітря, створювати індивідуальну кліматичну систему для кожного члена сім'ї, погоду в будинку (наприклад, відсутність протягів при постійно свіжому повітрі в дитячій кімнаті). В той же час, система клімат-контролю, незважаючи на виконання великої кількості функцій, вирішує проблему енергозбереження. Наприклад, систему можна налаштувати таким чином, щоб у вихідні дні та неробочий час подача тепла в приміщення скорочувалася або відключалася зовсім. Такий режим

роботи особливо актуальний для використання в заміських котеджах із автономним опаленням [11]. Зазначена система дозволяє дистанційно включати котел опалення або перемикати його в режим економії. З метою більш ефективної і раціональної організації життєдіяльності офісів можливо встановлення контролю над станом комунікацій теплопостачання, електропостачання, водопостачання, створення найбільш комфортних умов роботи для працівників компанії.

Система клімат-контролю «Розумного будинку» виключить можливість псування колекції картин, книг або вин шляхом створення найбільш сприятливих умов для їх зберігання.

Для забезпечення коригування параметрів роботи системи застосовуються різні датчики, які фіксують поточні показники мікроклімату в приміщеннях будинку, а також засоби для управління у вигляді перемикачів і панелей. При їх використанні система здатна управляти якістю повітря (температурою, вологістю, озонуванням) відповідно до пори року і доби, режимом провітрювання з використанням автоматичної системи відкривання вікон, змінювати режим роботи радіаторів опалення та теплої підлоги, автоматично підтримувати температуру і вологість у спеціальних приміщеннях, а також аварійно зупиняти систему опалення [14].

Таким чином, система клімат-контролю «Розумного будинку» дозволяє створити здоровий і комфортний мікроклімат для затишного проживання в будинку.

1.1.2. Системи безпеки і відеоспостереження

Постановка і зняття квартири з охорони виконується за допомогою кодової панелі, розміщеної у тамбурі. При відкритті вхідних дверей у людини є 30 секунд на введення правильного коду. Якщо ж код не буде введений, «Розумний будинок» включить сирени і відправить СМС-повідомлення на кілька телефонних номерів.

Датчики руху, розташовані на кухні, спальні і вітальні дозволять виявити проникнення через вікна. Схема застосування датчиків руху в квартирі з інтелектуальною системою «Розумний будинок» показана на рис. 1.1.

При виході з квартири достатньо ввести код на охоронній панелі, і «Розумний будинок» не тільки включить сигналізацію, але і відключить освітлення, переведе систему опалення в режим енергозбереження .

Система відеоспостереження дозволяє з будь-якого телевізора або пульта подивитися на гостей, дистанційно відкрити в'їзні ворота і впустити їх в будинок. Вона записує і дає можливість проглянути всі події, які сталися під час відсутності господарів.

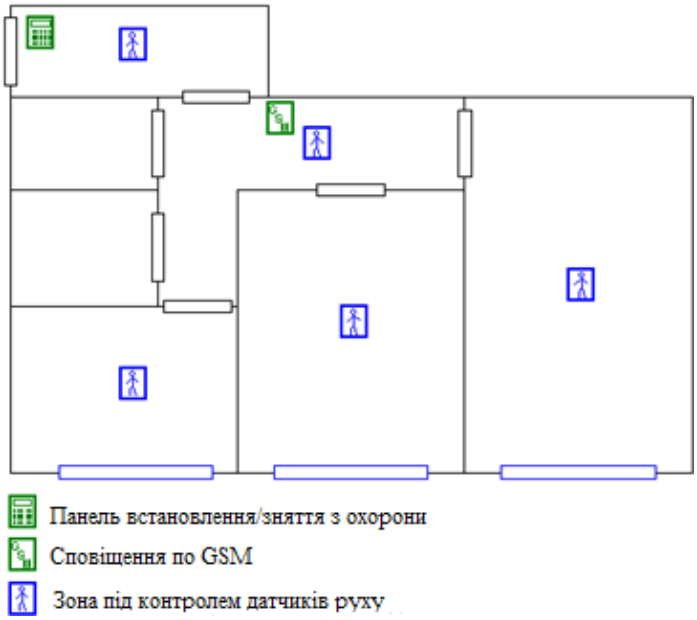


Рисунок 1.1 – Схема застосування датчиків руху в квартирі з системою «Розумний будинок»

Камери можуть автоматично наводитися на ту частину будинку або ділянки, де було помічено рух. За заданим сценарієм система може проводити як плановий, так і позаплановий моніторинг приміщень будинку і гаража.

У разі будь-якого порушення охоронного периметра сигналізація негайно проінформує власників про небезпеку, а при зломі – повідомить і в службу охорони.

Інтегрувавши її з системою відеоспостереження, можна робити запис всього, що відбувається [15].

Крім того, інтегрована система управління дозволяє зв’язати в єдину мережу датчик и пожежної безпеки, що знаходяться у всіх приміщеннях будинку. Вона не лише повідомить господарів про будь-які події, але і сама відключить електрику і перекриє газ. Як і у випадку з охоронною сигналізацією, можливе автоматичне інформування працівників пульта пожежного спостереження. Подібні системи забезпечення безпеки і охорони реалізуються, як правило, шляхом інтеграції із спеціалізованими охоронно-пожежними системами.

1.1.3. Контроль протікання води і витоку газу

Системи аварійної сигналізації – це, головним чином, датчики води і газу. При витоку відповідний датчик вмить повідомить про це центральний контролер, а той, у свою чергу, перекриває електроклапаном газ в будинку або воду в місці протікання. Одночасно будуть проінформовані власники і, якщо необхідно, аварійні служби .

Рисунок 1.2 – Схема використання датчиків протікання води

Виявити і запобігти витоку води так само допоможе «Розумний будинок». Схема використання датчиків протікання води показана на рис. 1.2. Контрольованими зонами є санвузли та кухня, тобто ті приміщення, де проходять труби водопостачання.

Прорив труби або перелив води через краї раковини фіксується за допомогою спеціальних датчиків. У випадку протікання «Розумний будинок» перекроїть доступ води в квартиру і відправить СМС-повідомлення на задані телефони .

1.1.4. Штучний інтелект в «Розумному будинку»

За даними компанії Cisco, до 2021 року близько 46% програм «Розумного будинку», що забезпечують автоматизацію, безпеку, відеоспостереження та інше, будуть працювати на базі штучного інтелекту. Доступність електронних пристроїв ще більш зросте, основна маса техніки в будинку, саду, гаражі буде керуватися зі смартфона або планшета [21]. Сучасні інтелектуальні системи, такі як Amazon Echo і Google home, об’єднують різні пристрої один з одним за допомогою інтернету речей (IoT), виконують голосові команди, але повна автоматизація домашнього простору все ще залишається завданням майбутнього .

Майбутнє домашньої автоматизації – це інтелектуальне середовище, здатне за певний час до повернення власника додому встановити потрібну температуру, включивши кондиціонер, відкрити штори або включити музику, почати варити каву і т. д. Звучить неймовірно, але вже сьогодні існують централізовані системи, здатні виконувати прості побутові завдання, поки господаря немає вдома.

Популярні сучасні смарт-системи :

1. Nest;
2. HomeKit;
3. Wink;

4. Z-Wave;
5. Zigbee;
6. SmartThings;
7. Brillo.

Технічні гіганти Amazon, Apple і Google роблять ставку на голосових помічників: системи «Розумного будинку» повинні забезпечувати комфорт всім без винятку, особливо допомоги потребують люди з інвалідністю або літні люди, яким важко ходити, управляти побутовими електричними приладами та гаджетами. Тому функція автоматизації вважається пріоритетною.

1.2 Мікроконтролери на базі Raspberry Pi

Використання мікроконтролерів для розміщення датчиків - економічний спосіб побудувати мережу датчиків. Але що робити, коли вам потрібна більша обчислювальна потужність, ніж може забезпечити мікроконтролер? Якщо вам потрібно перетворити дані в інший формат, включити дані в додаток або надрукувати жорсткі копії даних датчика? У таких ситуаціях вам, ймовірно, знадобиться комп'ютер, який має більше обчислювальної потужності, може дозволити використання звичайних додатків, дозволяє використовувати скриптові мови та надає доступ до периферійних пристроїв [6].

Хоча особисті комп'ютери відносно недорогі, існують чіткі недоліки використання їх у ваших мережах датчиків - особливо як датчикові вузли. Якщо датчики розташовані в областях, де мережеве живлення ненадійне або недоступне, або де існує ризик перегріву, або де просто немає місця для установки особистого комп'ютера, вам доведеться або передавати дані на інший вузол для обробки, або зберігати їх локально та обробляти пізніше [30].

Проте, існує ще одне обмеження використання особистого комп'ютера як датчикового вузла: особистий комп'ютер не має загальних входів/виходів (I/O)

портів. Ви можете придбати розширювальні карти для збору даних, але вони часто призначені для використання в серверних або настільних комп’ютерах. Якщо врахувати вартість комп’ютера та карти для збору даних, вартість датчикового вузла стає невигідною.

То що робити в цих випадках? Якби тільки існував недорогий комп’ютер з достатньою обчислювальною потужністю та пам’яттю, який використовував стандартні периферійні пристрої, підтримував програмовані порти введення/виведення та мав невеликий форм-фактор. Саме це може зробити Raspberry Pi.

Raspberry Pi - це невеликий, недорогий персональний комп’ютер. Хоча він і не має можливості розширення пам’яті та не може приймати вбудовані пристрої, такі як CD, DVD та жорсткі диски, він має все, що потрібно простому персональному комп’ютеру. Тобто в нього є два USB-порти, Ethernet-порт, HDMI (і комбінований) відеовихід, а також аудіо роз’єм для звуку[6].

Raspberry Pi має пристрій зчитування SD, який можна використовувати для завантаження комп’ютера в будь-яку з кількох операційних систем Linux. Вам потрібен лише монітор HDMI, USB-клавіатура та миша, а також джерело живлення 5 В, і ви готові до роботи.

Плата доступна в декількох версіях і продається як непайкова дошка вартістю всього 35 доларів. Її можна придбати в Інтернеті в електронних магазинах, таких як SparkFun і Adafruit. Більшість постачальників мають велику кількість аксесуарів, які були протестовані та підтверджені щодо сумісності з Raspberry Pi. Серед них невеликі монітори, мініатюрні клавіатури та навіть корпуси для установки плати.

Raspberry Pi була призначена як платформа для дослідження тем з інформатики. Розробники побачили потребу у наданні недорогих, доступних комп’ютерів, які можна було б програмувати для взаємодії з апаратним забезпеченням, таким як сервоприводи, відображувальні пристрої та датчики. Вони також хотіли порушити шаблон того, що потрібно витрачати сотні доларів на

особистий комп’ютер, і таким чином зробити комп’ютери доступними набагато ширшій аудиторії.

Розробники виявили зниження досвіду студентів, які вступають на курси інформатики. Замість того, щоб мати деякий досвід у програмуванні або апаратному забезпеченні, студенти вступають на академічний рік, не маючи майже жодного досвіду роботи з комп’ютерними системами, апаратним забезпеченням або програмуванням. Замість цього студенти добре ознайомлені з Інтернет-технологіями та додатками. Одним із чинників, які сприяли цьому, є вища вартість та більша складність особистого комп’ютера, що означає, що батьки не бажають дозволяти своїм дітям експериментувати на сімейному ПК.

Це становить виклик для академічних установ, які повинні адаптувати свої курси, щоб зробити інформатику зрозумілою для студентів. їм довелося відмовитися від тем нижчого рівня апаратного та програмного забезпечення через відсутність зацікавленості або здатності студентів. Студенти більше не бажають вивчати основи інформатики, такі як мова асемблера, операційні системи, теорія обчислень та конкурентне програмування. Замість цього вони хочуть вчитися мовам програмування вищого рівня для розробки додатків та веб-сервісів. Таким чином, деякі академічні установи більше не пропонують курси з фундаментальної інформатики [37]. Це може призвести до втрати знань і навичок у майбутніх поколінь інформаційних фахівців.

Щоб протистояти цьому тренду, розробники Raspberry Pi вважали, що, обладнавши правильною платформою, молодь може повернутися до експериментів з особистими комп’ютерами, як у дні, коли ПК вимагали набагато більшої залученості до вивчення системи та програмування для задоволення ваші потреби. Наприклад, почтений Commodore 64, Amiga та ранні комп’ютери Apple та IBM PC мали дуже обмежені пропозиції програмного забезпечення.

Назва була частково похідною від внесків комітету з дизайну і частково вибрана для продовження традиції назв нових обчислювальних платформ на честь

фруктів. Частина Pi походить від Python, оскільки дизайнери планували, що Python стане мовою програмування за замовчуванням для цього комп’ютера. Однак доступні і інші варіанти мов програмування.

Рисунок 1.3 – Плата Raspberry Pi Model A

Рисунок 1.4 – Плата Raspberry Pi Model B

Наразі існують дві моделі плат Raspberry Pi: модель A і модель B. Плата моделі A була першою масово виробленою платою і має 256 МБ оперативної пам’яті, один USB-порт та відсутність Ethernet-порту. Це було дуже близько від

послідовної плати моделі B, яка має 512 МБ оперативної пам’яті, два USB-порти та Ethernet-порт. Плати Raspberry Pi A та B зображені раніше на рисунках 1.3 і 1.4.

Таблиця 1.1 Відмінності між Arduino та Raspberry

№	Arduino	Raspberry
1	Плата мікроконтролера	Плата мікрокомп’ютера одноплатного типу
2	Частота процесора:16 МГц	Частота процесора: 1,8 МГц
3	Виконує одну задачу	Виконує кілька завдань
4	Містить: CPU, RAM, ROM	Містить: CPU, RAM, сховище, роз’єми, GPIO-піни, драйвер графіки
5	Один USB	Два USB
6	Низька вартість	Висока вартість
7	Мови програмування: C або C++	Мови програмування: Python, Scratch, Ruby, C, C++
8	Потрібний вихідний код	Потрібна операційна система
9	Відмова від живлення не впливає на пошкодження апаратного забезпечення	Відмова від електроживлення може пошкодити апаратне забезпечення, програмне забезпечення або програми
10	За допомогою щитів можна додавати приводи двигуна, пристрої Wi-Fi, читачі карток, камери, сенсорні екрани і т. д.	GPS та сенсорні екрани можна додавати за допомогою додаткових плат Raspberry або пристроїв HAT
11	Відкритий для більшого обсягу атак електронного взлому через мережу	Не піддається атакам взлому, оскільки підключений до сенсора через USB

Відмінності між Arduino та Raspberry наведено у таблиці 1.1. З неї видно, що Raspberry Pi є потужнішим і в його спільноті тема «Розумного будинку» є більш

поширеною. Саме тому ми вибрали його для розробки нашого середовища «Розумного будинку», тому що в контексті даної роботи він підходить краще [34].

1.3 Мікроконтролери на базі Arduino

Одним із найбільших проривів у фізичному обчисленні стала поширення мікроконтролерів, вони складаються з процесора з невеликим набором інструкцій, пам’яті та програмованого введення/виведення, що міститься на даному чіпі. Мікроконтролери зазвичай розміщені з супутніми схемами та з’єднаннями на маленькій друкованій платі.

Мікроконтролери використовуються у вбудованих системах, де можна створювати невеликі програми для керування та моніторингу апаратного забезпечення, що робить їх ідеальними для використання в сенсорних мережах. Одним із найуспішніших та найпопулярніших мікроконтролерів є платформа Arduino.

Arduino – це апаратна платформа для прототипування з відкритим вихідним кодом, підтримувана середовищем відкритого вихідного коду. Вона була вперше представлена в 2005 році та була розроблена з метою зробити апаратне та програмне забезпечення легкодоступним. Таким чином, вам не потрібно бути експертом в електроніці, щоб використовувати Arduino [32].

Оригінальна цільова аудиторія включала художників та хобістів, які потребували мікроконтролера для їх проектів та творінь щоб зробити їх більш цікавими. Однак, завдяки його легкій використання та універсальності, Arduino швидко стала вибором для більш широкої аудиторії та більшого різноманіття проектів.

Це означає, що ви можете використовувати Arduino для всіляких проектів – від реагування на навколишні умови до керування складними роботизованими

функціями. Arduino також спрощує вивчення електроніки за допомогою практичних застосувань.

Ще одним аспектом, який сприяв швидкому прийняттю платформи Arduino, є зростаюча спільнота внесення внесків до великої кількості інформації, доступної через офіційний веб-сайт Arduino (http://arduino.cc/en/). Під час відвідування веб-сайту ви знаходите відмінний посібник "початок роботи", а також список корисних ідей для проектів та повний довідковий посібник до мови подібної до C для написання коду для керування Arduino.

Arduino також надає інтегроване середовище розробки під назвою Arduino IDE. Воно працює на вашому комп'ютері, де ви можете писати та компілювати скетчі, а потім завантажувати їх на Arduino через з'єднання USB. IDE доступне для Linux, Mac та Windows. Воно розроблено навколо текстового редактора, спеціально призначеного для написання коду, та набору обмежених функцій, призначених для підтримки компіляції та завантаження скетчів.

Скетчі пишуться в спеціальному форматі, що складається з двох обов'язкових методів - одного, що виконується при скиданні або увімкненні живлення Arduino, та іншого, що виконується безперервно. Таким чином, ваш код ініціалізації розміщується в setup(), а код для керування Arduino - в loop(). Мова схожа на C, і ви можете визначати власні змінні та функції [15].

Ви можете розширити функціональність скетчів та забезпечити повторне використання, написавши бібліотеки, що упаковують певні функції, такі як мережі, використання карт пам'яті, підключення до баз даних, проведення математичних обчислень та інше. Багато таких бібліотек включені в IDE. Є також деякі бібліотеки, написані спільнотою Arduino.cc через угоди з відкритим вихідним кодом - деякі з них включені до IDE.

Arduino підтримує ряд аналогових та цифрових пінів, які ви можете використовувати для підключення до різних пристроїв та компонентів для взаємодії з ними. Основні плати мають конкретні розміщення пінів або заголовки, що

дозволяють використовувати розширювальні плати, які називаються щитами. Щити дозволяють додавати додаткові можливості апаратного забезпечення, такі як Ethernet, Bluetooth та підтримка XBee для вашого Arduino [26]. Фізична організація Arduino та щита дозволяє вам ставити щити один на одного. Таким чином, ви можете мати щит Ethernet, а також щит XBee, оскільки кожен використовує різні вводи/виводи.

Далі розглянемо різні моделі Arduino та коротко опишемо їх можливості. Я перелічую плати за часом їхнього випуску, починаючи з найновіших моделей. Важливо розуміти, що є багато інших плат та їх варіантів, які не будуть перелічені. В таблиці 1.2 наведені сфери, в яких можливе застосування даного контролера.

Таблиця 1.2 – Сфери застосування Arduino

№	**Сфера**	**Застосування**
1	Розумний будинок	кондиціонери повітря, освітлення, холодильник, телевізор, камери відеоспостереження, тощо
2	Транспорт	керування літаками, керування потягами, керування автомобілями, тощо
3	Дороги та мости	керування переходами доріг, світлофорами, вуличним освітленням, моніторинг доріг, залізничні шляхи, тощо
4	Аспекти здоров’я	моніторинг життєвоважливих показників для пацієнтів з раком, діабетом, небезпечними заразними хворобами, ЕКГ / ЕЕГ / ЕМГ, тощо.

Певні моделі налаштовані для спеціальних застосувань, у той час як інші розроблені з різними процесорами та конфігураціями пам’яті. Деякі плати вважаються офіційними платами Arduino, оскільки вони мають бренд та підтримку від Arduino.cc. Оскільки Arduino є відкритим вихідним кодом і, зокрема,

ліцензується за допомогою ліцензії Creative Commons Attribution Share-Alike, кожен може будувати сумісні з Arduino плати (часто називають Arduino клонами). Проте вам слід дотримуватися правил і рекомендацій, встановлених Arduino.cc.

Базова організація плати Arduino включає з'єднання USB, роз'єм живлення, перемикач скидання, світлодіоди для живлення та серійного зв'язку та стандартний розміщений набір заголовків для підключення щитів. Офіційні плати мають характерну синю плату з білим написом. З винятком однієї моделі, всі офіційні плати можна установити в шасі (вони мають отвори в платі для кріплення болтів). Виняток - Arduino, призначений для установки на контактну дошку [20].

Плата Leonardo, рисунок 1.1, представляє собою відважний крок вперед для платформи Arduino. Хоча вона підтримує стандартний розмір заголовка, забезпечуючи продовження використання щитів, вона також включає контролер USB, що дозволяє платі з'являтися як пристрій USB на комп'ютері-хості. Плата використовує новіший процесор ATmega32u4 з 20 цифровими пінами введення/виведення, з яких 12 можуть використовуватися як аналогові піни, а 7 - як вихід широтної модуляції (PWM). У неї є 32 КБ флеш-пам'яті і 2,5 КБ SRAM. У Leonardo більше цифрових пінів, ніж у його попередника, але вона продовжує підтримувати більшість щитів.

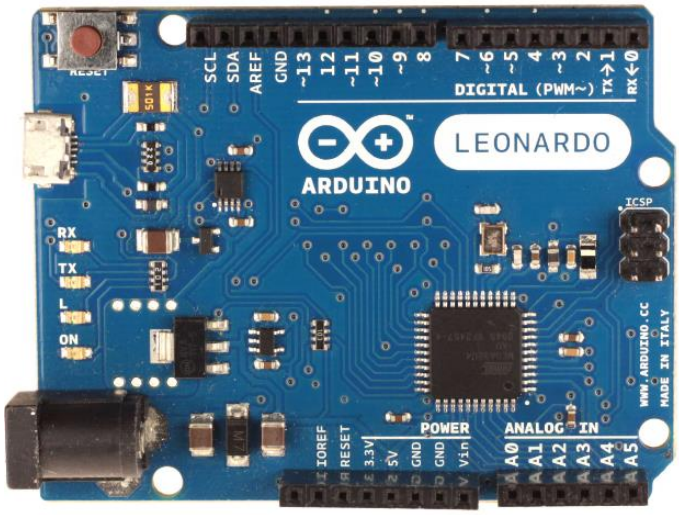


Рисунок 1.5 – Плата Arduino Leonardo

Плата Uno - це перша стандартна плата Arduino з процесором ATmega328; 14 цифровими пінами введення/виведення, з яких 6 можуть використовуватися як вихід широтної модуляції; та 6 аналогових вхідних пінів. На платі Uno є 32 КБ флеш-пам’яті та 2 КБ SRAM.

Плата Uno доступна як пристрій у поверхневому монтажі (SMD), так і у стандартному роз’ємі IC. Версія з роз’ємом IC дозволяє вам замінювати процесори, якщо ви бажаєте використовувати зовнішній пристрій програміста IC для створення власних рішень [5].

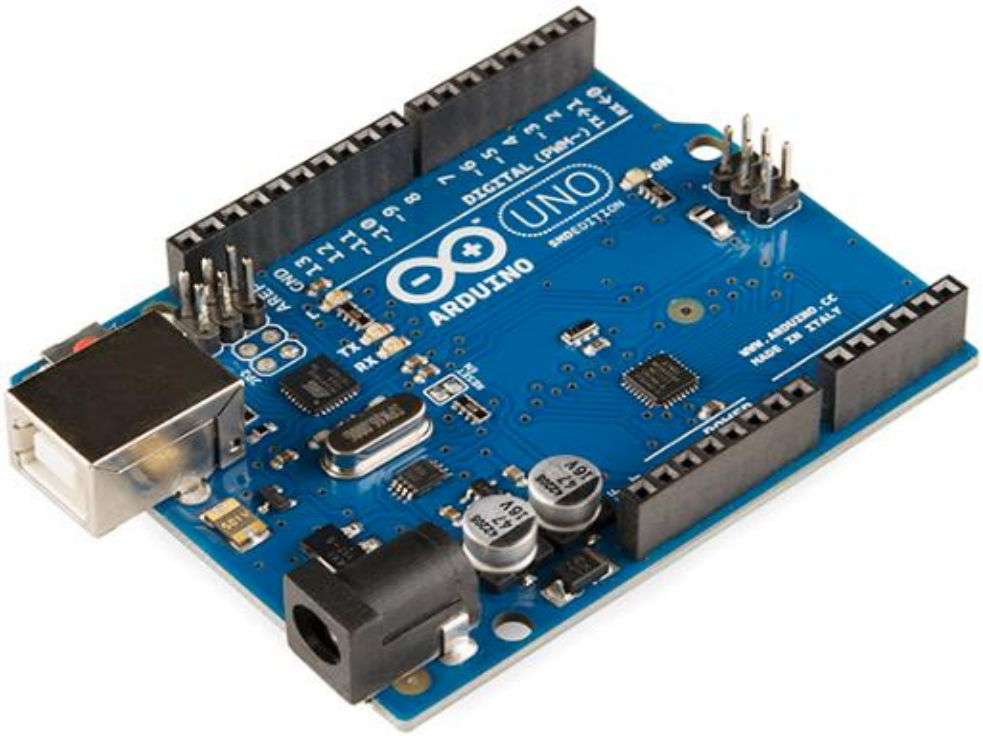


Рисунок 1.6 - Плата Arduino Uno

Arduino Due - це нова, більша та швидша плата на основі процесора Atmel SAM3X8E ARM Cortex-M3, це 32-бітний процесор, плата підтримує велику кількість 54 цифрових портів введення/виведення, з яких 14 можуть використовуватися для виводу ШІМ; 12 аналогових входів; та 4 чіпи UART

(серійних портів); а також 2 цифро-аналогових (ЦАП) та 2 двохпровідні інтерфейси (TWI). Новий процесор пропонує кілька переваг:

1) 32-бітні регістри
2) Контролер DMA (дозволяє виконувати завдання з пам’яттю незалежно від ЦП)
3) 512 КБ флеш-пам’яті
4) 96 КБ SRAM
5) 84 МГц годинник

Due має більший форм-фактор (називається мега-форм-фактором), але все ж підтримує використання стандартних щитів, а також щитів формату мега. Нова плата має одне обмеження: на відміну від інших плат, які можуть приймати до 5 В на пінах введення/виведення, Due обмежується 3,3 В на пінах введення/виведення.

Arduino Due призначений для використання в проектах, які вимагають більше обчислювальної потужності, більше пам’яті та більше пінів введення/виведення. Незважаючи на значні можливості нової плати, вона залишається відкритою для розробки та порівняною за ціною зі своїми попередниками.

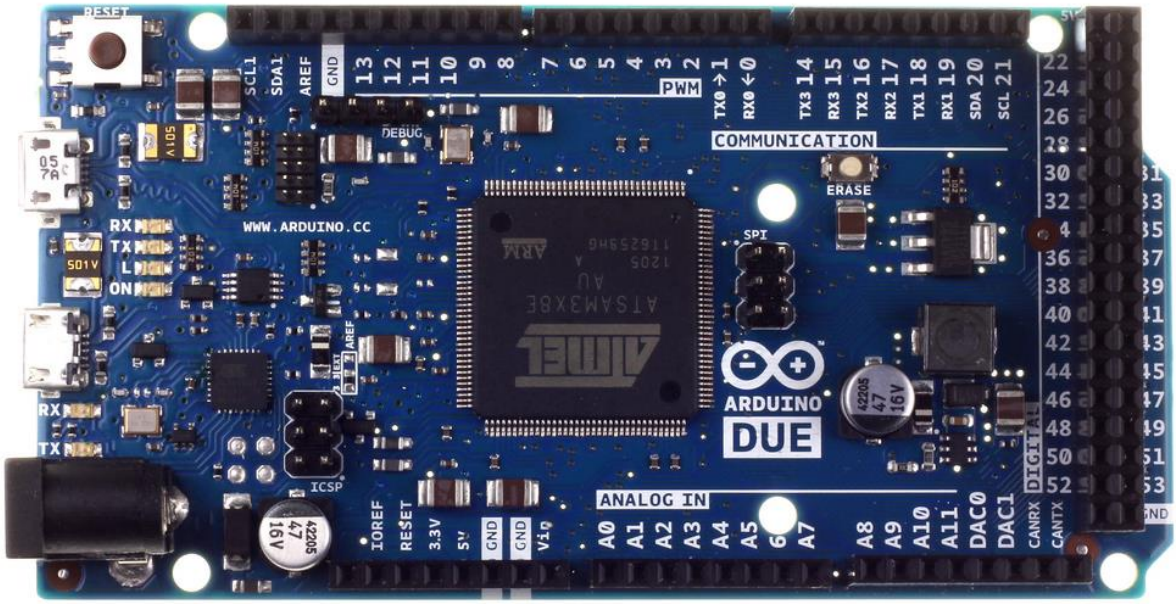


Рисунок 1.7 – Плата Arduino Due

Arduino Pro виготовляється компанією SparkFun (www.sparkfun.com/). Він базується на процесорі ATmega328 (старші моделі використовують ATmega168) та має 14 цифрових пінів введення/виведення, з яких 6 можуть використовуватися для виводу ШІМ, і 8 аналогових входів. У Pro є 32 КБ флеш-пам’яті і 2 КБ SRAM, і він використовує годинник з частотою 16 МГц.

Arduino Pro має форм-фактор, подібний до Uno, але не має заголовків. Проте він може підтримувати стандартні щити, якщо до нього додані заголовки. Це робить Arduino Pro ідеальним для використання в напівпостійних установках, де піни можуть бути припаяні до компонентів або схеми.

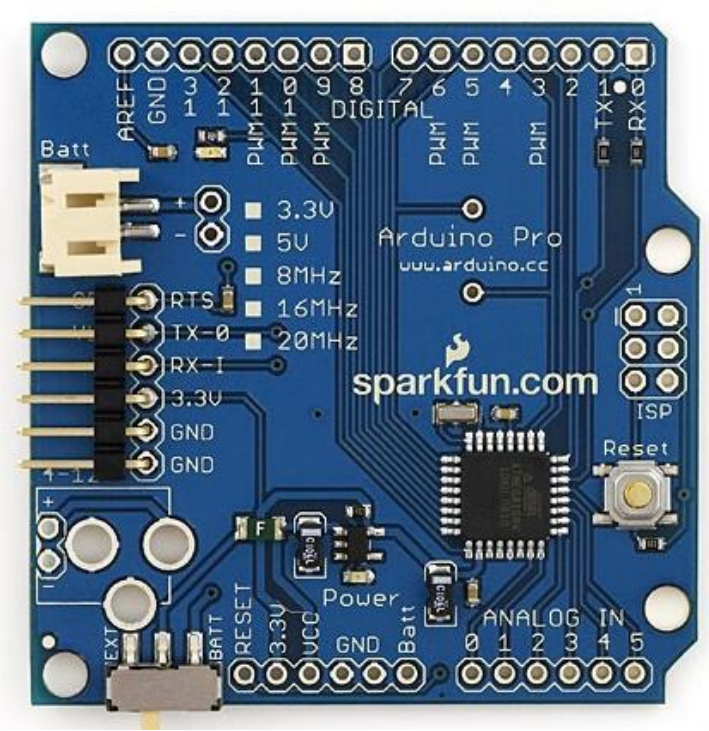


Рисунок 1.8 – Плата Arduino Pro

Крім того, Pro не має USB-роз’єму, тому повинен бути підключений та програмований за допомогою кабелю FTDI або подібної розбірної плати. Він постачається як модель з напругою 3,3 В і годинником 8 МГц або модель з

напругою 5 В і годинником 16 МГц. Так як він призначений для постійного використання, він також має роз’єм для живлення від батареї.

1.4 Висновки до першого розділу

У розділі проведено огляд та аналіз систем «Розумного будинку». Вони почали набирати темпи розвитку нещодавно, але основні положення були сформульовані досить давно, оскільки за відсутністю необхідного програмного та апаратного забезпечення неможливо створити системи подібного рівня.

«Розумний будинок» складається з таких частин: пристрої, датчики, мікроконтролери, сервер, канали передачі даних, хмара, мобільні пристрої, за допомогою яких користувач через сервер керує системою «Розумного будинку».

Було проаналізовано наступні групи контролю:

- Освітлення – відповідає за контроль над освітленістю дому, взаємодіє з групою знаходження.
- Енергозбереження – оптимізує роботу пристроїв.
- Клімат-контроль – регулює системи встановлення температури та вологості в залежності з потребами користувача.
- Безпеки – підсистема захисту від фізичного несанкціонованого вторгнення в дім та аварійних ситуацій.
- Протікання води – система визначення та усунення проблем з протіканням води та, відповідно, оповіщуванням користувача.
- Штучного інтелекту – система, що на основі статистичних даних сама від імені користувача підлагоджує роботи всіх інших систем.

2 РОЗРОБКА САЙТУ SMART HOME DASHBOARD

2.1 Опис функціоналу файлів сайту

У файлі index.html реалізованого вебсайту «Розумного будинку» є такі основні компоненти:

1. Заголовок (<head>):

- Мета теги: Задають кодування символів як UTF-8 та вигляд сторінки для різних пристроїв.
- Назва сторінки: Заголовок вкладки веб-браузера встановлено як «Smart Home Dashboard».
- Підключення стилів:
- Bootstrap для структурування та стилізації веб-сторінки.
- Іконки Font Awesome для візуального поліпшення інтерфейсу.
- Власні стилі з файлу «styles.css».
- Внутрішні стилі: Клас .move-up використовується для переміщення елементів на сторінці.
- Підключення скрипту: Chart.js для створення графіків та діаграм.

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Smart Home Dashboard</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css" rel="stylesheet">
    <link rel="stylesheet" href="styles.css">
    <style>
        .move-up {
            position: relative;
            top: -400px;
        }
    </style>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
```

Рисунок 2.1 – Лістинг заголовку <head>

2. Тіло сторінки (<body>):

- Контейнер Bootstrap: Використовується для центрування та організації вмісту сторінки.
- Великий заголовок, що вказує на функцію сторінки — панель управління «Розумним будинком».
- Розмітка сторінки:
- Колонка вибору кімнат: Список для навігації між різними кімнатами (вітальня, кухня, спальня тощо). Взаємодія з кожним пунктом списку викликає функцію changeRoom, яка змінює вміст інформації про кімнату.
- Колонка інформації та графіків: Показує динамічно змінюваний контент, такий як інформація про вибрану кімнату і графік температур.
- Зображення кімнати: Приклад зображення кімнати, який можна змінити у залежності від вибору кімнати. Зображення розміщено в колонці, яка використовує клас move-up для зміщення вверх.

```
<body>
    <div class="container">
        <div class="header">
            <h1>Smart Home Dashboard</h1>
        </div>
        <div class="row">
            <div class="col-md-4">
                <div class="list-group">
                    <a href="#" class="list-group-item list-group-item-action" onclick="changeRoom('living-room')">
                        Living Room
                    </a>
                    <a href="#" class="list-group-item list-group-item-action" onclick="changeRoom('kitchen')">
                        Kitchen
                    </a>
                    <a href="#" class="list-group-item list-group-item-action" onclick="changeRoom('bedroom')">
                        Bedroom
                    </a>
                    <a href="#" class="list-group-item list-group-item-action" onclick="changeRoom('bedroom-2')">
                        Bedroom 2
                    </a>
                </div>
            </div>
            <div class="col-md-8">
                <div id="room-info" class="room-info"></div>
                <canvas id="tempChart" class="temperature-chart"></canvas>
            </div>
            <div class="col-md-8 move-up">
                <img src="/prog/images/default-room.png" alt="Room Image" class="img-fluid room-image">
            </div>

        </div>
    </div>

    <script src="script.js"></script>
</body>
```

Рисунок 2.2 – Лістинг тіла сторінки <body>

3. Скрипт:

• Файл script.js містить JavaScript код, що керує інтерактивними функціями сторінки, зокрема зміною вмісту інформації про кімнати та відображенням даних на графіку.

Цей HTML файл створює інтерактивну панель управління, де користувач може вибирати різні кімнати будинку, переглядати специфічну інформацію про них, та візуалізувати зміни температури за допомогою графіку. Повний лістинг файду index.html знаходиться в додатку А.

Файл styles.css визначає візуальний стиль сайту «Розумного будинку». Ось детальний опис кожного сегмента стилю:

1) Загальні Стилі:

• Шрифти: Імпортовано шрифт Roboto з Google Fonts, який використовується у всьому документі для читабельності.

• Основні стилі для body та html: Встановлено нульові відступи, фоновий колір #f4f4f4 (світлий сірий), колір тексту #333 (темно-сірий), і задано повну висоту з відключенням горизонтального скролу.

```
body, html {
    margin: 0;
    padding: 0;
    font-family: 'Roboto', sans-serif;
    background: #f4f4f4; /* Новий світлий фон */
    color: #333; /* Колір тексту за замовчуванням */
    height: 100%;
    overflow-x: hidden;
}
```

Рисунок 2.3 – Лістинг загальних стилів

2) Стилі Контейнера:

- container: Максимальна ширина контейнера 1200px, автоматичні відступи для центрування, білий фон, закруглені кути, та легка тінь, що надає візуальної глибини.

```
.container {
    max-width: 1200px;
    margin: 0 auto;
    padding: 20px;
    background: #fff; /* Світлий фон контейнера */
    border-radius: 10px;
    box-shadow: 0 4px 8px 0 rgba(0,0,0,0.1); /* Легкий тінь */
}
```

Рисунок 2.4 – Лістинг стилів контейнера

3) Заголовок:

- header: Вирівнювання тексту по центру, збільшений розмір шрифту та змінений колір до світлішого відтінку сірого для кращої видимості.

```
.header {
    text-align: center;
    padding: 20px 0;
    font-size: 2.5rem;
    font-weight: 500;
    color: #555; /* колір заголовка */
}
```

Рисунок 2.5 – Лістинг стилю заголовка

4) Стилі Списку:

- .list-group-item: Світлий фон для елементів списку, легкий сірий контур, та анімація зміни фону при наведенні курсора.

```css
.list-group-item {
    background-color: #f4f4f4; /* Світлий фон для пунктів списку */
    color: #333; /* Колір тексту пунктів списку */
    border: 1px solid #ddd; /* Легкий контур */
    transition: background-color 0.3s ease;
}
```

Рисунок 2.6 – Лістинг стилів списку

5) Інформація про Кімнату та Графіки:

- room-info та .temperature-chart: Обидва мають світлий фон, м’яке тіньове оформлення, і виділені зліва смугами, що додає акцент. Для room-info смуга має кольоровий акцент.

```css
.room-info {
    background: #f9f9f9; /* Світлий фон для інформації про кімнату */
    padding: 15px;
    margin-bottom: 20px;
    border-radius: 5px;
    border-left: 5px solid #6a11cb;
    box-shadow: 0 2px 4px 0 rgba(0,0,0,0.1);
}
```

Рисунок 2.7 – Лістинг room-info

```css
.temperature-chart {
    background: #f9f9f9; /* Світлий фон для графіка температур */
    padding: 10px;
    border-radius: 5px;
    box-shadow: 0 2px 4px 0 rgba(0,0,0,0.1);
}
```

Рисунок 2.8 – Лістинг temperature-chart

6) Стилі Зображення:

- room-image: Задані фіксовані розміри для зображень, що гарантує однакову презентацію зображень на сторінці.

```
.room-image {
    width: 400px;
    height: 400px;
}
```

Рисунок 2.9 – Лістинг room-image

Ці стилі спрямовані на створення чистого, професійного та візуально приємного інтерфейсу, який полегшує навігацію та взаємодію користувача з веб-сайтом. Повний лістинг файду styles.css знаходиться в додатку А.

Файл script.js вебсайту «Розумного будинку» містить кілька ключових JavaScript функцій, які забезпечують інтерактивність і функціональність інтерфейсу. Ось детальний опис цих функцій:

1. Функція updateChart:

Ця функція призначена для оновлення графіка температур. Вона приймає два параметри:

- room: назва кімнати.
- temperatures: масив значень температур.

Функція виконує наступні дії:

- Виводить в консоль інформацію про оновлення графіка для вказаної кімнати.
- Отримує контекст 2D для елементу canvas (де буде малюватися графік).
- Якщо графік вже існує (window.myChart), він знищується, щоб створити новий.

- Створює новий графік лінійного типу з даними про температуру протягом тижня.
- Налаштування графіка включають параметри візуалізації та відповіді на зміни розмірів.

```
function updateChart(room, temperatures) {
    console.log('Updating chart for room:', room); // Це допоможе зрозуміти, чи функція взагалі викликається
    const ctx = document.getElementById('tempChart').getContext('2d');
    if (window.myChart) {
        console.log('Destroying old chart');
        window.myChart.destroy();
    }
    window.myChart = new Chart(ctx, {
        type: 'line',
        data: {
            labels: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'],
            datasets: [{
                label: 'Temperature over the week (°C)',
                data: temperatures,
                backgroundColor: 'rgba(75, 192, 192, 0.2)',
                borderColor: 'rgba(75, 192, 192, 1)',
                borderWidth: 3,
                pointBackgroundColor: 'white',
                pointBorderColor: 'rgba(75, 192, 192, 1)',
                pointBorderWidth: 2,
                pointRadius: 5,
                tension: 0.4
            }]
        },
        options: {
            responsive: true,
            scales: {
                y: {
                    beginAtZero: false
                }
            }
        }
    });
    console.log('Chart updated');
}
```

Рисунок 2.10 – Лістинг функції updateChart

2. Об’єкт roomImages:

Цей об’єкт мапингу використовується для зберігання шляхів до зображень різних кімнат. Ключі відповідають назвам кімнат, а значення — шляхи до зображень.

```
const roomImages = {
    'living-room': '../../images/living_room.png',
    'kitchen': '../../images/kitchen.png',
    'bedroom': '../../images/bedroom.png',
    'bedroom-2': '../../images/bedroom2.png',
};
```

Рисунок 2.11 – Лістинг об’єкту roomImages

3. Функція changeRoom:

Функція для зміни відображуваної інформації про кімнату:

- Виконує запит на сервер для отримання даних про кімнату.
- Оновлює HTML вміст блоку з інформацією про кімнату, включаючи температуру, рівень CO2, вологість, рух та енергоефективність.
- Оновлює графік за допомогою відповідних даних.
- Оновлює зображення кімнати залежно від вибору.

```
function changeRoom(room) {
    fetch(`http://localhost:3000/room-data/${room}`)
        .then(response => {
            if (!response.ok) {
                throw new Error('Network response was not ok');
            }
            return response.json();
        })
        .then(info => {
            const roomInfo = document.getElementById('room-info');
            roomInfo.innerHTML = `<h4>${room.replace('-', ' ').toUpperCase()}</h4>
                                  <p><i class="fas fa-thermometer-half"></i> Current Temperature: ${info.currentTemperature}°C</p>
                                  <p><i class="fas fa-wind"></i> CO2 Level: ${info.co2} ppm</p>
                                  <p><i class="fas fa-tint"></i> Humidity: ${info.humidity}%</p>
                                  <p><i class="fas fa-walking"></i> Motion: ${info.motion}</p>
                                  <p><i class="fas fa-bolt"></i> Energy Efficiency: ${info.energyEfficiency}%</p>`;
            updateChart(room, info.temperatures);
            changeRoomImage(room);
        })
        .catch(error => {
            console.error('Error fetching data:', error);
            alert('Failed to load room data. Please check your network connection and try again.');
        });
}
```

Рисунок 2.12 – Лістинг функції changeRoom

4. Функція changeRoomImage:

Функція для зміни зображення кімнати. Вона встановлює атрибут src для елементу зображення, використовуючи відповідний шлях з об’єкта roomImages.

```
function changeRoomImage(room) {
    const roomImage = roomImages[room] || 'default-room.png';
    const roomImg = document.querySelector('.room-image');
    roomImg.src = roomImage;
}
```

Рисунок 2.13 – Лістинг функції changeRoomImage

5. Початкова ініціалізація:

Коли документ завантажено (DOMContentLoaded), автоматично відображається інформація про вітальню (living-room) як стартова кімната.

```
document.addEventListener('DOMContentLoaded', () => {
    changeRoom('living-room'); // Default room to display
});
```

Рисунок 2.14 – Лістинг початкової ініціалізації

6. Веб-сервер на Node.js:

Код також містить елементи налаштування простого веб-сервера на Node.js за допомогою фреймворку Express, що обробляє статичні файли, такі як зображення.

```
const express = require('express');
const app = express();
app.use(express.static('images')); //зберігаються статичні файли
```

Рисунок 2.15 – Лістинг елементів налаштування веб-сервера на Node.js

Цей скрипт є важливою частиною функціональності сайту, оскільки він відповідає за забезпечення динамічної взаємодії користувача з різними кімнатами в «Розумному будинку», показує актуальну інформацію та управляє відображенням графіків. Повний лістинг файду script.js знаходиться в дотатку А.

Файл server.js представляє собою серверний код, написаний за допомогою Node.js і фреймворку Express. Він(код) відповідає за створення API, яке вебсайт використовує для отримання даних про різні кімнати у «Розумному будинку». Ось детальний опис основних компонентів цього файлу:

1) Залежності:

- Express: Веб-фреймворк для Node.js, що полегшує створення веб-серверів.
- CORS: Middleware для обробки CORS (Cross-Origin Resource Sharing) запитів, дозволяє вашому веб-клієнту безперешкодно звертатися до сервера.
- FS (File System): Модуль для роботи з файловою системою, використовується для читання даних з файлу.

```
const express = require('express');
const cors = require('cors');
const fs = require('fs');
```

Рисунок 2.16 – Лістинг залежностей express, cors, fs

2) Налаштування Express App:

- app.use(cors()): Використання CORS middleware для обробки міждоменних запитів.
- app.use(express.json()): Middleware для розбору JSON-форматованих запитів.

```
app.use(cors());
app.use(express.json());
```

Рисунок 2.17 – Лістинг налаштування Express App

3) Функція readData:

- Мета: Читання та парсинг JSON-даних з файлу data.json.
- Процес: Використовує метод fs.readFileSync для синхронного читання файла, потім перетворює прочитані дані з JSON-формату в JavaScript об’єкт за допомогою JSON.parse.

```
// Функція для читання даних з файлу
function readData() {
    const jsonData = fs.readFileSync('data.json');
    return JSON.parse(jsonData);
}
```

Рисунок 2.18 – Лістинг функції readData

4) API Endpoint:

- Маршрут: GET /room-data/:room
- Параметри: room — параметр URL, що визначає кімнату, дані про яку потрібно отримати.
- Логіка: Функція обробки запиту спершу отримує назву кімнати з параметрів запиту, потім викликає readData для отримання всіх даних, і, нарешті, відправляє відповідь з даними про вказану кімнату або статус 404, якщо кімнату не знайдено.

```
// API для отримання даних про кімнату
app.get('/room-data/:room', (req, res) => {
    const room = req.params.room;
    const data = readData();
    if (data[room]) {
        res.json(data[room]);
    } else {
        res.status(404).send('Room not found');
    }
});
```

Рисунок 2.19 – Лістинг API для отримання даних про кімнату

5) Слухання порту:

- Команда: app.listen(port, callback)
- Порт: 3000
- Функція зворотнього виклику: Виводить повідомлення в консоль про те, що сервер працює на порту 3000.

```
app.listen(port, () => {
    console.log(`Server running on port ${port}`);
});
```

Рисунок 2.20 – Лістинг слухання порту

Цей серверний код становить основу для бекенд-сервісу, що дозволяє додатку динамічно отримувати та відображати дані про різні кімнати в «Розумному будинку», взаємодіючи з JSON файлом, як з базою даних. Повний лістинг файду server.js знаходиться в дотатку А.

Файл data.json містить структуровані дані у форматі JSON, які використовуються сервером для відповіді на запити з вебсайту «Розумного будинку». Він включає інформацію для різних кімнат.

Об’єкти кімнат:

Кожна кімната представлена як властивість об’єкту на найвищому рівні, де ключ — назва кімнати, а значення — об’єкт із деталями про кімнату.

Спільні властивості для кожної кімнати:

- temperatures: Масив чисел, що представляє температури за тиждень (наприклад, від понеділка до неділі). Вказує динаміку змін температури в кімнаті.
- currentTemperature: Поточна температура в кімнаті.
- co2: Рівень вуглекислого газу в кімнаті, виміряний у частках на мільйон (ppm).
- humidity: Відсоток вологості в кімнаті.
- motion: Статус виявлення руху в кімнаті, що може бути «Detected», «Not Detected» або інше специфічне значення.
- energyEfficiency: Рівень енергоефективності в кімнаті, виражений у відсотках.

Файл data.json є ключовим для забезпечення динамічної інформації на сайті, дозволяючи оновлювати дані про кімнати у реальному часі в залежності від запитів, які робляться через веб-інтерфейс. Повний лістинг файду data.json знаходиться в дотатку А.

```
"living-room": {
    "temperatures": [200, 500, 10, 18, 18, 18, 18],
    "currentTemperature": 24,
    "co2": 220,
    "humidity": 46,
    "motion": "test",
    "energyEfficiency": 80
},
"kitchen": {
    "temperatures": [19, 18, 20, 22, 21, 20, 19],
    "currentTemperature": 20,
    "co2": 500,
    "humidity": 50,
    "motion": "Not Detected",
    "energyEfficiency": 82
},
```

Рисунок 2.21 – Лістинг бази даних

Для передачі даних з мікроконтролера Raspberry Pi до файлу data.json на сервері, ми можемо використати наступний план дій:

Крок 1: Збір даних з датчиків

Перше, що нам потрібно зробити, це підключити відповідні датчики до Raspberry Pi. Це можуть бути датчики температури, вологості, рівня CO2, а також датчики руху. Вибір датчиків залежить від параметрів, які потрібно виміряти.

Крок 2: Програмування Raspberry Pi

Ми можемо використати Python, як мову програмування для розробки скрипта, який буде читати дані з датчиків. Для більшості датчиків існують бібліотеки Python, що спрощують процес читання даних. Наприклад:

```
import Adafruit_DHT

sensor = Adafruit_DHT.DHT22
pin = 23  # GPIO pin, до якого підключений датчик

humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
if humidity is not None and temperature is not None:
    print(f"Temp: {temperature} C, Humidity: {humidity} %")
else:
    print("Failed to retrieve data from humidity sensor")
```

Рисунок 2.22 – Приклад лістингу скрипту для читання даних з датчиків

Крок 3: Відправлення даних на сервер

Для передачі даних на сервер необхідно використати HTTP-запити. В Python це можна зробити за допомогою бібліотеки requests. Ми повинні налаштувати ваш сервер так, щоб він приймав дані, наприклад, через POST-запити:

```python
import Adafruit_DHT

sensor = Adafruit_DHT.DHT22
pin = 23  # GPIO pin, до якого підключений датчик

humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
if humidity is not None and temperature is not None:
    print(f"Temp: {temperature} C, Humidity: {humidity} %")
else:
    print("Failed to retrieve data from humidity sensor")
```

Рисунок 2.23 – Приклад лістингу налаштування серверу

Крок 4: Оновлення data.json на сервері

На сервері, використовуючи Express.js, ми можему створити кінцеву точку API, яка оброблятиме POST-запити та оновлюватиме data.json. Для цього нам потрібно розробити логіку зчитування поточного стану файлу, оновлення його даних, та збереження змін:

```javascript
app.post('/update-data', (req, res) => {
    const newData = req.body;
    const data = readData(); // ваша функція для зчитування data.json
    // припустимо, ви хочете оновити дані для 'living-room'
    data['living-room'] = { ...data['living-room'], ...newData };
    fs.writeFileSync('data.json', JSON.stringify(data));
    res.send('Data updated successfully');
});
```

Рисунок 2.24 – Приклад реалізації логіки зчитування поточного стану файлу, оновлення та збереження

Перевірка і тестування

Перед використанням у «бойових» умовах варто ретельно протестувати всю систему, переконавшись, що дані коректно зчитуються з датчиків, передаються на сервер та правильно оновлюються у файлі data.json.

2.2 Висновок до другого розділу

У розділі був проведений огляд та висвітлені ключові аспекти розробленого сайту моніторингу «Розумного будинку», який включає веб-інтерфейс рисунок 2.25.

Також було описано, як можна реалізувати збір та передачу даних з Raspberry Pi на сервер, що дозволяє інтегрувати реальні дані з датчиків у систему.

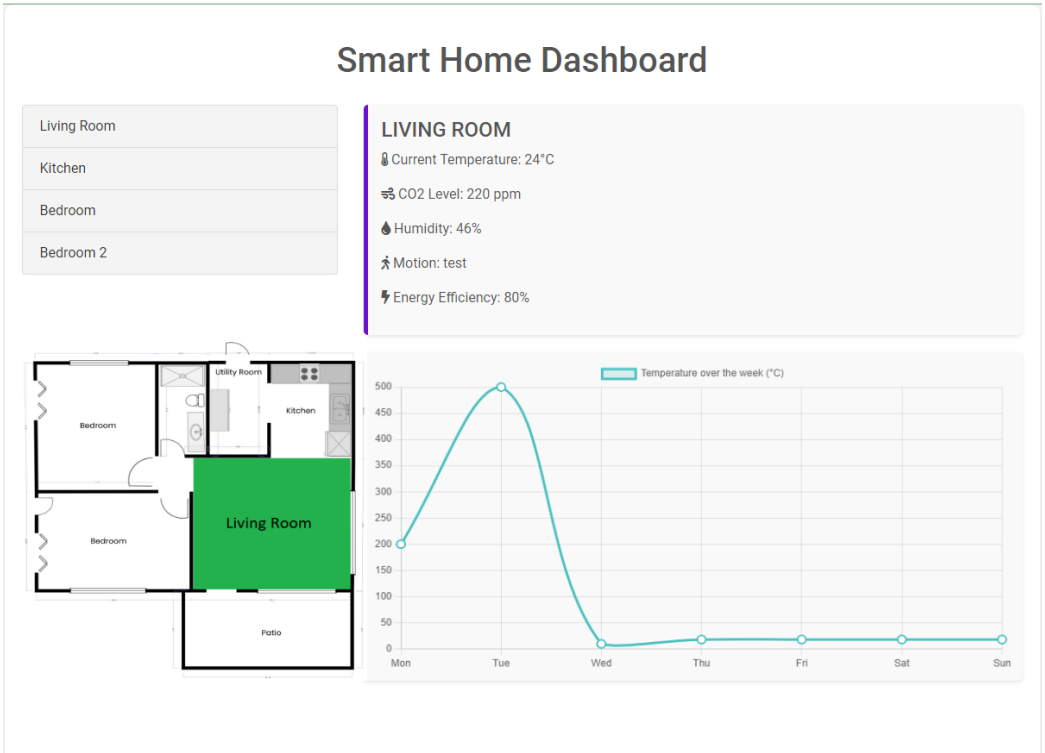


Рисунок 2.25 – Головне вікно сайту «Smart Home Dashboard»

Була наведена серверна логіка та інтеграція з Raspberry Pi, було детально описано наступні частини сайту:

1. index.html — основний веб-файл, що включає HTML-структуру для динамічного відображення інформації про кімнати та температури через графік.

2. styles.css — файл стилів, який задає візуальний вигляд вебсайту, забезпечуючи читабельність та сучасний дизайн.

3. script.js — скрипт, який управляє динамікою сайту, зокрема зміною інформації про кімнати та оновленням графіків температур.

4. server.js — серверний код на Node.js, що обробляє запити до сервера та оновлює файл data.json з актуальними даними.

5. data.json — файл даних, який містить інформацію про різні параметри кімнат «Розумного будинку».

Ця інформація та аналіз допоможуть у розвитку та удосконаленні проекту «Розумного будинку», забезпечуючи його надійність та функціональність.

3 РОЗРОБКА ТА ОПИС РОБОТИ «РОЗУМНОГО БУДИНКУ»

3.1 Особливості Raspberry Pi

«Розумні будинки» використовують різноманітні технології для автоматизації побутових процесів, покращуючи комфорт, безпеку та енергоефективність. Автоматизація включає управління освітленням, температурою, безпекою, медіа-системами, і багато іншого. «Розумні будинки» можуть також забезпечувати особам з особливими потребами більшу самостійність та безпеку [34].

Важливі переваги «Розумних будинків»:

1. Енергоефективність: Автоматичне вимкнення електроприладів, коли вони не використовуються, та оптимізація використання енергії.
2. Безпека: Відеонагляд, смарт-замки та системи охоронної сигналізації, які можуть бути керовані з будь-якої точки світу.
3. Комфорт: Автоматизація повсякденних задач, таких як налаштування освітлення та клімат-контролю, залежно від зовнішніх умов і персональних уподобань.

Raspberry Pi 4 є компактним, доступним і потужним мікрокомп’ютером, який ідеально підходить для DIY (зроби сам) проектів автоматизації дому. Зі значними покращеннями у швидкості процесора, пам’яті та підключеннях, Raspberry Pi 4 дозволяє реалізовувати більш складні та надійні системи «Розумного будинку».

Node-RED є візуальною розробною платформою, яка дозволяє створювати автоматизації за допомогою перетягування блоків (нод), що представляють пристрої, API, бази даних або онлайн-сервіси. Node-RED особливо підходить для інтеграції Інтернету речей, що робить його ідеальним для «Розумних будинків», оскільки він може легко з’єднуватись із різними сенсорами та актуаторами через Raspberry Pi [18]. Переваги поєднання Raspberry Pi та Node-RED є:

1. Низька вартість: Обидва інструменти доступні за ціною, що робить створення «Розумного будинку» доступним.

2. Гнучкість: Велика кількість доступних модулів та бібліотек для Node-RED дозволяє легко налаштувати систему під конкретні потреби.

3. Спільнота: Обидві платформи мають великі активні спільноти, які надають підтримку, бібліотеки та навчальні матеріали.

Raspberry Pi 4 Model B представляє собою значне покращення порівняно з попередніми моделями. Ось ключові характеристики, що роблять його потужним інструментом для «Розумного будинку»:

1. Процесор: Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz

2. ОЗП: Доступний у варіантах 2GB, 4GB, або 8GB LPDDR4-3200 SDRAM.

3. Підключення:

- Gigabit Ethernet
- Dual-band 802.11ac Wi-Fi
- Bluetooth 5.0
- 2 USB 3.0 порти та 2 USB 2.0 порти

4. Відео та звук:

- 2 micro-HDMI порти (підтримка до 4K 60Hz)
- H.265 (4Kp60 decode)

5. Живлення: 5V DC через USB-C (мінімум 3A)

6. Сховище: microSD карта для завантаження ОС та зберігання даних

Перевагами Raspberey PI для домашньої автоматизації є:

1. Потужність та швидкість: Новий процесор забезпечує достатню потужність для обробки даних з множини сенсорів і виконання кількох задач одночасно без затримок.

2. Мультизадачність: Збільшена кількість ОЗП та покращені мережеві можливості дозволяють Raspberry Pi 4 ефективно обробляти великі об’єми даних, що ідеально підходить для автоматизації та моніторингу різноманітних функцій в «Розумному будинку».

3. Підтримка відео високої роздільної здатності: Підтримка 4K відео робить можливим використання Raspberry Pi 4 для систем відеонагляду з високою чіткістю.

4. Надійність: Заснований на Linux, Raspberry Pi 4 пропонує стабільність та надійність, критичні для постійної роботи системи «Розумного будинку».

Хоча Raspberry Pi 4 має багато переваг, важливо враховувати його обмеження при плануванні «Розумного будинку»:

1. Тепловиділення: При інтенсивному використанні Raspberry Pi 4 може значно нагріватись, що може вимагати додаткового охолодження.

2. Живлення: Raspberry Pi 4 вимагає стабільного джерела живлення з достатнім струмом (мінімум 3А), що може вимагати використання якісних блоків живлення.

Для максимальної ефективності та надійності рекомендується використовувати Raspberry Pi 4 з охолоджувальним кулером та якісним джерелом живлення. Також важливо регулярно робити резервні копії системи та даних, оскільки це забезпечить відновлення системи у випадку збоїв [22].

Node-RED — це програмна платформа, розроблена для візуального програмування за допомогою потоків (flows), яка дозволяє інтегрувати різноманітні апаратні та програмні компоненти. Використовуючи концепцію «перетягування» (drag-and-drop), Node-RED робить процес створення програмних рішень доступним навіть для не-програмістів. Невеликий опис наведено нижче:

1. Вузли (Nodes): Основні будівельні блоки в Node-RED, кожен вузол виконує певну функцію, наприклад, зчитування даних з сенсора, виконання логічних операцій, відправка повідомлень тощо.

2. Потоки (Flows): Вузли з’єднуються між собою лініями, формуючи потоки, що представляють собою повні програмні алгоритми.

3. Інтерфейс користувача: Веб-базований редактор, що дозволяє користувачам легко взаємодіяти з компонентами системи, створюючи та модифікуючи потоки в реальному часі.

До основних вузлів відносяться:

1. Inject node: Тригер для ініціації потоку, може бути налаштований для відправки повідомлень в певний час або за певною подією.

2. Debug node: Використовується для відображення даних у вікні відладки, що допомагає в налагодженні потоків.

3. Function node: Дозволяє писати власний код на JavaScript для виконання різноманітних завдань всередині потоку.

4. HTTP nodes: Вузли для створення веб-сервера або відправки/отримання HTTP-запитів, що дозволяє інтегрувати Node-RED з веб-сервісами або IoT пристроями.

Встановлення Node-RED на Raspberry Pi можна виконати за допомогою кількох команд у терміналі:

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

Ця команда автоматично встановить Node-RED разом з необхідними залежностями, включаючи Node.js.

Після встановлення, Node-RED можна запустити через термінал командою: `node-red-start`

Для доступу до редактора Node-RED, необхідно ввести в браузері адресу: http://<IP_Raspberry_Pi>:1880 [19]. На веб-сторінці відобразиться інтерфейс, де зможемо почати створювати свій перший потік, перетягуючи вузли з панелі вузлів на робочу область і з’єднуючи їх між собою.

Реалізація «Розумного будинку» на базі Raspberry Pi 4 вимагає інтеграції різноманітних апаратних компонентів [35]. Ось основні категорії обладнання, які зазвичай використовуються у таких проектах:

1. Сенсори: Збір інформації про стан середовища в будинку.
2. Актуатори: Вплив на фізичний стан середовища через керування пристроями.
3. Комунікаційні модулі: Забезпечення зв’язку між компонентами системи.
4. Живлення: Забезпечення енергією всієї системи.

Опис сенсорів та актуаторів:

1. Температурні та вологісні сенсори:

- DHT22: Популярний сенсор для вимірювання температури та вологості. Він відносно дешевий та простий у використанні. Видає цифровий сигнал, який можна легко зчитати з Raspberry Pi через один GPIO пін.
- DS18B20: Температурний сенсор із можливістю підключення за допомогою шини 1-Wire, що дозволяє підключити багато таких сенсорів до одного GPIO піна.

2. Датчики руху:

- PIR (Passive Infrared Sensor): Використовується для виявлення руху в приміщенні. Це допомагає автоматизувати освітлення та системи безпеки.

3. Камери:

- Raspberry Pi Camera Module: Інтегрується безпосередньо з Raspberry Pi через спеціальний інтерфейс. Використовується для відеонагляду або інших застосунків, що потребують обробки зображень.

4. Актуатори:

- Реле: Використовуються для керування високовольтними навантаженнями, такими як домашнє освітлення або електричні обігрівачі. За

допомогою реле можна безпечно вмикати та вимикати ці пристрої за командами з Raspberry Pi [36].

3.2 Підключення датчиків до Raspberry Pi

Підключення DHT22 до Raspberry Pi:

1. **VCC** під’єднується до 5V на Raspberry Pi.
2. **GND** під’єднується до заземлення.
3. **Data** під’єднується до одного з GPIO пінів.

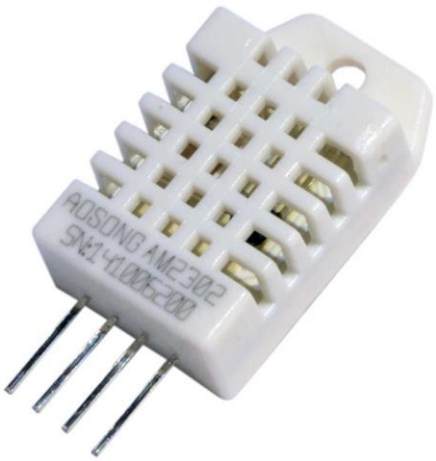
Для читання даних з DHT22 можна використати бібліотеку, таку як Adafruit_DHT:

```
import Adafruit_DHT

sensor = Adafruit_DHT.DHT22
pin = 23

humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
if humidity is not None and temperature is not None:
    print(f"Temp: {temperature:.1f} C  Humidity: {humidity:.1f}%")
else:
    print("Failed to retrieve data from humidity sensor")
```

3.1 – Лістинг підлкючення датчика DHT22



3.2 – Датчик DHT22

Підключення реле:

1. **VCC** під’єднується до 5V на Raspberry Pi.
2. **GND** під’єднується до заземлення.
3. **IN** під’єднується до одного з GPIO пінів для управління станом реле.

```
import RPi.GPIO as GPIO

relay_pin = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(relay_pin, GPIO.OUT)

GPIO.output(relay_pin, GPIO.HIGH) # Turn on
```

3.3 – Лістинг підключення реле

Схему підключення датчиків до мікроконтролера наведено в додатку Б.

Встановлення ОС Raspberry Pi (Raspberry Pi OS):

1. Завантаження образу ОС: Необхідно перейти на офіційний сайт Raspberry Pi та завантажити останню версію Raspberry Pi OS.
2. Запис образу на SD карту: Використовуючи інструмент Etcher для запису образу на SD карту.
3. Настроювання Wi-Fi та включення SSH (опціонально): Необхідно додати файл wpa_supplicant.conf та файл ssh у кореневий каталог SD карти перед першим запуском.
4. Запуск Raspberry Pi: Необхідно вставити SD карту в Raspberry Pi та підключити живлення. Після того потрібно підключитися до Raspberry Pi через SSH або використовати монітор та клавіатуру для налаштування.

Встановлення Node.js**:** Node-RED потребує Node.js, який можна встановити через команду: curl – sL https://deb.nodesource.com/setup_14.x | sudo dash - sudo apt-get install –y nodejs

Встановлення Node-RED**:** sudo npm install –g –unsafe-perm node-red

Запуск Node-RED: node-red-start

Тепер Node-RED буде доступний за адресою http://<IP_Raspberry_Pi>:1880.

Встановлення бібліотек:

1. Необхідно використати менеджер палітр Node-RED для встановлення вузлів, які розширюють функціональність, наприклад node-red-dashboard для створення графічних панелей управління або node-red-contrib-usb для зчитування даних з USB пристроїв [36].

Конфігурація потоків у Node-RED:

1. Створення базового потоку: Потрібно додати inject вузол для ініціації дій, function вузол для обробки даних, та debug вузол для моніторингу виводу.

2. Інтеграція з апаратними компонентами: Наприклад, для читання даних з DHT22, потрібно використати відповідний вузол, налаштувати його для зчитування даних з відповідного GPIO піна.

3. Розробка користувацького інтерфейсу: Потрібно використати dashboard вузли для створення інтерактивних кнопок, слайдерів та індикаторів, які дозволяють користувачам керувати системою в режимі реального часу.

Конфігурація безпеки:

1. Налаштування брандмауера: Потрібно використати ufw або аналогічні інструменти для обмеження доступу до відкритих портів.

2. Захист доступу до Node-RED: Встановити паролі для інтерфейсу Node-RED та веб-доступу до дашборду.

Node-RED використовує концепцію «потоків» (flows) для візуальної організації логіки програми [19]. Кожен потік складається з набору «вузлів» (nodes), з’єднаних між собою, що представляють різні операції, такі як зчитування даних, обробка, та керування пристроями.

Приклади потоків для моніторингу та управління

1. Моніторинг температури та вологості:

- Вузли: Використання сенсорного вузла (наприклад, для DHT22), вузла функцій для обробки даних, та вузла відладки для відображення результатів.
- Логіка: Сенсор зчитує дані кожні 30 секунд, дані обробляються та відображаються у вікні відладки та на дашборді.
- Код Node-RED:

```
[{«id»:»e1»,»type»:»sensor»,»z»:»b787c9.7c541238»,»name»:»DHT22»,»sensor»:»DHT22»,»x»:70,»y»:100,»wires»:[[«e2»]]},
{«id»:»e2»,»type»:»function»,»z»:»b787c9.7c541238»,»name»:»Обробка даних»,»func»:»msg.payload = `Температура: ${msg.payload.temperature}°C, Вологість: ${msg.payload.humidity}%`; return msg;»,»outputs»:1,»x»:270,»y»:100,»wires»:[[«e3»]]},
{«id»:»e3»,»type»:»debug»,»z»:»b787c9.7c541238»,»name»:»Вивід даних»,»console»:»false»,»complete»:»payload»,»x»:470,»y»:100,»wires»:[]}]
```

Автоматизація освітлення:

1. Вузли: Датчик руху, логічний вузол (наприклад, switch), та вузол реле для контролю освітлення.
2. Логіка: Якщо датчик руху активний, ввімкнути світло на 5 хвилин.
3. Код Node-RED:

```
[{«id»:»a1»,»type»:»motion-sensor»,»z»:»b787c9.7c541238»,»name»:»PIR Sensor»,»x»:50,»y»:200,»wires»:[[«a2»]]},
{«id»:»a2»,»type»:»switch»,»z»:»b787c9.7c541238»,»name»:»Check Motion»,»property»:»payload»,»propertyType»:»msg»,»rules»:[{«t»:»eq»,»v»:»true»,»vt»:»bool»}],»checkall»:»true»,»outputs»:1,»x»:250,»y»:200,»wires»:[[«a3»]]},
{«id»:»a3»,»type»:»relay»,»z»:»b787c9.7c541238»,»name»:»Light Relay»,»action»:»toggle»,»duration»:»300»,»x»:450,»y»:200,»wires»:[]}]
```

Використання Node-RED Dashboard: Створення інтерфейсу для візуалізації та контролю системи. Використання графічних вузлів, таких як графіки, текстові поля, кнопки, для створення інтерактивного дашборду.

Використання e-mail або SMS вузлів: Налаштування вузлів для надсилання сповіщень користувачам при виявленні критичних змін у параметрах середовища або системи безпеки [37].

Інтеграція з Amazon Alexa або Google Assistant: Використання вузлів Node-RED для прийому голосових команд та виконання відповідних дій в системі «Розумного будинку».

Інтеграція системи у контексті «Розумного будинку» включає з'єднання всіх апаратних та програмних компонентів для створення єдиної функціональної мережі [27]. Це передбачає правильне підключення всіх сенсорів, актуаторів, та інших пристроїв до Raspberry Pi, налаштування комунікацій та забезпечення зв'язку між ними через програмне забезпечення, зокрема Node-RED.

З'єднання апаратних компонентів

1. Сенсори та Актуатори:

- Підключення: Кожен сенсор або актуатор підключається до Raspberry Pi через GPIO піни або через додаткові модулі комунікації, як-от I2C чи SPI.
- Протоколи зв'язку: Використання стандартних протоколів, таких як MQTT для IoT пристроїв, забезпечує надійну комунікацію між компонентами.

2. Мережеве обладнання:

- Маршрутизатори та комутатори: Забезпечення міцної мережевої інфраструктури для взаємодії всіх компонентів системи, включаючи Wi-Fi та Ethernet підключення.

Налаштування програмного забезпечення

1. Node-RED:

- Конфігурація вузлів: Налаштування вузлів у Node-RED для читання даних з сенсорів, керування актуаторами, та логічної обробки даних.

- Потоки даних: Створення потоків, які ефективно збирають, обробляють та реагують на дані у реальному часі.

2. Програмне забезпечення для керування даними:

- Бази даних: Використання баз даних, таких як InfluxDB для збору та аналізу часових рядів даних з сенсорів.
- Веб-інтерфейс: Розробка користувацького інтерфейсу за допомогою Node-RED Dashboard для візуалізації та керування системою.

Автоматизація та інтеграція протоколів:

1. MQTT:

- Broker: Встановлення MQTT broker на Raspberry Pi для управління повідомленнями між IoT пристроями.
- Підписка та публікація: Конфігурація IoT пристроїв для підписки та публікації на топіки MQTT, забезпечуючи надійний обмін даними.

2. Home Assistant:

- Інтеграція з Node-RED: Використання Home Assistant як централізованої платформи для управління «Розумним будинком», з інтеграцією через Node-RED для розширеного керування та автоматизації.

Тестування та Валідація:

- Функціональне тестування: Перевірка кожного аспекту системи, від відгуку сенсорів до виконання команд актуаторами.
- Валідація комунікації: Забезпечення, що всі компоненти системи можуть надійно спілкуватися один з одним.

Ефективна автоматизація «Розумного будинку» залежить від злагодженої взаємодії між програмним та апаратним забезпеченням [26]. Цей розділ описує, як програмування та автоматизація інтегруються для керування різними системами в будинку, такими як освітлення, клімат-контроль, безпека та розваги.

Сценарії для різних типів обладнання:

1. Автоматизація освітлення:

- Розумні перемикачі та розумне освітлення: Інтеграція перемикачів з датчиками руху для автоматичного вмикання/вимикання світла. Використання таймерів та сценаріїв для адаптації освітлення відповідно до часу доби.
- Контроль через мобільний додаток та голосові команди: Інтеграція з платформами як Google Home або Amazon Alexa для зручного керування освітленням.

2. Клімат-контроль:

- Термостати та HVAC системи: Програмування термостатів для оптимізації енергоспоживання та комфорту. Використання даних з температурних сенсорів для автоматичної адаптації налаштувань.
- Інтеграція з погодними сервісами: Автоматичне налаштування системи HVAC на основі змін у погодних умовах через онлайн погодні API.

3. Системи безпеки:

- Датчики руху та відеонагляд: Автоматичне включення запису або сповіщень при виявленні руху. Інтеграція з мобільними додатками для миттєвого повідомлення користувачів.
- Розумні замки та системи контролю доступу: Програмування замків для віддаленого керування та моніторингу доступу до будинку.

4. Розважальні системи:

- Медіа сервери та мультимедіа: Автоматизація відтворення медіа контенту залежно від сценаріїв (наприклад, вечірнє кіно).
- Інтеграція з розумними телевізорами та аудіосистемами: Використання голосових команд для керування медіа системами через інтеграцію з віртуальними помічниками.

Інтеграція з зовнішніми API:

1. Використання API для покращення функціоналу: Інтеграція з погодними сервісами, календарями, медіа сервісами для забезпечення контекстно-залежної автоматизації.

2. Безпека та конфіденційність при використанні API: Забезпечення захисту даних через шифрування та безпечні методи автентифікації.

Особливості програмування для автоматизації:

1. Модульність та масштабованість: Створення модульних та легко адаптованих сценаріїв, що дозволяють легко додавати нові пристрої та сервіси.

2. Тестування та налагодження: Ретельне тестування кожного сценарію автоматизації для забезпечення стабільності та надійності системи.

Безпека та конфіденційність є критично важливими аспектами у системах «Розумного будинку», оскільки вони забезпечують захист від несанкціонованого доступу та зловмисного використання особистих даних. На захисті цих аспектів має бути зосереджена увага під час проектування, розробки, імплементації та експлуатації розумних систем.

Заходи безпеки для IoT пристроїв:

- Шифрування даних: Використання стандартів шифрування для захисту даних, які передаються між пристроями та серверами. Це включає шифрування зв’язку через SSL/TLS для всіх мережевих транзакцій.
- Безпечне зберігання даних: Впровадження рішень для безпечного зберігання даних на пристроях, таких як зашифровані бази даних або використання надійних хмарних рішень з сильними політиками безпеки.
- Аутентифікація та авторизація: Забезпечення, що доступ до системи мають тільки авторизовані користувачі. Використання багатофакторної аутентифікації, управління доступом на основі ролей.

Проблеми конфіденційності даних

1. Збір та використання даних: Переконайтеся, що користувачі знають, які дані збираються, і надають згоду на їх використання. Впровадження політик конфіденційності, що відповідають законодавству, наприклад GDPR у Європі.

2. Мінімізація даних: Збір та зберігання лише тих даних, які необхідні для роботи системи, та їх видалення після того, як вони більше не потрібні.

3. Моніторинг та виявлення вторгнень: Встановлення систем моніторингу для виявлення та реагування на потенційні безпекові інциденти. Використання сучасних технологій аналізу даних для раннього виявлення зловмисних дій.

Реалізація заходів безпеки та конфіденційності:

1. Розробка безпечної архітектури: Включення безпеки та конфіденційності на ранніх етапах проектування системи, щоб забезпечити інтеграцію необхідних заходів безпеки на всіх рівнях.

2. Навчання та свідомість користувачів: Забезпечення того, щоб користувачі були обізнані щодо потенційних ризиків та методів захисту своїх даних.

Тестування та оптимізація є ключовими процесами у розробці систем «Розумного будинку», забезпечуючи їхню надійність, ефективність і безпеку. Цей розділ описує методи і стратегії, які можна застосувати для перевірки функціональності і підвищення продуктивності системи.

Методи тестування

1. Функціональне тестування:

- Ціль: Переконатися, що всі функції системи відповідають вимогам специфікації.
- Методи: Використання тестових сценаріїв для кожної функції, включаючи включення та вимкнення розумного освітлення, реакцію системи на команди датчиків, тощо.

2. Тестування навантаження:

- Ціль: Перевірка здатності системи справлятися з максимальними та піковими навантаженнями.
- Методи: Симуляція одночасної роботи всіх компонентів системи, включаючи одночасну активацію всіх сенсорів та актуаторів.

3. Тестування безпеки:

- Ціль: Виявлення потенційних вразливостей у системі.

- Методи: Проведення аудиту безпеки, пентестування, перевірка на вразливість до зовнішніх атак.

4. Тестування стійкості:

- Ціль: Перевірка здатності системи працювати стабільно протягом тривалого часу під звичайним навантаженням.
- Методи: Запуск системи на тривалий час, моніторинг ресурсів, пошук витоків пам’яті та інших проблем.

Оптимізація системи:

1. Покращення продуктивності:

- Аналіз профілювання: Використання інструментів для моніторингу використання ресурсів (ЦПУ, пам’ять, мережевий трафік).
- Рефакторинг коду: Покращення алгоритмів обробки даних, оптимізація запитів до баз даних.

2. Мінімізація енергоспоживання:

- Ефективне управління живленням: Автоматизація відключення неактивних пристроїв, оптимізація графіків роботи систем HVAC.
- Використання енергоефективних компонентів: Перехід на більш ефективні моделі сенсорів та актуаторів.

3. Забезпечення масштабованості:

- Архітектурні рішення: Планування системи так, щоб вона могла легко розширюватися, додавання нових компонентів без перепроектування основних систем.

Впровадження та експлуатація «Розумного будинку» є завершальними етапами проекту, де система вводиться в дію та використовується на повну потужність. Ці процеси включають реалізацію проектованих рішень, забезпечення їх сталої роботи та адаптацію до потреб користувачів, тож необхідно мати інструкцію з експлуатації.

Інструкції з експлуатації:

1. Запуск системи**:**

- Поетапне введення в експлуатацію кожного компонента системи з детальними перевірками їх працездатності.
- Встановлення та конфігурація обладнання та програмного забезпечення згідно з технічною документацією.

2. Користувацькі налаштування**:**

- Налаштування параметрів системи (освітлення, температура, системи безпеки) згідно з індивідуальними вподобаннями користувачів.
- Навчання користувачів використовувати систему через інтерактивні дашборди, мобільні додатки, та голосові помічники.

3. Підтримка та обслуговування**:**

- Регулярне обслуговування обладнання для запобігання поломок та зниження зносу.
- Надання технічної підтримки користувачам для вирішення можливих проблем та відповідей на запитання.

Рекомендації щодо обслуговування

1. Моніторинг системи**:**

- Використання програмного забезпечення для моніторингу стану системи в реальному часі та виявлення неполадок.
- Встановлення сповіщень про критичні події, щоб швидко реагувати на потенційні проблеми.

2. Оновлення програмного забезпечення**:**

- Регулярне оновлення ПЗ для усунення помилок, поліпшення функціональності та забезпечення безпеки системи.
- Використання автоматизованих інструментів для розгортання оновлень з мінімальним втручанням користувачів.

3. Технічна документація**:**

- Підтримка актуальності технічної документації та керівництв користувачів для забезпечення легкості обслуговування та використання системи.
- Надання онлайн-ресурсів та бази знань для самостійного вирішення типових проблем користувачами.

Аналіз викликів у контексті «Розумних будинків» включає ідентифікацію та розгляд потенційних проблем, які можуть вплинути на ефективність, безпеку, та прийняття технологій користувачами. Цей розділ допомагає визначити типові та потенційні проблеми, а також розробити стратегії для їх усунення або мінімізації [35].

Типові проблеми та їх рішення:

1. Технічні несправності**:**

- Проблема**:** Відмови обладнання, збої програмного забезпечення.
- Рішення**:** Розробка системи з високим рівнем відмовостійкості, включення резервних компонентів, регулярне оновлення та підтримка ПЗ.

2. Проблеми з сумісністю**:**

- Проблема**:** Конфлікти між різними компонентами системи, складність інтеграції нових пристроїв.
- Рішення**:** Використання стандартизованих протоколів та інтерфейсів, участь у відкритих екосистемах.

3. Безпека**:**

- Проблема**:** Вразливості до кібератак, несанкціонований доступ.
- Рішення**:** Застосування сучасних методів криптографічного захисту, регулярні аудити безпеки, впровадження політик управління доступом.

4. Проблеми з конфіденційністю**:**

- Проблема**:** Ризик витоку персональних даних.
- Рішення: Застосування політик мінімізації даних, шифрування даних, надання користувачам контролю над їхніми даними.

Вплив зовнішніх факторів:

1. Зміни в законодавстві**:**

- Вплив**:** Нові закони можуть вимагати змін у способі збору, зберігання та обробки даних.
- Дії**:** Постійний моніторинг законодавчих змін, адаптація політик та процедур згідно з новими вимогами.

2. Технологічні зміни**:**

- Вплив: Розвиток нових технологій може зробити поточні рішення застарілими.
- Дії: Впровадження стратегії постійного навчання та інновацій, регулярне оновлення обладнання та програмного забезпечення.

Підготовка до майбутніх викликів:

1. Стратегічне планування**:** Розробка довгострокових планів для впровадження нових технологій та мінімізації потенційних ризиків.
2. Гнучкість системи**:** Проектування системи таким чином, щоб її можна було легко адаптувати до змін у технологіях і законодавстві.
3. Залучення спільноти**:** Співпраця зі спільнотою розробників та користувачів для обміну досвідом та кращими практиками.

Майбутнє «Розумних будинків» виглядає обіцяюче, з новими технологічними інноваціями та зростаючим прийняттям серед споживачів. Цей розділ досліджує ключові технологічні тенденції, що формують майбутнє домашньої автоматизації, а також вплив соціальних, економічних і законодавчих змін на індустрію «Розумних будинків».

Технологічні тенденції:

1. Штучний інтелект та машинне навчання:

- Впровадження: Використання AI для аналізу поведінки користувачів і автоматичне налаштування систем з метою підвищення комфорту та енергоефективності.

- Перспективи: Розвиток самонавчальних систем, які можуть передбачати потреби користувачів та адаптуватися до змін у їхніх звичках без зовнішнього втручання.

2. Інтернет речей (IoT):

- Впровадження: Посилення інтеграції домашніх приладів та систем через мережеві технології.
- Перспективи: Повна інтеграція домашнього обладнання в єдину екосистему, що забезпечує взаємодію всіх пристроїв у реальному часі.

3. Блокчейн:

- Впровадження: Застосування технології блокчейн для забезпечення прозорості та безпеки в транзакціях між пристроями.
- Перспективи: Використання блокчейну для створення децентралізованих мереж «Розумних будинків», де користувачі можуть контролювати свої дані та взаємодіяти з іншими без посередників.

Соціальні та економічні перспективи:

1. Привабливість для споживачів:

- Фактори: Збільшення інтересу до енергозбереження, безпеки, та зручності.
- Перспективи: Розширення ринку «Розумних будинків», особливо серед молодого покоління, яке є більш відкритим до нових технологій.

2. Економічні аспекти:

- Фактори: Зниження вартості компонентів IoT та спрощення впровадження технологій.
- Перспективи: Доступність «Розумних будинкових» систем для ширшого кола споживачів, зростання конкуренції та інновацій.

Законодавчі та регуляторні виклики:

1. Нормативна база:

- Проблематика: Розробка та імплементація законів, що регулюють збір та обробку персональних даних, безпеку IoT пристроїв.
- Перспективи: Створення міжнародних стандартів для регулювання технологій «Розумних будинків», що може сприяти їх ширшому прийняттю.

3.3 Висновок третього розділу

У висновку цього розділу необхідно виділити наступні пункти

1. Технічні аспекти:

- Важливість вибору відповідного апаратного та програмного забезпечення.
- Значення інтеграції різних систем і технологій для створення справді інтерактивного та автоматизованого дому.

2. Програмування та автоматизація:

- Роль Node-RED як потужного інструменту для візуального програмування та автоматизації складних систем.
- Необхідність глибокої інтеграції з системами дому для забезпечення їх ефективності.

3. Безпека та конфіденційність:

- Важливість забезпечення безпеки та захисту персональних даних у системах «Розумного будинку».
- Рекомендації щодо впровадження сучасних заходів безпеки для мінімізації ризиків.

4. Виклики та вплив зовнішніх факторів:

- Аналіз потенційних труднощів у впровадженні і експлуатації систем «Розумних будинків».
- Стратегії подолання цих викликів з використанням інноваційних технологічних рішень.

Напрямки для подальших досліджень:

1. Розширення технологічних можливостей:

- Подальше дослідження в області штучного інтелекту та машинного навчання для розробки більш автономних та адаптивних систем.
- Вивчення нових методів інтеграції більшої кількості побутових пристроїв в єдину екосистему.

2. Поліпшення інтерфейсів користувача:

- Розробка більш інтуїтивних і доступних інтерфейсів для користувачів всіх вікових груп.
- Забезпечення більшої персоналізації налаштувань та управління для забезпечення максимального комфорту та задоволення користувачів.

3. Законодавчі та етичні аспекти:

- Вивчення впливу законодавчих змін на розвиток технологій «Розумних будинків».
- Розробка рекомендацій та норм, які б забезпечували етичне використання технологій у приватному просторі.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗИВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

4.1.1 Шкідливі та небезпечні фактори при користуванні ПК

Одним із найважливіших елементів будь-якого підприємства є робоче місце, в межах якого відбувається цілеспрямована діяльність (тобто праця) конкретного працівника. Робоче місце — це частина виробничого простору одного або групи працівників, оснащена основним і допоміжним технологічним обладнанням, інвентарем, інструментом, робочими меблями, необхідними для виробництва певного виду робіт. [46] З розвитком виробничих процесів та інформаційні технології все частіше робочі місця працівників оснащуються персональним комп'ютером. Однак їх використання загострило проблеми збереження власного та суспільного здоров'я, вимагає удосконалення існуючих та розробки нових підходів до організації робочих місць, проведення профілактичних заходів для запобігання розвитку негативних наслідків впливу на здоров'я робітників. [47].

Враховуючи той факт, що виконання досліджень вимагає використання засобів обчислювальної техніки, то необхідним є створенням безпечних умов праці при використанні ПК. Також потрібно приділити особливу увагу питанням електро- та пожежної безпеки.

З урахуванням вимог ДСанПіН 3.3.2.007-98 та НПАОП 0.00-7.15-18 необхідно визначити небезпечні і шкідливі фактори, які впливають на користувачів ПК при експлуатації, дослідити ці фактори, розглянути їхній вплив, принципи їх нормування і способи запобігання їхнього шкідливого впливу на людину .

Основними шкідливими та небезпечними виробничими факторами, які пов’язані з використанням ПК, є такі [48]:

1. електромагнітне випромінювання радіочастотного діапазону;
2. випромінювання оптичного діапазону (ультрафіолетове, інфрачервоне і випромінювання видимого діапазону);
3. електростатичне поле;
4. недостатня освітленість робочої зони;
5. підвищений рівень шуму;
6. значна напруга зорових органів і пов’язане з цим перевтомлення користувача ПК;
7. значне навантаження на пальці і кисті рук, що при відсутності профілактики і медичного контролю, може викликати професійні захворювання;
8. тривале перебування в одному й тому ж самому положенні сидячи, що викликає застійні явища в організмі людини;
9. відблиски на екрані монітора;
10. можливість ураження електричним струмом;
11. можливість виникнення пожежі.

Облаштування робочих місць, обладнаних відеотерміналами, повинно забезпечувати: [40]

- належні умови освітлення приміщення і робочого місця, відсутність відблисків;
- оптимальні параметри мікроклімату (температура, відносна вологість, швидкість руху та рівень іонізації повітря);
- належні ергономічні характеристики основних елементів робочого місця.

А також необхідно вжити заходи проти таких небезпечних і шкідливих чинників, як наявність шуму та вібрації, електромагнітного, ультрафіолетового і

інфрачервоного випромінювання, електростатичного поля між екраном монітору і оператором, наявність пилу, озону, оксидів азоту й аероіонізації.

Приміщення і їх частини, в яких розташовуються ПК та частини системи «Розумний будинок», повинні мати не нижче II ступеня вогнестійкості.

Службові приміщення, в яких розташовані ПК, не повинні межувати з приміщеннями, де рівні шуму та вібрації перевищують норму (механічні цехи, майстерні тощо).

Відповідно до ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» площа приміщення, у яких розташовують відеотермінали, визначається, виходячи з розрахунку на одне робоче місце - не менше 6,0 кв. м, об’єм - не менше 20,0 куб. м, з урахуванням максимальної кількості осіб, які одночасно працюють у зміні. [48]:

Загальне освітлення має бути виконане у вигляді суцільних або переривчастих ліній світильників, що розміщуються збоку від робочих місць (переважно зліва) паралельно лінії зору працівників. Як джерело світла при штучному освітленні повинні застосовуватися, як правило, люмінесцентні лампи типу ЛБ. Допускається у світильниках місцевого освітлення застосовувати лампи розжарювання.

Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300-500 лк. У разі неможливості забезпечити даний рівень освітленості системою загального освітлення допускається застосування світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрана та збільшення освітленості екрана більше ніж до 300 лк.

При дотриманні цих вимог, виконання всіх видів робіт на ПК при дослідженні методів та засобів адаптивної селекції каналів зв’язку для

«Розумного будинку», є безпечним з точки зору охорони праці, техніки безпеки та протипожежної безпеки. [43]

4.1.2 Електробезпека при користуванні системою «Розумний будинок»

З розвитком систем «Розумний будинок», збільшується кількість побутових приладів, які взаємодіють з електричними мережами. Це ставить підвищені вимоги до електробезпеки, адже неуважне або неправильне використання таких систем може призвести до аварійних ситуацій, зокрема короткого замикання, перевантаження мережі та, в крайньому випадку, до пожежі. Важливо розробити та дотримуватись чітких принципів безпеки під час використання систем «Розумного будинку».

Основні аспекти електробезпеки: [39]

1. Правильне проектування та інсталяція

- Кваліфіковане встановлення: Всі електричні інсталяції та підключення мають виконуватися кваліфікованими фахівцями, які знають норми та правила установки електрообладнання.
- Використання якісних матеріалів: Обрання надійних і сертифікованих матеріалів та компонентів, які відповідають стандартам безпеки.

2. Захист від перевантаження та коротких замикань

- Встановлення захисних пристроїв: Використання автоматичних вимикачів та диференційних реле, які можуть автоматично відключити живлення у разі виявлення перевантаження або витоку струму.
- Перевірка та обслуговування системи: Регулярний огляд електричної системи для виявлення та усунення можливих несправностей.

3. Захист від перепадів напруги

- Використання стабілізаторів напруги: Захист чутливих електронних компонентів від перепадів напруги, які можуть призвести до їх виходу з ладу.
- Застосування блискавкозахисту: Наявність захисту для відведення блискавкових розрядів, особливо в регіонах з високою активністю гроз.

4. Інтелектуальне управління навантаженням

- Програмування системи управління: Налаштування автоматизованої системи для оптимізації споживання електроенергії та уникнення надмірного навантаження на електромережу.
- Моніторинг енергоспоживання: Використання сенсорів та програмного забезпечення для моніторингу і контролю споживання енергії в реальному часі.

Безпека в системах «Розумний будинок» не обмежується лише технічними аспектами встановлення та експлуатації. Вона також включає освіту та обізнаність користувачів про ризики та правильне використання електрообладнання. Розуміння основ електробезпеки дозволяє кожному користувачеві знизити ризик аварій та забезпечити безпечне використання інноваційних технологій у своєму домі.

Приміщення де розміщено систему по рівню безпеки ураження людей електричним струмом згідно ПУЕ можна віднести, до помешкань без підвищеної небезпеки, тому що:

1. відносна вологість повітря не перевищує 75%;
2. матеріал підлоги (паркет) є діелектриком;
3. температура повітря не досягає значень, більших +30 °С/ [41]

Відсутня можливість одночасного дотику людини до з’єднаних з землею металоконструкцій будівлі, технологічних апаратів, механізмів і т.п., з одного боку, і до металевих корпусів електроустаткування - з іншого.

Для того, щоб не допустити ураження мешканця електричним струмом при виникненні аварійних ситуацій, необхідно заземлити все обладнання, що працює від мережі 220 В, 50 Гц. Опір нульового дроту повинен бути таким, щоб при замиканні на корпус або нульовий дріт; виникав струм короткого замикання, сила якого повинна перевищувати в 1,4 рази номінальний струм спрацювання автомата струмового захисту (при струмі короткого замикання менше 100 А).

Заземлені конструкції, що знаходяться в приміщеннях, мають бути надійно захищені діелектричними щитками або сітками від випадкового дотику. У приміщеннях з ПК слід щоденно проводити вологе прибирання, повинні бути медичні аптечки першої допомоги, а також система автоматичної пожежної сигналізації.

4.2 Безпека в надзвичайних ситуаціях

В умовах сучасного життя, системи «Розумний будинок» не лише покращують комфорт і зручність, але й сприяють безпеці мешканців. Впровадження автоматизованих систем може значно зменшити ризики і забезпечити ефективне реагування на надзвичайні ситуації. Однак, для досягнення максимальної безпеки потрібне глибоке розуміння потенційних загроз і відповідне налаштування системи.

Ідентифікація потенційних ризиків: [42]

Важливий крок у забезпеченні безпеки — це аналіз та ідентифікація можливих надзвичайних ситуацій, які можуть виникнути в домашніх умовах:

- Пожежі: Спалахи внаслідок електричних несправностей або побутових інцидентів.
- Затоплення: Розриви труб або інші водні аварії.
- Газові витоки: Витік природного газу або інших небезпечних парів.

- Вторгнення: Несанкціонований доступ або спроби проникнення до приміщення.

Системи попередження та оповіщення:

- Детектори диму та тепла: Автоматичне виявлення ознак пожежі і сповіщення мешканців та служб порятунку.
- Датчики витоку води: Моніторинг приміщень на предмет можливого затоплення і автоматичне відключення водопостачання.
- Газові детектори: Виявлення витоків газу і аварійне відключення газопостачання, а також оповіщення мешканців.
- Системи безпеки: Відеоспостереження, датчики руху, і системи контролю доступу для забезпечення захисту від вторгнення.

Аварійні комунікаційні системи:

- Автоматичне повідомлення служб порятунку: Налаштування системи на автоматичне відправлення повідомлень у разі виявлення загроз.
- Інформування мешканців: Використання внутрішніх комунікаційних систем для інформування мешканців про надзвичайні ситуації та необхідні дії.

Планування евакуації:

- Евакуаційні маршрути: Розробка та візуалізація оптимальних шляхів евакуації з будинку.
- Інструкції для мешканців: Освітні програми і тренінги з евакуації та поведінки в екстрених ситуаціях.

Тестування та обслуговування:

- Регулярні перевірки: Тестування всіх систем безпеки на предмет їхньої працездатності та надійності.
- Оновлення програмного забезпечення: Постійне оновлення системного програмного забезпечення для усунення можливих вразливостей.

Безпека в надзвичайних ситуаціях є критично важливою складовою систем «Розумний будинок». Інтеграція розумних технологій в системи безпеки дозволяє не тільки автоматизувати багато процесів реагування на потенційні загрози, але й значно підвищує швидкість та ефективність цих відповідей, забезпечуючи максимальний захист для мешканців.

4.3 Висновок до четвертого розділу

Розглянуто критично важливі аспекти охорони праці та забезпечення безпеки в надзвичайних ситуаціях в контексті впровадження та експлуатації систем «Розумного будинку».

Основною метою було підкреслити значення ретельної підготовки, кваліфікованого встановлення обладнання та систематичного моніторингу засобів захисту для запобігання робочим травмам і аваріям в домашніх умовах.

З огляду на потенційні ризики, було визначено ключові заходи безпеки, серед яких:

1. Належне навчання та сертифікація працівників, зайнятих на встановленні та обслуговуванні систем «Розумного будинку».
2. Використання якісних і сертифікованих матеріалів та компонентів для забезпечення довговічності та безпеки експлуатації.
3. Регулярні перевірки і технічне обслуговування обладнання для виявлення та усунення потенційних несправностей.
4. Розробка та втілення ефективних планів евакуації та тренувань для мешканців, щоб мінімізувати ризики в надзвичайних ситуаціях.

Також було акцентовано на важливості впровадження передових технологій для автоматизації процесів реагування на аварійні ситуації, що включає автоматичне відключення пошкоджених систем, використання датчиків

виявлення диму, газу, та води, а також системи контролю доступу для підвищення рівня безпеки.

Завдяки цим заходам, система «Розумного будинку» не тільки забезпечує комфорт та зручність для своїх користувачів, але й гарантує високий рівень безпеки та готовність до адекватного реагування на потенційні небезпеки, що зробить життя в такому будинку не тільки комфортнішим, але й безпечнішим.

У роботі розглянуто питання охорони праці та безпеки в надзвичайних ситуаціях при впровадженні та використанні системи «Розумний будинок». Розглянуто особливі аспекти виникнення шкідливих та небезпечних факторів при роботі із системою керування «Розумний будинок» та передбачено заходи щодо їх попередження. Передбачено заходи з електробезпеки із врахуванням систем керування та збільшення кількості електроприладів, які використовуються в системі «Розумний будинок».

Варто розуміти, що під час воєнного стану важливо подбати про ефективні та своєчасні методи оповіщення працівників про повітряну тривогу та забезпечити їх надійним укриттям, яке обладнане згідно всіх норм.

ВИСНОВКИ

У межах даного дослідження було проаналізовано можливості мікроконтролерів Arduino та Raspberry Pi, розглянуто можливості систем «Розумного будинку», таких як: клімат-контроль, безпека і відеоспостереження, протікання води та витоку газу, можливості штучного інтелекту в контексті «Розумного будинку». Розроблено сайт, в якому передбачено функції моніторингу температури, рівню вологості, газу та наявності руху, також реалізовано графік температури за останні 7 днів. Проведено аналіз системи керування внутрішнім середовищем «Розумного будинку», що базується на мікроконтролері Raspberry Pi, наведено код для підключення відповідних датчиків до мікроконтролера, а також можливості підключення до сайту. Цей проект демонструє, як можна інтегрувати різні технологічні компоненти для створення ефективної, гнучкої та безпечної системи домашньої автоматизації. Основною метою роботи було не тільки підвищення комфорту користувачів, але й забезпечення енергоефективності та підвищення безпекових стандартів.

Детальний огляд результатів дослідження:

1. Технічна реалізація: Було успішно створено і запущено веб-сайт для моніторингу та управління системами «Розумного будинку». Інтерфейс користувача був розроблений з орієнтацією на зручність і інтуїтивність використання, що дозволяє користувачам легко керувати параметрами свого житла.

2. Інтеграція та функціональність: Система успішно інтегрувала ряд датчиків для моніторингу основних показників домашнього середовища, таких як температура, вологість і світло.

3. Безпека та надзвичайні ситуації: Було реалізовано автоматичні протоколи реагування на надзвичайні ситуації, включно з витоком води, газу та диму, що дозволяє швидко ідентифікувати потенційні загрози і вживати заходів для їх усунення.

Внесок у наукову галузь і практичне застосування:

Ця робота зробила значний внесок у галузь домашньої автоматизації:

1. Науковий внесок: Розширення наявних знань щодо застосування мікроконтролерів у системах «Розумного будинку», зокрема, в контексті інтеграції різноманітних датчиків і управління домашніми процесами.

2. Практичний внесок: Розробка надійної, масштабованої та вартісно ефективної системи, яка може бути адаптована для різних типів житла і користувацьких потреб.

Рекомендації для майбутніх розробок

1. Розширення функціоналу: Інтеграція штучного інтелекту та машинного навчання для прогнозування поведінки користувачів та автоматичної адаптації системи до їхніх звичок та потреб.

2. Забезпечення кібербезпеки: Посилення захисту даних і приватності користувачів з використанням передових технологій шифрування та аутентифікації.

3. Екологічність та сталість: Розвиток функцій системи для оптимізації споживання ресурсів, зокрема енергії та води, для сприяння сталому розвитку та зменшення впливу на довкілля.

Ці рекомендації мають потенціал не тільки підвищити ефективність та функціональність систем «Розумного будинку», але й забезпечити їхнє безпечне, стале та користувацьки орієнтоване майбутнє.

ПЕРЕЛІК ДЖЕРЕЛ

1. Granzer W. P. Security in Building Automation Systems / Wolfgang Praus Granzer. Munich: Appress, 2018. – 578 с.

2. Що таке розумний будинок? Все що потрібно знати про систему Розумний Дім [Електронний ресурс]. – Режим доступу до ресурсу: https://bron.ua/article/scho- take-rozumnij-budinok-vse-scho-potrbno-znati-pro-sistemu-rozumnij-dm/5/

3. Розумне освітлення [Електронний ресурс]. – Режим доступу до ресурсу: https://milight.com.ua/ua/umnoe-osveshchenie/

4. Розумний дім [Електронний ресурс] - https://stylus.ua/uk/articles/528.html interesting/what_is_mp_mc_plc.php

5. Arduino UNO [Електронний ресурс] - https://uk.wikipedia.org/wiki/Arduino_Uno

6. Raspberry Pi [Електронний ресурс] - https://uk.wikipedia.org /wiki/Raspberry_Pi

7. [Електронний ресурс] – Режим доступу до ресурсу: https://pupenasan.github.io/TI40/%D0%9B%D0%B5%D0%BA%D1%86/2_nodered.html

8. [Електронний ресурс] – Режим доступу до ресурсу: https://er.knutd.edu.ua/bitstream/123456789/24290/5/Dyplom123_Buryak_Zlotenko.pdf

9. Наконечний А. Й., Верес З. Є. Інтернет речей і сучасні технології. Вісн. Нац. ун-ту «Львів. політехніка». Автоматика, вимірювання та керування. 2016. № 852. С. 3-9.

10. Строкань О.В., Литвин Ю.О. Програмні рішення з підвищення ефективності управління параметрами мікроклімату у закритому приміщенні. Матеріали міжнародного науково-практичного форуму «Сучасні наукові дослідження на шляху до Євроінтеграці» (21-22 червня 2019 р.) Таврійський

державний агротехнологічний університет імені Дмитра Моторного. Мелітополь: ФОП Однорог Т.В., 2019. Частина 2. С. 105-107.

11. Бєлов А. В. Розробка пристроїв на мікроконтролерах AVR: крокуємо від «чайника» до профі. СПб. : Наука та Техніка, 2013. 528 с.

12. Грищук Ю. С. Мікропроцесорні пристрої. Харків : Вид-во НТУ ХПІ, 2007. 280 с.

13. Електронні системи розумного будинку з підвищеною ефективністю URL: https://ela.kpi.ua/bitstream/123456789/42426/1/KalininDV_bachelor.pdf.

14. Розробка системи «Розумний будинок» на базі «Arduino» URL: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa2018/paper/download/4541/4612.

15. Система моніторингу для розумного дому(клієнтська частина) URL: https://ela.kpi.ua/bitstream/123456789/34845/1/Dobrovolskyi_bakalavr.pdf.

16. *Головна - Репозитарій Вінницького Національного Технічного Університету*.
URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/32491/Презентація.pdf?sequence=1&isAllowed=y.

17. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. *Офіційний вебпортал парламенту України*. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98#Text.

18. ЛР1. Основи роботи з Node-RED. *Школа автоматики*. URL: http://edu.asu.in.ua/mod/book/tool/print/index.php?id=123.

19. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. *Офіційний вебпортал парламенту України*. URL: https://zakon.rada.gov.ua/laws/show/z0508-18#Text.

20. Що таке мікропроцесор, мікроконтролер та програмований логічний контролер. URL: https://elprivod.nmu.org.ua/ua/interesting/what_is_mp_mc_plc.php.

21. Bell C. Beginning Sensor Networks with Arduino and Raspberry Pi. *SpringerLink*. URL: https://link.springer.com/book/10.1007/978-1-4302-5825-4.

22. В.О Церковний Я.В Литвиненко. Аналіз платформ Arduino, Raspberry Pi та ESP8266 в задачах моніторингу перевезень багажу. URL: https://elartu.tntu.edu.ua/handle/lib/40648.

23. Бачинський Я. Аналіз способів зв'язку роботів на базі мікроконтролерів Arduino. URL: https://elartu.tntu.edu.ua/handle/lib/30382.

24. К.К Зеленський Я.В Литвиненко. Давачі які застосовують в розумному будинку. URL: https://elartu.tntu.edu.ua/handle/lib/44393.

25. К.К Зеленський Я.В Литвиненко. Огляд мікроконтролерів для побудови розумного будинку. URL: https://elartu.tntu.edu.ua/handle/lib/44395.

26. Роман Золотий Андрій Недошитко. Розробка автоматизованих систем керування на базі контролерів Arduino. URL: https://elartu.tntu.edu.ua/handle/lib/24208.

27. *SumDU Repository: Home*. URL: https://essuir.sumdu.edu.ua/bitstream-download/123456789/49041/1/Dudarenko_rozumnyi_budynok.pdf.

28. Олександр Пупена. Технології індустрії: основи Node-RED. URL: https://pupenasan.github.io/TI40/Лекц/2_nodered.html.

29. A Hands-On Course in Sensors Using the Arduino and Raspberry Pi | Volk. *Taylor & Francis*. URL: https://www.taylorfrancis.com/books/mono/10.1201/9781351188319/hands-course-sensors-using-arduino-raspberry-pi-volker-ziemann.

30. Air quality monitoring system based on IoT using Raspberry Pi. *IEEE Xplore*. URL: https://ieeexplore.ieee.org/abstract/document/8230005.

31. Beginning Sensor Networks with Arduino and Raspberry Pi. *Google Books*. URL: https://books.google.com.ua/books?hl=uk&lr=&id=G5kQAwAAQBAJ&oi=fnd&pg=PP3&dq=raspberry+pi+arduino+differences&ots=ET

zzlxFVxY&sig=VG6lW7CQKeuDq9EjVNJWIlY_2fc&redir_esc=y#v=onepage&q=raspberry%20pi%20arduino%20differences&f=false.

32. Comparison of Wireless Communication Technologies used in a Smart Home: Analysis of wireless sensor node based on Arduino in home automation scenario. *DIVA*. URL: https://www.diva-portal.org/smash/record.jsf?pid=diva2:1118965&dswid=-6311.

33. Difference Between Arduino and Raspberry Pi [Comparison Table] | Webbylab. *webbylab*. URL: https://webbylab.com/blog/arduino-vs-raspberry-pi-key-differences-comparison-table/.

34. Getting Started With Raspberry Pi. *Google Books*. URL: https://books.google.com.ua/books?hl=uk&lr=&id=sdhyEAAAQBAJ&oi=fnd&pg=PT7&dq=raspberry+pi+arduino+differences&ots=zNOOrrU5fP&sig=pgVrVahRcxB006A51SkFjqWUZIU&redir_esc=y#v=onepage&q=raspberry%20pi%20arduino%20differences&f=false.

35. Make: Sensors. *Google Books*. URL: https://books.google.com.ua/books?hl=uk&lr=&id=6OaGAwAAQBAJ&oi=fnd&pg=PT2&dq=raspberry+pi+arduino+differences&ots=uWmEzO4M0Q&sig=XjmVpcRvzTY9if51V5-fI1yFrsE&redir_esc=y#v=onepage&q=raspberry%20pi%20arduino%20differences&f=false.

36. Prototype Wireless Controller System Based on Raspberry Pi and Arduino for Engraving Machine. *IEEE Xplore*. URL: https://ieeexplore.ieee.org/abstract/document/8359046.

37. Smart farm monitoring using Raspberry Pi and Arduino. *IEEE Xplore*. URL: https://ieeexplore.ieee.org/abstract/document/7219582.

38. *University of Babylon Private CDN*. URL: https://cdnx.uobabylon.edu.iq/research/HHd2HhzS90GUf8MuOeg.pdf.

39. Гандзюк, М. П. Основи охорони праці [Текст] : підручник / М. П.

Гандзюк, Є. П. Желібо, М. О. Халімовський ; за ред. М. П. Гандзюка ; МОН України. – 4-е видання. – К. : Каравела, 2008. – 384 с. – ISBN 966-8019-01-6.

40. Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин. Онлайн. Офіційний веб-портал парламенту України. [б. д.]. Режим доступу: https://zakon.rada.gov.ua/laws/show/z0293- 10#Text.

41. Гогіташвілі, Г. Г. Основи охорони праці [Текст] : навчальний посібник / Г. Г. Гогіташвілі, В. М. Лапін ; 4-те вид. випр. і доп. – К. : Знання, 2008. – 302 с. – ISBN 978-966-346-400-8.

42. Атаманчук, П. С., В. В. Мендерецький, О. П. Панчук та О. Г. Чорна. Безпека життєдіяльності [Текст] : навчальний посібник. – К. : Центр учбової літератури, 2011. – 276с. – ISBN 9786110102452.

43. Голінько В.І. Г 60 Охорона праці в галузі інформаційних технологій : навч. посіб. / В.І. Голінько, М.Ю. Іконніков, Я.Я. Лебедєв ; М-во освіти і науки України, Нац. гірн. ун-т. – Д. : НГУ, 2015. – 246 с.

44. Природнє і штучне освітлення: ДБН В.2.5-28-2006.

45. Санітарні норми виробничого шуму, ультразвуку та інфразвуку: ДСН 3.3.6.037-99-2000.

46. Ергономічні вимоги для організації робочого місця. Портал споживача. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.gpp.in.ua/robota/ergonomichni-vimogi-dlya-organizatsiji-robochogo-mistsya

47. ДСТУ 8604:2015. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. На заміну ГОСТ 12.2.032-78; чинний від 2017-07-01. Вид. офіц. Київ, 2015. 9 с. [Електронний ресурс]. – Режим доступу до ресурсу: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=71028

48. Бедрій І.Я., Нечай В.Я. Безпека життєдіяльності. Навчальний посібник. – Львів: Манголія 2006, 2007. 499 с

49. Основи охорони праці. / Під ред. Ткачука К.Н., Халімовського Н.О. – К.: Основа, 2006. 448 с.

50. Вимоги безпеки до робочого місця. Ергономічні вимоги. Pidru4niki. [Електронний ресурс]. – Режим доступу до ресурсу: https://pidru4niki.com/16850303/bzhd/vimogi_bezpeki_robochogo_mistsya_ergonomichni_vimogi.

51. Methodology of the Formation of Sports Matches Statistical Information Using Neural Networks Sorokivska, O., Lytvynenko, I., Sorokivskyi, O., Kozbur, H., Strutynska, I. CEUR Workshop Proceedings, 2023, 3628, pp. 389–403

52. Software for statistical processing and modeling of a set of synchronously registered cardio signals of different physical nature Lupenko, S., Lytvynenko, I., Sverstiuk, A., Horkunenko, A., Shelestovskyi, B. CEUR Workshop Proceedings, 2021, 2864, pp. 194–205

53. Software for segmentation, statistical analysis and modeling of surface ordered structures // I.V. Lytvynenko, P.O. Maruschak, S.A. Lupenko, Yu. I. Hats, A. Menou, S.V. Panin // MECHANICS, RESOURCE AND DIAGNOSTICS OF MATERIALS AND STRUCTURES (MRDMS-2016): Proceedings of the 10th International Conference on Mechanics, Resource and Diagnostics of Materials and Structures. AIP Publishing, 2016, Vol. 1785, No.1, pp. 030012-1-030012-7.

ДОДАТКИ

Додаток А

Лістинг програми Smart Home Dashboard

Лістинг файлу index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Smart Home Dashboard</title>
    <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstra
p.min.css" rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.1/css/all.min.css" rel="stylesheet">
    <link rel="stylesheet" href="styles.css">
    <style>
        .move-up {
            position: relative;
            top: -400px;
        }
    </style>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>

<body>
    <div class="container">
        <div class="header">
            <h1>Smart Home Dashboard</h1>
        </div>
        <div class="row">
            <div class="col-md-4">
                <div class="list-group">
                    <a href="#" class="list-group-item list-group-
item-action" onclick="changeRoom('living-room')">
                        Living Room
                    </a>

                    <a href="#" class="list-group-item list-group-
item-action" onclick="changeRoom('kitchen')">
```

```
                        Kitchen
                    </a>

                        <a href="#" class="list-group-item list-group-
item-action" onclick="changeRoom('bedroom')">
                            Bedroom
                        </a>
                        <a href="#" class="list-group-item list-group-
item-action" onclick="changeRoom('bedroom-2')">
                            Bedroom 2
                        </a>
                    </div>
                </div>
                <div class="col-md-8">
                    <div id="room-info" class="room-info"></div>
                    <canvas id="tempChart" class="temperature-
chart"></canvas>
                </div>
                <div class="col-md-8 move-up">
                    <img src="/prog/images/default-room.png" alt="Room
Image" class="img-fluid room-image">
                </div>

            </div>
        </div>

        <script src="script.js"></script>
</body>

</html>
```

Лістинг файлу styles.css

```
@import
url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;500&dis
play=swap');

body, html {
    margin: 0;
    padding: 0;
    font-family: 'Roboto', sans-serif;
    background: #f4f4f4; /* Новий світлий фон */
```

```
    color: #333; /* Колір тексту за замовчуванням */
    height: 100%;
    overflow-x: hidden;
}

.container {
    max-width: 1200px;
    margin: 0 auto;
    padding: 20px;
    background: #fff; /* Світлий фон контейнера */
    border-radius: 10px;
    box-shadow: 0 4px 8px 0 rgba(0,0,0,0.1); /* Легкий тінь */
}

.header {
    text-align: center;
    padding: 20px 0;
    font-size: 2.5rem;
    font-weight: 500;
    color: #555; /* колір заголовка */
}

.sidebar, .content, .room-info, .temperature-chart {
    color: #555; /* Колір тексту */
}

.list-group-item {
    background-color: #f4f4f4; /* Світлий фон для пунктів списку */
    color: #333; /* Колір тексту пунктів списку */
    border: 1px solid #ddd; /* Легкий контур */
    transition: background-color 0.3s ease;
}

.list-group-item:hover {
    background-color: #e0e0e0; /* Легке затемнення при наведенні */
}

.room-info {
    background: #f9f9f9; /* Світлий фон для інформації про кімнату */
    padding: 15px;
    margin-bottom: 20px;
    border-radius: 5px;
    border-left: 5px solid #6a11cb;
    box-shadow: 0 2px 4px 0 rgba(0,0,0,0.1);
}
```

```
.temperature-chart {
    background: #f9f9f9; /* Світлий фон для графіка температур */
    padding: 10px;
    border-radius: 5px;
    box-shadow: 0 2px 4px 0 rgba(0,0,0,0.1);
}

.room-image {
    width: 400px;
    height: 400px;
}
```

Лістинг файлу script.js

```
function updateChart(room, temperatures) {
    console.log('Updating chart for room:', room); // Це допоможе
зрозуміти, чи функція взагалі викликається
    const ctx =
document.getElementById('tempChart').getContext('2d');
    if (window.myChart) {
        console.log('Destroying old chart');
        window.myChart.destroy();
    }
    window.myChart = new Chart(ctx, {
        type: 'line',
        data: {
            labels: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat',
'Sun'],
            datasets: [{
                label: 'Temperature over the week (°C)',
                data: temperatures,
                backgroundColor: 'rgba(75, 192, 192, 0.2)',
                borderColor: 'rgba(75, 192, 192, 1)',
                borderWidth: 3,
                pointBackgroundColor: 'white',
                pointBorderColor: 'rgba(75, 192, 192, 1)',
                pointBorderWidth: 2,
                pointRadius: 5,
                tension: 0.4
            }]
        },
        options: {
            responsive: true,
```

```
            scales: {
                y: {
                    beginAtZero: false
                }
            }
        }
    });
    console.log('Chart updated');
}

const roomImages = {
    'living-room': '../../images/living_room.png',
    'kitchen': '../../images/kitchen.png',
    'bedroom': '../../images/bedroom.png',
    'bedroom-2': '../../images/bedroom2.png',
};


function changeRoom(room) {
    fetch(`http://localhost:3000/room-data/${room}`)
        .then(response => {
            if (!response.ok) {
                throw new Error('Network response was not ok');
            }
            return response.json();
        })
        .then(info => {
            const roomInfo = document.getElementById('room-info');
            roomInfo.innerHTML = `<h4>${room.replace('-', '
').toUpperCase()}</h4>
                                  <p><i class="fas fa-thermometer-
half"></i> Current Temperature: ${info.currentTemperature}°C</p>
                                  <p><i class="fas fa-wind"></i> CO2
Level: ${info.co2} ppm</p>
                                  <p><i class="fas fa-tint"></i>
Humidity: ${info.humidity}%</p>
                                  <p><i class="fas fa-walking"></i>
Motion: ${info.motion}</p>
                                  <p><i class="fas fa-bolt"></i>
Energy Efficiency: ${info.energyEfficiency}%</p>`;
            updateChart(room, info.temperatures);
            changeRoomImage(room);
        })
        .catch(error => {
            console.error('Error fetching data:', error);
```

```
            alert('Failed to load room data. Please check your
network connection and try again.');
        });
}

function changeRoomImage(room) {
    const roomImage = roomImages[room] || 'default-room.png';
    const roomImg = document.querySelector('.room-image');
    roomImg.src = roomImage;
}

document.addEventListener('DOMContentLoaded', () => {
    changeRoom('living-room'); // Default room to display

});

const express = require('express');
const app = express();
app.use(express.static('images')); //зберігаються статичні файли
```

Лістинг файлу server.js

```
const express = require('express');
const cors = require('cors');
const fs = require('fs');

const app = express();
const port = 3000;

app.use(cors());
app.use(express.json());

// Функція для читання даних з файлу
function readData() {
    const jsonData = fs.readFileSync('data.json');
    return JSON.parse(jsonData);
}

// API для отримання даних про кімнату
app.get('/room-data/:room', (req, res) => {
    const room = req.params.room;
```

```
    const data = readData();
    if (data[room]) {
        res.json(data[room]);
    } else {
        res.status(404).send('Room not found');
    }
});

app.listen(port, () => {
    console.log(`Server running on port ${port}`);
});
```

Лістинг файлу data.json

```
{
    "living-room": {
        "temperatures": [200, 500, 10, 18, 18, 18, 18],
        "currentTemperature": 24,
        "co2": 220,
        "humidity": 46,
        "motion": "test",
        "energyEfficiency": 80
    },
    "kitchen": {
        "temperatures": [19, 18, 20, 22, 21, 20, 19],
        "currentTemperature": 20,
        "co2": 500,
        "humidity": 50,
        "motion": "Not Detected",
        "energyEfficiency": 82
    },
    "bedroom": {
        "temperatures": [17, 18, 17, 16, 15, 16, 17],
        "currentTemperature": 20,
        "co2": 450,
        "humidity": 55,
        "motion": "Detected",
        "energyEfficiency": 88
    },
    "bedroom-2": {
        "temperatures": [20, 21, 20, 19, 20, 21, 20],
        "currentTemperature": 20,
        "co2": 480,
```

```
            "humidity": 48,
            "motion": "Not Detected",
            "energyEfficiency": 80
        }
    }
```

Додаток Б

Схема підключення датчиків в «Розумному будинку»