

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
магістра

(назва освітньо-кваліфікаційного рівня)

студента	<i>Деркача Артура Ігоровича</i> (ПІБ)		
академічної групи	<i>121М-22-2</i> (шифр)		
спеціальності	<i>121 Інженерія програмного забезпечення</i> (код і назва спеціальності)		
освітньої програми	<i>«121 Інженерія програмного забезпечення»</i> (назва освітньої програми)		
на тему:	<i>Розробка та дослідження ефективності впровадження програмного забезпечення розрахунку системи опалення в проєкті розумний будинок.</i>		

А.І.Деркач

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинг овою	інституційною	
розділів кваліфікаційної роботи				
спеціальний	<i>Проф. Бердник М.Г.</i>			

Рецензент				
-----------	--	--	--	--

Нормоконтролер				
----------------	--	--	--	--

Дніпро
2023

**Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»**

ЗАТВЕРДЖЕНО:

Завідувач кафедри
Програмного забезпечення комп'ютерних
систем

(повна назва)

Алексєєв М.О.

(підпис)

(прізвище, ініціали)

« »

2023 Року

**ЗАВДАННЯ
на виконання кваліфікаційної роботи магістра**

спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)

студенту 121М-22-2 Деркачу Артуру Ігоровичу
(група) (прізвище та ініціали)

Тема кваліфікаційної роботи Розробка та дослідження ефективності
впровадження програмного забезпечення розрахунку системи опалення в проєкті
розумний будинок.

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від від 09.10.2023 р. № 1227-с

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – процес розробки та впровадження програмного забезпечення для розрахунку системи опалення в проєкті розумного будинку.

Предмет досліджень – програмне забезпечення, для розрахунку оптимального режиму опалення розумного будинку.

Мета роботи – розробка інтегрованої системи керування елементами будинку, що базується на інтернеті речей (IoT) та інноваційних технологіях, за допомогою застосування для смартфона.

Методи дослідження – базуються на основних принципах оптимізації ресурсів, кореляційного аналізу, теорії баз даних. Також було використане математичне моделювання для опису залежностей температурних параметрів від різних чинників.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна – запропоновано використання алгоритмів оптимізації для мінімізації витрат енергії при забезпеченні комфортних умов в будинку.

Практична цінність полягає в тому, що результат дослідження можна використовувати при реалізації реальних проєктів розумних будинків і дозволяє оптимізувати енергію для опалення.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень мають бути подані у вигляді, що дозволяє побачити та оцінити безпосереднє використання запропонованих методів. В результаті роботи повинен бути розроблена інтегрована система керування елементами будинку, що базується на інтернеті речей (IoT) та інноваційних технологіях, за допомогою застосунку для смартфона.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Аналіз теми та постановка задачі.	15.04.2023-30.04.2023
Дослідження існуючих підходів до інтеграції системи керування елементами будинку, що базується на інтернеті речей (IoT).	01.05.2023-05.06.2023
Створення програмного забезпечення розрахунку системи опалення в проєкті розумний.	06.08.2023-01.12.2023

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект від реалізації результатів роботи очікується позитивним завдяки розробці оптимізованої системи витрат енергії для опалення.

Соціальний ефект від реалізації результатів роботи очікується позитивним, завдяки побутовому управлінню температурою будинку. Зменшення витрат енергії природно впливає на зниження екологічного відбитку.

Завдання видав

_____ (підпис)

Бердник М. Г.

_____ (прізвище, ініціали)

Завдання прийняв до виконання

_____ (підпис)

Деркач А.І.

_____ (прізвище, ініціали)

Дата видачі завдання: 15.04.2023 р.

Термін подання дипломного проєкту до ЕК 24.01.2024

РЕФЕРАТ

Пояснювальна записка: 74 с., 6 рис., 3 дод., 25 джерел.

Темою кваліфікаційної роботи є «Розробка та дослідження ефективності впровадження програмного забезпечення розрахунку системи опалення в проєкті розумний будинок».

Об'єкт дослідження: процес розробки та впровадження програмного забезпечення для розрахунку системи опалення в проєкті розумного будинку.

Предмет досліджень: програмне забезпечення, для розрахунку оптимального режиму опалення розумного будинку.

Мета кваліфікаційної роботи: розробка та імплементація інтегрованої системи керування елементами будинку, що базується на інтернеті речей (IoT) та інноваційних технологіях.

Методи дослідження: базуються на основних принципах оптимізації ресурсів, кореляційного аналізу, теорії баз даних. Також були використані методи математичного моделювання для опису залежностей температурних параметрів від різних чинників.

Наукова новизна: запропоновано використання алгоритмів оптимізації для мінімізації витрат енергії при забезпеченні комфортних умов в будинку.

Прогнози щодо розвитку досліджень: дослідження, описані в кваліфікаційній роботі, можуть стати базисом для розробки систем, які ефективно використовують відновлювані джерела енергії, такі як сонячні батареї або геотермальна енергія.

Практична цінність полягає в тому, що результат дослідження можна використовувати при реалізації реальних проєктів розумних будинків і дозволяє оптимізувати енергію для опалення.

Список ключових слів: оптимізація, IoT, додаток, кореляційний аналіз.

ABSTRACT

Explanatory note: 74 p., 6 fig., 3 app., 25 sources.

The topic of the qualification work is "Development and research of the effectiveness of the implementation of the heating system calculation software in the smart house project."

Research object: the process of development and implementation of software for calculating the heating system in a smart house project.

The subject of research is software aimed at calculating and optimizing the heating system in the context of a smart home project.

The purpose of the qualification work: development and implementation of an integrated control system for building elements based on the Internet of Things (IoT) and innovative technologies.

Research methods: based on the basic principles of resource optimization, correlation analysis, database theory. Mathematical modeling was also used to describe the dependence of temperature parameters on various factors.

Scientific novelty: the use of optimization algorithms is proposed to minimize energy consumption while providing comfortable conditions in the house.

Predictions for the development of the findings: the research described in the qualification work can become the basis for the development of systems that efficiently use renewable energy sources such as solar panels or geothermal energy.

The practical value is that the research result can be used in the implementation of real projects of smart houses and allows to optimize energy for heating.

List of keywords: optimization, IoT, application, correlation analysis.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ	12
1.1. Аналіз сучасних тенденцій у розвитку розумних домів	12
1.2. Проблеми розумних будинків, аналіз та контекстуалізація	13
1.3. Постановка задачі	25
1.4. Висновки	31
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	32
2.1. Загальна схема керуванням розумним будинком	32
2.2. Математичні моделі керуванням системами опалення	33
2.3. Опис використаних технологій та мов програмування	34
2.4. Розробка мобільного застосунку для Android	35
2.5. Мікросхема Arduino MEGA 2560rev3	44
2.6. Серверна частина	47
2.7. База даних	49
2.8 Висновки	53
РОЗДІЛ 3. РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ І ПРОГРАМНОГО ЗАБЕЗПЕННЯ КЕРУВАННЯМ ТЕМПЕРАТУРНИМ РЕЖИМОМ БУДИНКУ	54
3.1. Розробка математичної моделі оптимізації витрат енергії для опалення	54
3.2. Підключення до Arduino	61
3.3 Висновки	63
ВИСНОВКИ	65

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
ДОДАТОК А. СХЕМА АРХІТЕКТУРИ СИСТЕМИ«РОЗУМНИЙ БУДИНОК».....	70
ДОДАТОК Б. КОД ПРОГРАМИ.....	71
ДОДАТОК В. ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ.....	74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

APK – Android Package (формат архівних файлів-додатків для «Android»);

SDK – Software Development Kit (набір із засобів розробки);

PHP – Hypertext Preprocessor (гіпертекстовий препроцесор);

ПЗ – програмне забезпечення;

БД – база даних;

ІС – інформаційна система;

ОС – операційна система;

ІТ – інформаційні технології;

АВР – автоматичне введення резерву;

ШІ(AI) – штучний інтелект;

ІОТ – інтернет речей;

JS – Java script;

ММ – математична модель;

ЦП – центральний процесор;

РАМ – оперативна пам'ять.

ВСТУП

Розвиток сучасних технологій надає нам неймовірні можливості для зручного та ефективного способу життя. Цифрові інновації входять у всі сфери нашого існування, а однією з провідних концепцій, що проникає в наше повсякдення, є концепція "розумного дому".

Заснований на інтернеті речей (IoT) та інтелектуальних системах, розумний дім втілює мрію про інноваційне та інтелектуальне житло. Він дає змогу забезпечити високий рівень комфорту, енергоефективності та безпеки, дозволяючи жителям керувати різними аспектами будинку за допомогою мобільних пристроїв або голосових асистентів.

У сучасній інформаційній епохі, важливість розумних домів зростає експоненційно. Народження цієї ідеї пов'язане з бажанням створити сучасне житло, що враховує потреби сучасних життєвих реалій та вирішує складні завдання. Вимоги до житла зростають із зростанням нашого рівня знань, впровадженням новітніх технологій, збільшенням обсягу інформації та активізацією ритму життя.

Однак, разом зі зростанням інтересу до розумних домів, постають нові виклики та можливості. Суттєві проблеми, такі як захист персональних даних, стійкість до кібератак та ефективне використання енергії, стають актуальними в контексті розвитку цього нового напрямку в будівництві.

Ці виклики потребують нових досліджень та розробок у галузі розумних домів та керування елементами будинку. Тому, актуальність теми визначається сучасними викликами та необхідністю вдосконалення систем опалення у зв'язку з ростом інтересу до ефективного використання ресурсів у розумних будинках. Розробка програмного забезпечення для розрахунку оптимального режиму опалення стає ключовою у контексті прагнення до енергоефективності та сталого розвитку.

Об'єкт дослідження: процес розробки та впровадження програмного забезпечення для розрахунку системи опалення в проєкті розумного будинку.

Предмет досліджень: програмне забезпечення, для розрахунку оптимального режиму опалення розумного будинку.

Мета кваліфікаційної роботи: розробка та імплементація інтегрованої системи керування елементами будинку, що базується на інтернеті речей (IoT) та інноваційних технологіях.

Методи дослідження: базуються на основних принципах оптимізації ресурсів, кореляційного аналізу, теорії баз даних. Також було використане математичне моделювання для опису залежностей температурних параметрів від різних чинників.

Наукова новизна: запропоновано використання алгоритмів оптимізації для мінімізації витрат енергії при забезпеченні комфортних умов в будинку.

Прогнози щодо розвитку досліджень: дослідження, описані в кваліфікаційній роботі, можуть стати базисом для розробки систем, які ефективно використовують відновлювані джерела енергії, такі як сонячні батареї або геотермальна енергія.

Практична цінність полягає в тому, що результат дослідження можна використовувати при реалізації реальних проєктів розумних будинків і дозволяє оптимізувати енергію для опалення.

Кваліфікаційна робота складається з трьох розділів. У першому розділі зроблений аналіз літературних джерел, існуючих систем "розумних будинків" та технологій керування, проаналізовано попередні дослідження та роботи, пов'язані з даною темою, що дало зрозуміти актуальні проблеми та визначити постановку задачі кваліфікаційної роботи.

У другому розділі описуються методи та інструменти, які було використано для розробки проєкту "Розумний будинок". Було обґрунтовано вибір певних технологій, протоколів зв'язку та платформ для реалізації системи керування. Розкрито деталі розробки програмних модулів, їх функціональність, вибір мов програмування та платформи для реалізації.

У третьому розділі описується процес створення фізичного прототипу розумного будинку та проведено тестування для оцінки ефективності системи.

Структура і обсяг роботи. Робота складається з вступу, трьох розділів, висновків, списку використаних джерел і трьох додатків. Загальний обсяг роботи становить 74 с., 6 рис., 3 дод., 25 джерел.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Аналіз сучасних тенденцій у розвитку розумних домів

В першому підрозділі ми детально розглянемо сучасні тенденції розвитку розумних домів, звернемо увагу на ключові інновації та досягнення, які сприяють становленню цієї технології. Ми визначимо переваги та можливості, що надаються розумними системами у плані енергоефективності, безпеки, зручності управління та підвищення якості життя мешканців.

Сучасний світ висуває все більше вимог до зручності, ефективності та екологічної придатності житла. Розумні дома, в яких інтегровані передові технології та системи керування, відповідають цим вимогам. Досягнення сучасних технологій сприяють становленню розумних систем у побутових будівлях.

Ключові інновації в сфері розробки розумних будинків:

1. Інтернет речей (IoT): підключені до мережі пристрої дозволяють мешканцям керувати всім – від освітлення та опалення до кухонних приладів – через смартфони або голосові помічники. Це створює інтегроване та зручне середовище.

2. Штучний інтелект (ШІ) та машинне навчання: розумні системи можуть навчитися аналізувати звички мешканців та адаптувати режими роботи до їхніх потреб, забезпечуючи оптимальну енергоефективність та зручність.

3. Енергоефективність: розумні дома можуть самостійно регулювати опалення, охолодження та освітлення в залежності від погодних умов та присутності мешканців. Це допомагає знизити витрати на енергію та сприяє екологічності.

4. Системи безпеки: від камер спостереження та датчиків диму до систем виявлення зламів, розумні системи забезпечують високий рівень безпеки, сповіщаючи мешканців про будь-які надзвичайні ситуації.

5. Управління водопостачанням: розумні системи можуть контролювати використання води та оптимізувати полив рослин, сприяючи збереженню ресурсів.

Переваги та можливості розробки розумних будинків:

1. Зручність та комфорт: мешканці можуть віддалено керувати будинком, навіть коли вони далеко. Віддалений контроль освітлення, температури та інших параметрів забезпечує максимальний комфорт.

2. Ефективне використання ресурсів: розумні системи допомагають уникнути непотрібних витрат, регулюючи енергоспоживання та водоспоживання відповідно до потреб.

3. Енергоефективність та зниження витрат: оптимізація енергоспоживання та автоматичне вимкнення пристроїв допомагають знизити рахунки за енергію.

4. Збільшення безпеки: віддзеркалення в реальному часі, датчики виявлення вторгнень та інші заходи підвищують рівень безпеки.

5. Екологічна придатність: завдяки ефективному управлінню ресурсами розумні дома сприяють зниженню негативного впливу на навколишнє середовище.

Ці тенденції та інновації привертають увагу до розумних домів як до майбутнього будівництва, що поєднує ефективність, комфорт та стійкість до викликів сучасного життя.

1.2. Проблеми розумних будинків, аналіз та контекстуалізація

Здійснений аналіз проблем та викликів, з якими стикаються розумні дома. Було розглянуто питання приватності, кібербезпеки, стійкості та взаємодії із

зовнішнім середовищем. Було визначено важливість цих аспектів у контексті розробки системи керування розумним домом.

Розумні будинки, незважаючи на їх вагомні переваги, зіткнулися з декількома важливими проблемами та викликами, які потребують серйозної уваги у процесі розробки та впровадження систем керування. Докладний аналіз цих аспектів є необхідним для досягнення ефективної та стійкої роботи розумних домів.

Приватність та конфіденційність: один з найважливіших аспектів, який викликає обурення серед користувачів розумних будинків – це збір та зберігання особистих даних. Датчики, камери та інші пристрої можуть відслідковувати рухи та звички мешканців. Ця інформація може бути використана не тільки для покращення комфорту, але й для рекламних та аналітичних цілей. Важливо розробити прозорі та чіткі політики збору даних та їхньої обробки, а також забезпечити можливість вимикати деякі функції з метою захисту приватності користувачів.

Кібербезпека та захист від кіберзагроз: розумні будинки є потенційною мішенню для хакерів та зловмисників, оскільки вони підключені до Інтернету та містять велику кількість вразливих точок доступу. Інтеграція IoT-пристроїв та інших технологій збільшує ризик витоку особистих даних, втрати контролю над системою, або навіть може призвести до фізичних загроз. Забезпечення надійного захисту, шифрування даних та використання найсучасніших методів кібербезпеки є критично важливими завданнями для розумних будинків.

Стійкість та надійність: розумні системи вимагають високої стійкості та надійності, оскільки будь-яка помилка або збій може призвести до незручностей, а в деяких випадках – і до реальних небезпек. Недостатня стійкість може спричинити аварійне вимкнення опалення, електрики, або навіть витік води. Важливо розробити системи резервування та відновлення, а також використовувати високоякісне обладнання.

Взаємодія з зовнішнім середовищем: розумні будинки мають взаємодіяти з різними зовнішніми системами та середовищами, такими як міська

інфраструктура, електромережі, телекомунікаційні мережі тощо. Ця інтеграція може вимагати стандартизації та вирішення питань взаємодії між різними виробниками та постачальниками послуг. Недоліки в цій взаємодії можуть призвести до зниження продуктивності систем та навіть до зростання ризику аварій.

Враховуючи ці аспекти, розробка системи керування розумним домом повинна бути не тільки спрямована на технічні інновації, а й на забезпечення високого рівня приватності, безпеки, стійкості та забезпечення взаємодії з зовнішнім середовищем.

Для досягнення ефективності та оптимальної функціональності системи керування розумним домом ми розробимо математичну модель (ММ), яка відображає основні взаємозв'язки між різними системами та параметрами будинку. Наша модель буде базуватися на теорії керованих систем та оптимізаційних методах.

Огляд математичних методів при розробці розумних будинків:

1. Оптимізація ресурсів: для оптимізації енергоспоживання та інших ресурсів в розумному будинку, будуть використовуватись методи лінійного та нелінійного програмування. Ці методи дозволяють знайти оптимальні значення керуючих параметрів системи з урахуванням обмежень та цільової функції [1].

2. Прогнозування: для покращення ефективності системи ми будемо застосовувати методи статистичного прогнозування. Аналіз історичних даних допоможе нам передбачити пікові навантаження на ресурси та оптимізувати їх використання.

3. Теорія черг: у контексті керування розумним будинком, теорія черг може бути застосована для оптимізації послідовності виконання різних завдань та оптимального використання ресурсів.

4. Аналіз даних: методи аналізу даних, зокрема класифікація та кластеризація, допоможуть виявити патерни споживання ресурсів та користувацьких переваг, що може бути використано для покращення системи керування.

5. Системи інтелектуального аналізу: використання штучного інтелекту та машинного навчання дозволить системі розумного дому навчатися та адаптуватися до змінних умов та вимог користувачів.

Цей огляд математичних методів надає нам основу для подальшого аналізу та розробки математичної моделі системи керування розумним домом. У наступних розділах ми детальніше розглянемо конкретні застосування цих методів для вирішення завдань оптимізації, прогнозування та керування в розумному будинку.

Приклад ММ для оптимізації системи опалення

Нехай:

T_{inside} – температура усередині будинку;

$T_{outside}$ – зовнішня температура;

P_{heat} – потужність опалювальної системи;

C_{heat} – теплоємність;

R – опір теплопередачі через стіни та вікна;

S – площа будинку;

K – кількість одиниць тепла, необхідних для підігріву одного кубометра повітря на один градус Цельсія;

Δt – інтервал часу;

$T_{desired}$ – бажана температура всередині приміщення.

Математична модель може бути такою:

$$\Delta T_1 = (T_{outside} - T_{inside}) \cdot R \cdot S \cdot \Delta t / (Q \cdot 3600),$$

де ΔT_1 – зміна температури у будинку протягом інтервалу часу Δt .

Потужність опалювальної системи буде залежати від різниці між бажаною температурою і температурою усередині будинку:

$$P_{heat} = (T_{desired} - T_{inside}) \cdot C_{heat},$$

де C_{heat} – це фізична властивість, яка визначає кількість теплової енергії, необхідної для підвищення температури приміщення на 1 градус Цельсія. C_{heat} вимірюється в джоулях на градус Цельсія ($J/^\circ C$) або кілоджоулях на градус Цельсія ($kJ/^\circ C$).

Завдання оптимізації полягатиме у знаходженні оптимальної потужності опалювальної системи, яка б забезпечила бажану температуру в будинку, мінімізуючи вартість опалення:

$$\text{Мінімізувати } P_{\text{heat}} = (T_{\text{desired}} - T_{\text{inside}}) \cdot C_{\text{heat}}.$$

З обмеженнями:

$$\Delta T_1 = (T_{\text{outside}} - T_{\text{inside}}) \cdot R \cdot S \cdot \Delta t / (Q \cdot 3600),$$

$$T_{\text{inside}} + \Delta T_1 \leq T_{\text{desired}},$$

$$T_{\text{inside}} + \Delta T_1 \geq T_{\text{min}},$$

де T_{min} – мінімальна прийнятна температура у будинку.

Ця математична модель дозволить системі розумного будинку оптимізувати використання опалювальної системи, забезпечуючи комфортні умови для мешканців та ефективно використання ресурсів.

Проведено аналіз літературних джерел та досліджень, де було вирішено подібні задачі. Представимо різноманітні підходи до розв'язання проблем ефективного керування будинком та автоматизації процесів, які були успішно використані у минулому.

Вивчення попередніх досліджень та літературних джерел розкриває ряд підходів та методів, що були успішно використані для розв'язання проблем ефективного керування будинком та автоматизації процесів.

Методи оптимізації ресурсів: у дослідженні [1] розглядається використання генетичних алгоритмів для оптимізації використання електроенергії в розумних домах. Цей підхід дозволяє знизити споживання електроенергії та знижує витрати на електроенергію.

Принцип генетичних алгоритмів полягає в емуляції процесів природного відбору, де кращі адаптовані організми мають більше шансів передати свої гени наступному поколінню. В контексті управління енергоспоживанням розумного будинку, це означає створення та еволюцію популяції "індивідів" – можливих планів споживання електроенергії – до тих пір, поки не буде знайдено оптимальний план.

Конкретні кроки реалізації методу:

1. Початкова популяція: спочатку створюється початковий набір різних планів споживання електроенергії для розумного будинку.

2. Функція пристосованості: кожен план оцінюється за певною функцією пристосованості, яка може включати в себе такі параметри, як вартість електроенергії, рівень комфорту, попередження перевантажень мережі тощо.

3. Схрещування та мутація: плани з найвищою функцією пристосованості вибираються для схрещування та мутації. Це дозволяє створювати нові плани, які можуть бути кращими за попередні.

4. Оцінка та відбір: нові плани оцінюються за функцією пристосованості, і ті з них, які є кращими за попередні, вибираються для наступного покоління.

5. Ітерація: процес схрещування, мутації, оцінки та відбору повторюється кілька разів (ітерацій), доки не буде знайдено план, який задовольняє критерії оптимізації (наприклад, мінімізація витрат на електроенергію).

Переваги використання генетичних алгоритмів:

1. Гнучкість та адаптивність: генетичні алгоритми можуть знаходити оптимальні рішення в складних та змінних середовищах, де інші методи можуть не працювати ефективно.

2. Можливість врахування багатьох факторів: генетичні алгоритми дозволяють моделювати багатофакторні проблеми, де рішення залежить від багатьох змінних.

3. Ефективність: генетичні алгоритми можуть працювати швидко та ефективно, здатні знаходити прийнятні рішення навіть у великих просторах можливих рішень.

Висновок:

Використання генетичних алгоритмів для оптимізації використання електроенергії у розумних будинках є перспективним підходом, який дозволяє знизити витрати та покращити енергоефективність. Проте, при реалізації такого підходу слід враховувати складність обчислень, можливість зациклювання на

підоптимальних рішеннях та потребу відповідних алгоритмічних підходів для ефективної реалізації генетичних алгоритмів.

Аналіз даних для прогнозування: у статті [2] розглядається використання аналізу даних для передбачення витрат ресурсів та потенційних поломок обладнання. Цей підхід допомагає планувати обслуговування та зменшує негативний вплив від збоїв.

Аналіз даних для прогнозування витрат ресурсів: один із ключових аспектів використання аналізу даних у розумних будинках – це передбачення витрат ресурсів, таких як електроенергія, вода та інші. Шляхом аналізу даних, зібраних від датчиків та систем керування, можна створити модель, яка дозволяє передбачати пікові та загальні споживання ресурсів. Це допомагає користувачам оптимізувати своє споживання, планувати часи пікових навантажень, а також зменшує витрати на ресурси.

Прогнозування потенційних несправностей: ще однією важливою областю є передбачення потенційних поломок обладнання. Зібрані дані про роботу різних систем у розумному будинку можуть допомогти розробити моделі, які ідентифікують паттерни перед поломкою. Наприклад, аномальні зміни в роботі систем опалення чи кондиціонування можуть служити сигналами можливої поломки. Це дозволяє здійснювати передбачувальний ремонт або обслуговування, запобігаючи значним збоєм та зберігаючи оптимальний рівень функціональності розумного будинку.

Планування обслуговування та підвищення ефективності: застосування аналізу даних для передбачення можливих збоїв та витрат ресурсів дозволяє розумному будинку розробляти оптимальні плани обслуговування. Відповідно до прогнозів, можна планувати ремонтні роботи або заміну обладнання на більш зручний час, що зменшує негативний вплив на комфорт мешканців.

Переваги використання аналізу даних для прогнозування:

1. Попередження збоїв: аналіз даних дозволяє виявити аномалії та паттерни, що передують поломкам, що сприяє попередженню серйозних збоїв.

2. Оптимізація ресурсів: прогнозування витрат дозволяє ефективно використовувати ресурси та знижувати витрати.

3. Покращення зручності: вчасне обслуговування та управління збоями сприяють збереженню комфорту мешканців.

Аналіз даних для прогнозування в розумних будинках є важливим інструментом, який допомагає покращити управління, підвищити ефективність та знизити негативний вплив від можливих збоїв. Враховуючи цей підхід у розробці системи керування розумним будинком, можна досягти більш надійної та стійкої роботи системи.

Інтелектуальні агенти та системи керування: у дослідженні [3] описується застосування інтелектуальних агентів для управління різними аспектами розумного будинку. Цей підхід дозволяє створювати гнучкі та адаптивні системи керування.

Основні аспекти дослідження при розробці розумного будинку:

1. Інтелектуальні агенти: у розумних будинках різні пристрої та системи можуть бути розглянуті як інтелектуальні агенти, які мають здатність взаємодіяти та приймати рішення на основі зібраних даних та заздалегідь визначених правил. Це можуть бути агенти, відповідальні за опалення, освітлення, безпеку, розподіл ресурсів.

2. Взаємодія та координація: інтелектуальні агенти взаємодіють один з одним та з мешканцями розумного будинку, обмінюючись даними та координуючи свою діяльність. Наприклад, агент відповідальний за освітлення може враховувати наявність людей у кімнатах та погодні умови для оптимального регулювання освітлення.

3. Адаптація до змінних умов: один із ключових аспектів інтелектуальних агентів – це їх здатність адаптуватись до змінних умов та нової інформації. Наприклад, якщо погодні умови змінилися, агент, що відповідає за кондиціонування, може автоматично змінити налаштування для ефективнішого споживання енергії.

Потенційні переваги інтелектуальних агентів для управління:

1. Гнучкість та адаптивність: система інтелектуальних агентів може бути легко налаштована та адаптована під потреби конкретного будинку та мешканців.
2. Автономність: інтелектуальні агенти можуть приймати рішення без прямого втручання користувачів, забезпечуючи зручність та ефективність.
3. Висока відповідальність: кожен агент відповідає за конкретний аспект, що дозволяє досягти більшої відповідальності та оптимальності управління.

Дослідження [3] розкриває потенціал застосування інтелектуальних агентів для створення гнучких та адаптивних систем керування розумним будинком. Цей підхід відкриває шлях до створення більш розумних та ефективних систем, які можуть самостійно приймати рішення та адаптуватись до змінних умов.

Використання датчиків та IoT: в дослідженні [4] проведено огляд використання IoT для створення розумних будинків. Датчики вимірюють параметри середовища та передають дані до центральної системи для прийняття рішень щодо управління.

Огляд використання IoT для розумних будинків: ця студія фокусується на ідеї створення розумного середовища в будинках, за допомогою якого датчики та пристрої можуть взаємодіяти між собою та з мешканцями, створюючи сприятливі та ефективні умови проживання. Основний принцип полягає у використанні різноманітних датчиків для збору інформації про стан різних систем та параметри середовища в реальному часі.

Роль датчиків та передача даних: датчики, розташовані у різних частинах розумного будинку, вимірюють різні параметри, такі як температура, вологість, освітленість, рух тощо. Ці дані збираються та передаються до центральної системи, яка аналізує їх та вживає відповідних заходів для управління будинком.

Принципи роботи IoT-заснованого розумного будинку:

1. Збір даних: датчики реєструють навколишні умови та стан різних систем в будинку, таких як опалення, освітлення, безпека.
2. Передача даних: зібрані дані передаються через мережу до центральної системи, де вони обробляються та аналізуються.
3. Аналіз та взаємодія: центральна система використовує алгоритми аналізу для зрозуміння поточного стану та прийняття рішень. Наприклад, при засіванні низької температури, система може автоматично підняти рівень опалення.
4. Управління: за допомогою мобільних додатків або інших інтерфейсів, користувачі можуть взаємодіяти з системою, давати вказівки та контролювати різні параметри будинку.

Важливість та переваги використання IoT для розумних будинків:

1. Зручність: мешканці можуть віддалено контролювати різні аспекти будинку за допомогою смартфонів або планшетів.
2. Ефективність ресурсів: автоматичне регулювання освітлення, опалення та інших систем дозволяє ефективно використовувати енергію та ресурси.
3. Безпека: системи безпеки можуть відслідковувати незвичайну активність та сповіщати мешканців про можливі загрози.

Дослідження [4] відкриває світ IoT-технологій для розумних будинків, де датчики та системи взаємодіють для забезпечення комфорту, ефективності та безпеки. Це дослідження розкриває потенціал IoT для перетворення будинків в інтелектуальні та адаптивні середовища для мешканців.

Енергоефективність та відновлювальні джерела енергії: у статті [5] досліджується використання відновлювальних джерел енергії та перенесення навантаження для забезпечення сталого та ефективного споживання енергії.

Використання відновлювальних джерел енергії: у статті розглядається принцип використання відновлювальних джерел енергії, таких як сонячні панелі та вітрові турбіни, для виробництва електроенергії. Це дозволяє розумним

будинкам отримувати енергію з природних джерел, зменшуючи залежність від традиційних, необхідність у зовнішніх джерелах та викиди в атмосферу.

Перенесення навантаження: стаття вивчає концепцію перенесення навантаження як стратегії для ефективного використання електроенергії. Це означає, що споживачі можуть перенести використання певних електричних пристроїв на періоди з низьким споживанням електроенергії, наприклад, ночі. Це дозволяє розподіляти навантаження більш рівномірно протягом доби, зменшуючи пікове навантаження та споживання.

Важливість дослідження використання відновлювальних джерел енергії та перенесення навантаження для забезпечення сталого та ефективного споживання енергії:

1. Сталість: використання відновлювальних джерел допомагає забезпечити стале споживання енергії та зменшити екологічний вплив.

2. Ефективність: перенесення навантаження допомагає уникнути пікового споживання, зменшуючи витрати та навантаження на енергетичну інфраструктуру.

3. Економія: використання відновлювальних джерел та стратегій перенесення навантаження може знизити витрати на електроенергію.

Стаття [5] пропонує цінний внесок у розуміння та розв'язання важливої проблеми управління енергетичними ресурсами в розумних будинках. Використання відновлювальних джерел та стратегій перенесення навантаження допомагає покращити сталість, ефективність та екологічність споживання енергії, сприяючи розвитку екологічно чистих та стало розвиваючихся розумних домів.

Ці дослідження та літературні джерела вказують на великий потенціал різноманітних підходів до розв'язання проблем розумних домів. Ефективне керування ресурсами, використання аналізу даних для передбачення та прийняття рішень, застосування інтелектуальних систем та використання IoT дозволяють досягти високої ефективності та зручності управління будинком. Врахування питань енергоефективності, кібербезпеки та взаємодії з зовнішнім

середовищем стає критично важливим для забезпечення надійності та стійкості розумної системи керування будинком.

Огляд структури роботи. У цьому підрозділі буде надано огляд основних джерел та літературних джерел, що використовуються для розробки та підтримки дослідницької роботи.

По завершенні аналізу предметної галузі та постановки задачі, переходимо до другого розділу, де розглядаються методи та підходи, що використовуються в розробці проекту "Розумний будинок, керування елементами будинку".

Підтримка та розвиток кваліфікаційної роботи базується на широкому спектрі джерел, що включають наукові статті, книги, технічні документи. Основна увага акцентується на сучасних підходах та інноваціях у сфері розробки розумних домів, їхнього управління та оптимізації ресурсів.

Наукові статті та журнали:

1. У статті [1] досліджується використання генетичних алгоритмів для оптимізації споживання електроенергії в розумних домах. Вона надає інсайти щодо зниження витрат та збільшення ефективності ресурсів.

2. У статті [2] розглядається використання аналізу даних для передбачення витрат ресурсів та поломок обладнання. Вона наголошує на важливості використання прогнозування для оптимізації функціонування розумних домів.

3. У статті [3] вивчається використання інтелектуальних агентів для управління різними аспектами розумного будинку. Вона розкриває переваги гнучких та адаптивних систем керування.

4. Дослідження [4] оглядає використання Інтернету Речей для створення розумних домів. Воно розкриває переваги використання IoT для ефективного управління ресурсами та забезпечення комфорту мешканців.

5. У статті [5] розглядається використання відновлювальних джерел енергії та перенесення навантаження для ефективного споживання енергії в

розумних домах. Вона надає інформацію про важливість сталого управління енергетичними ресурсами.

Книги та монографії:

1. У книзі [6] надається широкий огляд розумних домів та їхнього впливу на підвищення якості життя та ефективного управління ресурсами.
2. У книзі [7] розглядаються основні принципи та практичні аспекти автоматизації розумних домів, включаючи використання технологій IoT та інших інноваційних рішень.

Технічні документи та ресурси:

1. Офіційний веб-сайт стандарту Zigbee [8], що використовується для безпроводного зв'язку між різними пристроями у розумних домах.
2. Ресурс [9] для стандарту KNX, що розробляється для забезпечення стандартизованого зв'язку та управління між різними системами та пристроями у будинках.
3. Електронна бібліотека [10] Інституту інженерів електротехніки та електроніки, де можна знайти багато наукових статей та досліджень з розумних домів та їхнього управління.

1.3. Постановка задачі

Метою даного проекту є розробка інтегрованої системи керування елементами будинку, що базується на IoT та інноваційних технологіях, за допомогою застосунку для смартфона (рис. 1.1). Ця система повинна базуватись на використанні мікроконтролера Arduino як ключового компонента для забезпечення функціональності та взаємодії різних підсистем.

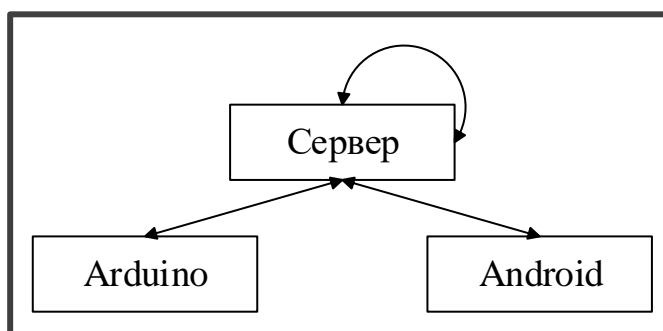


Рис. 1.1. Загальна схема роботи інформаційної системи

У результаті розробки системи "Розумний будинок", повинні задовольнятися такі умови:

1. Сучасні технології: розробка розумного дому базується на сучасних технологіях, таких як IoT, мікроконтролери, датчики та мобільні додатки. Ці технології сприяють створенню ефективних та зручних систем керування.

2. Безпека і приватність: забезпечення інформаційної безпеки та захисту приватності користувачів є критично важливим завданням при розробці системи розумного дому. Для цього необхідно дотримуватися найвищих стандартів безпеки та враховувати можливі ризики.

3. Сумісність і інтеграція: система повинна бути сумісною з різними пристроями та компонентами, що використовуються в розумному будинку. Інтеграція з різними платформами та протоколами також є ключовим аспектом.

4. Енергоефективність: забезпечення енергоефективності системи є важливим для зменшення витрат електроенергії при здійсненні опалення та підвищення сталості роботи.

5. Серверна інфраструктура: велика увага має бути приділена налагодженню та обслуговуванню серверної частини системи, зокрема, забезпеченню додаткового простору на диску, потужності ЦП та оперативної пам'яті.

6. Масштабованість і майбутнє розвитку: система має бути масштабованою та готовою до росту, оскільки можливості розумного дому

постійно розширюються. Додавання нових пристроїв та функцій має бути простим та зручним процесом.

Вимоги до інформаційної безпеки в системі "Розумний будинок" є дуже важливим аспектом, оскільки ця система здатна обробляти чутливі дані та взаємодіяти з фізичними системами будинку. Забезпечення безпеки має включати наступні вимоги:

1. Автентифікація та авторизація: кожен користувач має бути ідентифікованим і авторизованим перед отриманням доступу до системи.
2. Встановлення різних рівнів доступу для різних користувачів залежно від їхніх прав та ролей.
3. Захист даних: всі дані, які передаються між апаратними та програмними компонентами системи, мають бути зашифрованими.
4. Дані, збережені на сервері, повинні бути захищені від несанкціонованого доступу.
5. Відслідковування та аудит: реєстрація всіх дій користувачів у системі для виявлення та аналізу можливих вторгнень або порушень безпеки.
6. Захист від вторгнень: встановлення заходів захисту, таких як брандмауери, системи виявлення вторгнень та оновлення програмного забезпечення для запобігання вторгненням.
7. Резервне копіювання та відновлення: регулярне резервне копіювання всіх даних та програмних компонентів системи для можливості відновлення в разі втрати даних або інших негараздів.
8. Фізична безпека: захист апаратної частини системи від фізичного доступу, зокрема від несанкціонованого доступу до мікроконтролера та інших пристроїв.
9. Захист від вірусів та зловмисного програмного забезпечення: використання антивірусного програмного забезпечення та систем виявлення та усунення загроз для виявлення та блокування потенційно шкідливих програм.
10. Навчання та освіта користувачів: надання інструкцій користувачам щодо безпечного використання системи та захисту своїх облікових записів.

11. Апаратна безпека: захист фізичної апаратної частини системи від несанкціонованого доступу або фізичних пошкоджень.

Ці вимоги допоможуть забезпечити безпеку інформації та функціонування системи "Розумний будинок", зменшити ризик можливих загроз і зберегти приватність користувачів.

Завданням даного проєкту є створення повноцінної автоматизованої інформаційної системи "Розумний будинок", базованої на мікроконтролері Arduino та ряді компонентів та програмних рішень. Проєкт повинен включати апаратну частину, серверну частину, сенсори та актуатори, програмну частину та мобільний додаток.

Апаратна частина складається з мікроконтролеру Arduino: на ньому буде розроблена програма керування, яка буде відповідати за збір даних від сенсорів, взаємодію з актуаторами та управління різними системами у розумному будинку.

Сенсори та актуатори передбачають установку датчиків, які будуть моніторити параметри оточуючого середовища, такі як температура, вологість, освітлення, рух, а також пристрої для виконання команд, такі як включення та вимикання освітлення, обігріву тощо.

Серверна частина повинна складатись з веб-серверу. Буде розроблений та запущений веб-сервер для можливості віддаленого керування та моніторингу системи "Розумний будинок". Користувачі матимуть змогу взаємодіяти з системою через веб-інтерфейс.

База даних необхідна для зберігання даних про стан системи та їх історію. Це допоможе у вивченні та аналізі роботи системи з плином часу.

Мобільний додаток передбачає розробку мобільного додатку для смартфона (Android), який дає можливість користувачам взаємодіяти з системою у режимі реального часу навіть поза межами будинку.

Програмна частина повинна складатись з мови програмування та технології: використання мов програмування, таких як, Python, JavaScript (JS) для розробки програм для мікроконтролера та веб-інтерфейсу.

IoT протоколи передбачають застосування стандартних IoT протоколів, таких як MQTT для забезпечення комунікації між мікроконтролером та серверною частиною системи, а також HTTP для взаємодії з веб-інтерфейсом.

Кінцевий результат повинен мати розроблену та впроваджену повноцінну інформаційну систему "Розумний будинок", яка здатна забезпечувати функціональність моніторингу, керування та оптимізації різними системами розумного будинку.

Система повинна здатна збирати дані від сенсорів, аналізувати їх та надавати користувачам можливість взаємодіяти з будинком через мобільні додатки або веб-інтерфейс.

Оптимізація споживання енергії, підвищення безпеки, зручності та комфорту мешканців будинку буде досягнута завдяки аналізу та впровадженню інтелектуальних стратегій керування ресурсами.

Таким чином, розробка цієї системи передбачає інтеграцію апаратної та програмної частин, використання сучасних мов програмування та технологій, а також забезпечення зручного та ефективного управління будинком для користувачів (рис. 1.2).

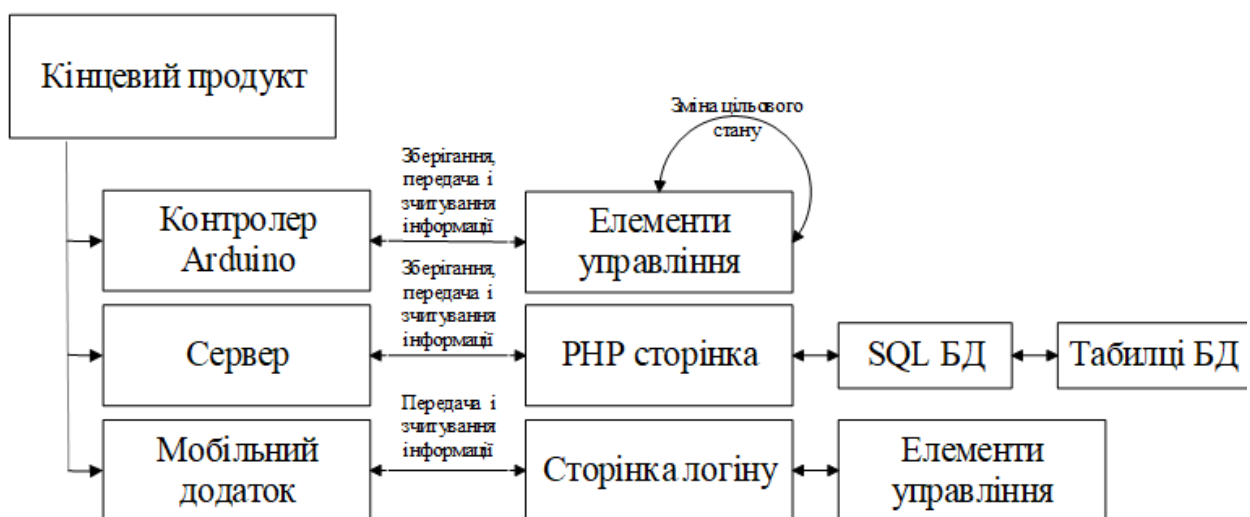


Рис. 1.2. Схема кінцевого результату

Для оптимальної працездатності мобільного додатку необхідно, щоб платформа працювала на операційній системі Android версії 7.0 або вище та мала доступ до мережі Інтернет.

Програмне забезпечення, призначене для контролера, повинно бути розроблене спеціально для мікроконтролера Arduino MEGA 2560, який використовує Ethernet Shield W5100 Arduino для забезпечення веб-серверної функціональності. Для ефективної роботи серверу необхідні наступні ресурси:

1. Понад 150 гігабайтів вільного місця на накопичувачі SSD.
2. Центральний процесор (ЦП) не нижче класу Pentium 4.
3. Відеопам'ять обсягом не менше 512 мегабайт.
4. Мінімум 4 гігабайти оперативної пам'яті (RAM).
5. Доступ до мережі Інтернет зі швидкістю не менше 10 MB/s.

Ці вимоги є необхідними для забезпечення стабільної та ефективної роботи серверу та його взаємодії з контролером і мобільним додатком.

Вимоги до інформаційної та програмної сумісності. Вимоги до інформаційної та програмної сумісності в системі "Розумний будинок" повинні передбачати:

1. Сумісність інтерфейсів: інформаційні системи повинні використовувати стандартизовані інтерфейси та протоколи зв'язку, щоб забезпечити можливість взаємодії між компонентами системи.

2. Сумісність версій програмного забезпечення: усі програмні компоненти, включаючи мобільний додаток, серверну частину і мікроконтролер, мають бути сумісними за версіями. Оновлення повинні бути ретельно перевірені та впроваджені, щоб уникнути конфліктів і сумісності.

3. Доступ до бази даних: система повинна мати доступ до спільної бази даних, яка забезпечує централізоване зберігання та обмін даними між компонентами системи.

4. Сумісність між пристроями: програмне забезпечення має підтримувати широкий спектр пристроїв, таких як сенсори різних виробників і моделей, щоб забезпечити їхню безпроблемну інтеграцію в систему.

5. Мобільна сумісність: мобільний додаток повинен бути розроблений для платформи, яку використовують користувачі (Android, не нижче 7.0) і бути сумісним з різними версіями операційних систем.

6. Забезпечення даними: інформаційні дані повинні зберігатися та передаватися в безпечному форматі з врахуванням стандартів шифрування та захисту даних.

7. Масштабованість: система повинна бути здатною масштабуватися і приймати нові компоненти без необхідності ревізії всієї інфраструктури.

Забезпечення інформаційної та програмної сумісності відіграє важливу роль у забезпеченні прозорої та ефективної роботи системи "Розумний будинок" і забезпечує її готовність до майбутніх змін і розширень.

В цілому, розробка системи "розумний дім" – це важлива та перспективна галузь, яка може значно полегшити життя мешканців та сприяти ефективному розрахунку та використанню ресурсів. Врахування всіх зазначених вимог та аспектів є ключовим для успішної реалізації такої системи.

1.4. Висновки

У розділі було проведено детальний огляд сучасного стану систем опалення в розумних будинках та визначено ключові аспекти, що впливають на їх ефективність. На підставі цього аналізу були визначені основні виклики та завдання, які вирішує розроблюване програмне забезпечення.

На основі аналізу було сформульовано постановку задачі для розробки програмного забезпечення:

1. Розробка адаптивного алгоритму: розробити алгоритм, який адаптивно реагує на зміни в умовах та забезпечує оптимальне опалення.

2. Інтеграція з іншими системами: забезпечити можливість інтеграції з іншими розумними системами в будинку для синхронізації та координації дій.

3. Оптимізація витрат ресурсів: розробити алгоритми для оптимізації витрат енергії та інших ресурсів при забезпеченні комфортних умов.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Загальна схема керуванням розумним будинком

Розроблена інформаційна система, складається з таких складових: керуючий контролер, який базується на мікропроцесорі Arduino, серверна частина системи та мобільний додаток, призначений для платформи операційної системи (ОС) Android версії 7.0 і вище. Показано на рис. 1.2.

Функціональне призначення програми "Розробка та дослідження ефективності впровадження програмного забезпечення розрахунку системи опалення в проєкті розумний будинок" полягає в забезпеченні зручного та ефективного управління різними системами та пристроями в будинку за допомогою мобільного додатку. Основні функції цієї програми включають в себе:

1. Управління освітленням: користувач може вмикати та вимикати світло в різних частинах будинку або регулювати яскравість за допомогою мобільного додатку.

2. Контроль опалення та кондиціонування повітря: додаток дозволяє користувачеві встановлювати та регулювати температуру в приміщеннях, управляти системами опалення та кондиціонування.

3. Безпека і захист: програма надає можливість включати та вимикати систему безпеки, таку як відеоспостереження або сигналізація. Користувач може отримувати повідомлення про події, пов'язані з безпекою будинку.

4. Енергозбереження: додаток надає можливість моніторити та керувати використанням електроенергії в будинку, допомагаючи ефективно використовувати ресурси та знижувати витрати.

5. Розклад та автоматизація: користувач може налаштовувати розклади для різних функцій, таких як вмикання світла або опалення, а також створювати сценарії автоматизації для реагування на певні події або умови.

6. Віддалений доступ: користувач може віддалено керувати елементами будинку через Інтернет з будь-якого місця за допомогою мобільного додатку.

7. Моніторинг та статистика: програма забезпечує можливість моніторити використання ресурсів, а також надає статистику щодо споживання електроенергії та інших параметрів будинку.

Функціональне призначення програми полягає в забезпеченні комфорту, безпеки та ефективного управління різними аспектами життя в будинку через мобільний додаток.

2.2. Математичні моделі керування системами опалення

У кваліфікаційній роботі використовуються наступні математичні методи:

1. Моделювання системи: для створення системи "Розумний будинок" була використана математична модель, яка відображає взаємозв'язки між різними пристроями та параметрами будинку. Ця модель дозволила розробити логіку управління та автоматизації системи.

2. Аналіз даних для прогнозування: у статті [2] був застосований аналіз даних для передбачення витрат ресурсів та можливих поломок обладнання. Цей метод допомагає планувати обслуговування та зменшувати витрати на ремонт.

3. Оптимізація ресурсів: у дослідженні [1] використовуються генетичні алгоритми для оптимізації використання електроенергії при здійсненні опалення в розумних домах. Цей метод допомагає знизити споживання енергії та знизити витрати.

4. Використання інтелектуальних агентів: у дослідженні [3] описується застосування інтелектуальних агентів для управління різними аспектами розумного будинку. Цей метод дозволяє створювати гнучкі та адаптивні системи керування.

Ці математичні методи були використані для покращення функціональності та ефективності системи "Розумний будинок", забезпечуючи оптимальне використання ресурсів при здійсненні опалення та зручне управління будинком.

2.3. Опис використаних технологій та мов програмування

У кваліфікаційній роботі були використані різні технології та мови програмування для реалізації різних компонентів системи "Розумний будинок":

1. Мікропроцесор Arduino: для контролю різних елементів будинку, таких як світло, опалення та інші, було використано мікроконтролер Arduino. Arduino була вибрана через свою доступність, простоту програмування та можливість підключення до різних сенсорів та пристроїв.

2. JS та React Native: для створення мобільного додатку для Android було використано JavaScript та фреймворк React Native. JavaScript є популярною мовою для розробки мобільних додатків, а React Native дозволяє створювати кросплатформні додатки для Android та iOS з використанням одного коду.

3. HTML, CSS та JS (для веб-інтерфейсу): для створення веб-інтерфейсу для взаємодії з системою були використані стандартні веб-технології, такі як HTML для структури сторінки, CSS для стилізації та JS для взаємодії з сервером Arduino через HTTP-запити.

4. Node.js та Express.js (для серверної частини): для створення серверної частини системи було використано Node.js разом з фреймворком Express.js. Node.js дозволяє створювати ефективні та швидкодіючі сервери, а Express.js спрощує розробку серверного застосунку.

5. Axios (для HTTP-запитів): для взаємодії між мобільним додатком та сервером на Arduino була використана бібліотека Axios для виконання HTTP-запитів.

Ці технології та мови програмування були вибрані з метою створення функціональної та ефективної системи "Розумний будинок" з інтуїтивно зрозумілим інтерфейсом користувача.

2.4. Розробка мобільного застосунку для платформи Android

Мова програмування Java була вибрана для написання цього коду через кілька ключових переваг, особливо коли мова йдеться про розробку мобільних додатків для платформи Android:

1. Офіційна мова Android: JS є офіційною мовою розробки для Android, підтримуваною і рекомендованою компанією Google, яка стоїть за розробкою Android. Це означає, що вона має велику підтримку, обширну документацію та багато інструментів для розробки Android додатків.

2. Кросплатформеність: JS є кросплатформеною мовою, що означає, що ви можете розробляти додатки для Android і інших платформ, таких як Windows, macOS і Linux, використовуючи одну й ту ж кодову базу.

3. Безпека: JS відома своєю високою безпекою. Вона використовує систему керування пам'яттю та має вбудовану підтримку для обробки винятків, що допомагає уникнути численних типових помилок.

4. Широкі можливості бібліотек: для розробки під Android доступно велике число бібліотек та фреймворків, написаних на Java, що спрощує роботу з різними аспектами розробки додатків, включаючи мережеву взаємодію, роботу з базами даних, графічний інтерфейс і багато інших.

5. Велика спільнота розробників: Java має велику та активну спільноту розробників. Це означає, що ви можете легко знайти відповіді на свої питання, довідку та приклади коду в Інтернеті.

Загалом, JS є надійним і добре підтримуваним вибором для розробки Android-додатків, і вона допомагає забезпечити високу продуктивність і безпеку вашого додатку.

Програмний код для реалізації системи керування розумним будинком для додатку Adroid:

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class SmartHomeControllerActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_smart_home_control);
        // Ініціалізація кнопок для керування системою
        Button turnOnLightsButton = findViewById(R.id.btnTurnOnLights);
        Button turnOffLightsButton = findViewById(R.id.btnTurnOffLights);
        Button adjustTemperatureButton =
        findViewById(R.id.btnAdjustTemperature);
        // Додавання обробників подій для кнопок
        turnOnLightsButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Логіка для увімкнення світла
                Toast.makeText(SmartHomeControllerActivity.this, "Світло увімкнено",
                Toast.LENGTH_SHORT).show();
            }
        });
        turnOffLightsButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Логіка для вимкнення світла
```

```

        Toast.makeText(SmartHomeControlActivity.this, "Світло вимкнено",
Toast.LENGTH_SHORT).show();
    }
});
adjustTemperatureButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Логіка для налаштування температури
        Toast.makeText(SmartHomeControlActivity.this, "Температура
налаштована", Toast.LENGTH_SHORT).show();
    }
});
}
}

```

Це шаблон, який містить три кнопки для управління світлом та температурою. При кліку на кожну кнопку виводиться спливаюче повідомлення.

Додавання безпеки до мобільного додатку для керування системою розумного будинку є дуже важливим завданням. Ось кілька основних практик безпеки [12]:

1. Аутентифікація та авторизація: для доступу до функцій додатку користувач повинен ввести ім'я користувача та пароль. Після аутентифікації слід перевірити, чи має користувач доступ до певних функцій (авторизація).
2. Захист даних: для передачі конфіденційних даних (наприклад, ім'я користувача та пароль) слід використовувати шифрування. Необхідно використовувати HTTPS для захисту з'єднання.
3. Захист від атак: важливо перевіряти дані, які вводяться користувачем, щоб запобігти атакам, таким як SQL-ін'єкція та внесення коду. Використати параметризовані запити до бази даних.

4. Віддалене відключення: забезпечення можливості віддаленого відключення доступу до системи в разі втрати смартфона або вразливості акаунта.

5. Логування та моніторинг: вести журнал подій для реєстрації активності користувачів та відслідковувати незвичайну поведінку.

6. Оновлення безпеки: регулярно оновлювати додаток, щоб виправити виявлені вразливості.

7. Обмеження доступу: дати користувачам лише необхідний доступ до функцій системи, обмежити їх права.

Приклад аутентифікації за допомогою SharedPreferences та простого паролю:

```
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class SmartHomeControlActivity extends AppCompatActivity {
    // Метод для перевірки аутентифікації користувача
    private boolean isUserAuthenticated(String username, String password) {
        SharedPreferences preferences = getSharedPreferences("users",
MODE_PRIVATE);
        String savedPassword = preferences.getString(username, "");
        return savedPassword.equals(password);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

setContentView(R.layout.activity_smart_home_control);
// Ініціалізація кнопок для керування системою
Button turnOnLightsButton = findViewById(R.id.btnTurnOnLights);
Button turnOffLightsButton = findViewById(R.id.btnTurnOffLights);
Button adjustTemperatureButton =
findViewById(R.id.btnAdjustTemperature);
// Додавання обробників подій для кнопок
turnOnLightsButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Логіка для увімкнення світла
Toast.makeText(SmartHomeControlActivity.this, "Світло увімкнено",
Toast.LENGTH_SHORT).show();
}
});
turnOffLightsButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Логіка для вимкнення світла
Toast.makeText(SmartHomeControlActivity.this, "Світло вимкнено",
Toast.LENGTH_SHORT).show();
}
});
adjustTemperatureButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Логіка для налаштування температури
Toast.makeText(SmartHomeControlActivity.this, "Температура
налаштована", Toast.LENGTH_SHORT).show();
}
}

```

```

});
// Додавання обробників подій для аутентифікації
Button loginButton = findViewById(R.id.btnLogin);
loginButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        EditText usernameEditText = findViewById(R.id.editTextUsername);
        EditText passwordEditText = findViewById(R.id.editTextPassword);
        String username = usernameEditText.getText().toString();
        String password = passwordEditText.getText().toString();
        if (isUserAuthenticated(username, password)) {
            // Аутентифікація успішна, дозвіл на керування системою
            turnOnLightsButton.setEnabled(true);
            turnOffLightsButton.setEnabled(true);
            adjustTemperatureButton.setEnabled(true);
        } else {
            // Помилка аутентифікації, виведення повідомлення
            Toast.makeText(SmartHomeControlActivity.this, "Неправильне ім'я користувача або пароль", Toast.LENGTH_SHORT).show();
        }
    }
});
}
}

```

Цей код включає аутентифікацію, а також можливість управління системою розумного будинку лише після успішної аутентифікації.

Моніторинг може бути реалізований за допомогою журналювання подій та відслідковування активності користувачів [13]. Додатковий код, який включає журналювання та виведення активності в консоль:

```
import android.content.SharedPreferences;
```



```

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class SmartHomeControlActivity extends AppCompatActivity {
private static final String TAG = "SmartHomeControlActivity";
// Метод для перевірки аутентифікації користувача
private boolean isUserAuthenticated(String username, String password) {
    SharedPreferences preferences = getSharedPreferences("users",
MODE_PRIVATE);
    String savedPassword = preferences.getString(username, "");
    return savedPassword.equals(password);
}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_smart_home_control);
    // Ініціалізація кнопок для керування системою
    Button turnOnLightsButton = findViewById(R.id.btnTurnOnLights);
    Button turnOffLightsButton = findViewById(R.id.btnTurnOffLights);
    Button adjustTemperatureButton =
findViewById(R.id.btnAdjustTemperature);

    // Додавання обробників подій для кнопок
    turnOnLightsButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {

```

```

// Логіка для увімкнення світла
Log.i(TAG, "Користувач увімкнув світло");
Toast.makeText(SmartHomeControlActivity.this, "Світло увімкнено",
Toast.LENGTH_SHORT).show();
}
});
turnOffLightsButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Логіка для вимкнення світла
Log.i(TAG, "Користувач вимкнув світло");
Toast.makeText(SmartHomeControlActivity.this, "Світло вимкнено",
Toast.LENGTH_SHORT).show();
}
});
adjustTemperatureButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Логіка для налаштування температури
Log.i(TAG, "Користувач налаштував температуру");
Toast.makeText(SmartHomeControlActivity.this, "Температура
налаштована", Toast.LENGTH_SHORT).show();
}
});
// Додавання обробників подій для аутентифікації
Button loginButton = findViewById(R.id.btnLogin);
loginButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
EditText usernameEditText = findViewById(R.id.editTextUsername);

```

```

EditText passwordEditText = findViewById(R.id.editTextPassword);
String username = usernameEditText.getText().toString();
String password = passwordEditText.getText().toString();
if (isUserAuthenticated(username, password)) {
    // Аутентифікація успішна, дозвіл на керування системою
    turnOnLightsButton.setEnabled(true);
    turnOffLightsButton.setEnabled(true);
    adjustTemperatureButton.setEnabled(true);
    Log.i(TAG, "Користувач увійшов в систему");
} else {
    // Помилка аутентифікації, виведення повідомлення
    Toast.makeText(SmartHomeControlActivity.this, "Неправильне ім'я
користувача або пароль", Toast.LENGTH_SHORT).show();
    Log.w(TAG, "Неуспішна спроба входу в систему");
}
});
}
}

```

У цьому коді додано логування подій за допомогою Log, яке записує події у консоль Android [13]. Це допомагає відстежувати активність користувачів та події в додатку для подальшого аналізу і моніторингу.

2.5. Мікросхема Arduino MEGA 2560 rev3

На рис 2.1. показано повну схему виводів, пінів мікросхеми Arduino MEGA 2560 rev3.

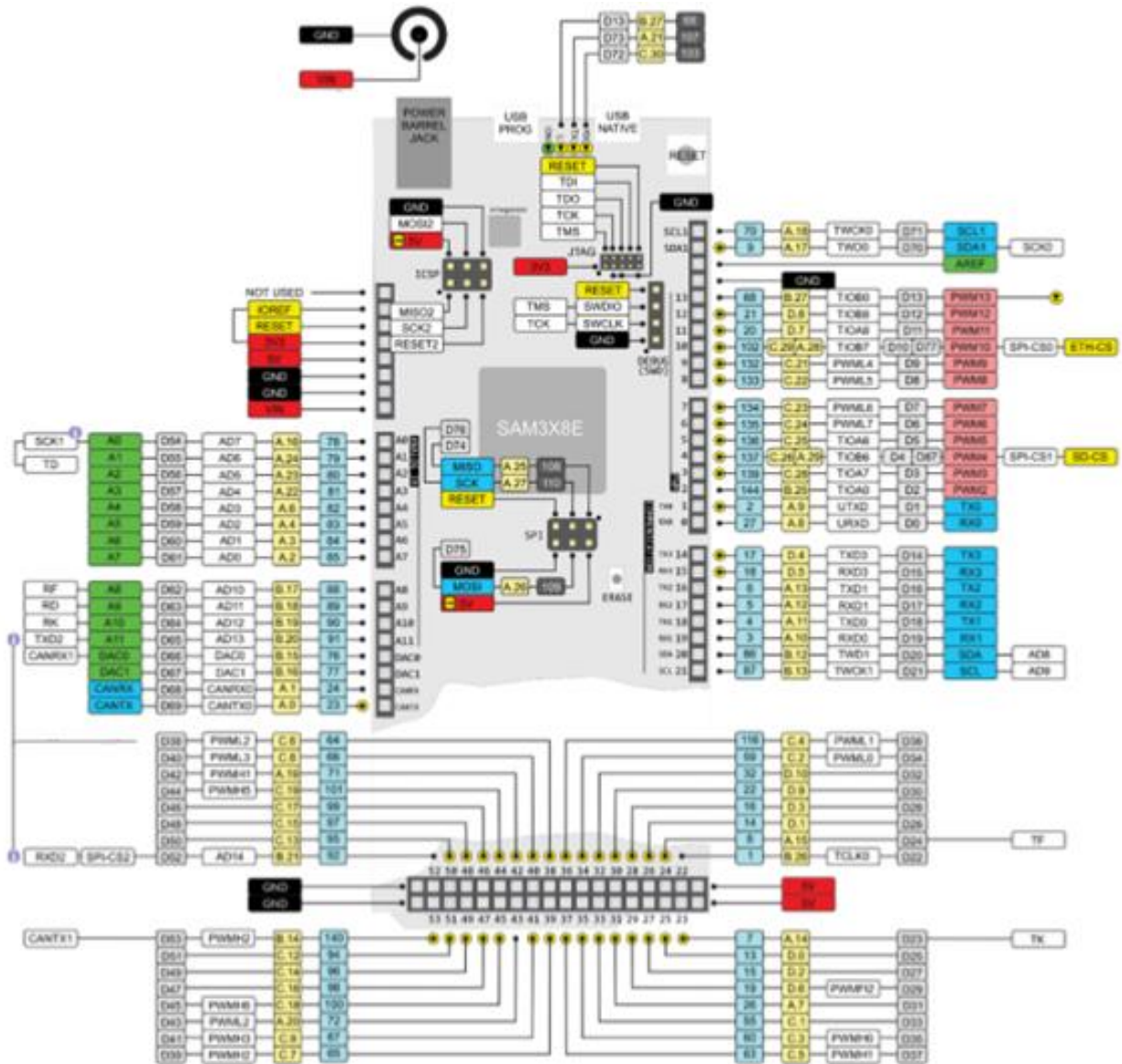


Рис 2.1. Повна схема виводів, пінів мікросхеми.

Робочий алгоритм програми мікроконтролера має просту структуру та ділиться на кілька ключових етапів:

1. Ініціалізація налаштувань: по запуску мікроконтролера програма встановлює підключення до локальної мережі і реєструє веб-сервер Ethernet

Shield W5100. Також задаються інші необхідні налаштування, такі як IP-адреса та MAC-адреса.

2. Ініціалізація змінних: один раз виконується ініціалізація глобальних змінних, які використовуються для керування елементами розумного будинку. Тут також визначаються змінні для таймерів та інтервалу оновлення даних з серверу. Створюється буфер для зберігання інформації, отриманої з сервера.

3. Ініціалізація Serial Port: налаштовується віддзеркалення процесів, що відбуваються всередині мікроконтролера. Тут також ініціалізуються піни, і вказується, які з них будуть використовуватися для введення інформації та для керованих елементів.

4. Запуск веб-сервера: мікроконтролер запускає веб-сервер Ethernet Shield W5100 в локальній мережі із зареєстрованими IP- та MAC-адресами. Тут також встановлюється затримка для ініціалізації шилда та відправки POST-запиту на сервер бази даних.

5. Запити на сервер: мікроконтролер виконує запити на сервер для отримання даних з MySQL бази даних. Отримані дані записуються в відповідні змінні в буфері контролера.

6. Відображення готовності: після цих дій контролер показує готовність до роботи. Наприклад, включає вивід під номером 13, що служить індикатором готовності. Також, через Serial Port виводиться текстова лінія з написом "READY". В разі неполадок виводиться повідомлення "FALSE update".

7. Постійний моніторинг: мікроконтролер регулярно перевіряє стан елементів розумного будинку. Наприклад, кожні 10 секунд він перевіряє дані на сервері та оновлює стан кожного функціонального елемента.

8. Взаємодія з кнопками: при натисканні на кнопку керування, програма визначає змінну FLAG, яка відображає стан ввімкнення або вимкнення освітлення. Це дозволяє змінювати стан елемента світла звичайною кнопкою, подібно до клавіші на клавіатурі.

9. Запис у базу даних: при зміні стану перемикача дані, такі як номер піна, його стан та попередній стан перед натисканням кнопки, відправляються на сервер і зберігаються в базі даних MySQL.

10. Функції для комунікації з сервером: функції `sendpin` та `sendmass` використовуються для взаємодії з сервером. `sendpin` відправляє інформацію про стан конкретного піна на сервер. `sendmass` відправляє дані про стан всіх елементів розумного будинку.

Цей алгоритм роботи дозволяє мікроконтролеру ефективно керувати розумним будинком, зчитувати та відправляти дані на сервер, а також надавати зручний інтерфейс для користувачів для керування системою.

Рівень напруги на виводах обмежений 5В. Максимальний струм, який може віддавати або споживати один вивід, становить 40 мА. Усі виводи пов'язані з внутрішніми підтягувальними резисторами (за замовчуванням вимкненими) номіналом 20-50 кОм [15].

Зв'язок Arduino Mega 2560 надає низку можливостей для здійснення зв'язку з комп'ютером, ще одним Ардуіно або іншими мікроконтролерами. У ATmega2560 є чотири апаратні приймачі-передавачі UART для реалізації послідовних інтерфейсів (з логічним рівнем TTL 5В) [14]. Мікроконтролер ATmega16U2 (або ATmega8U2 на платах версії R1 і R2) забезпечує зв'язок одного з приймачів із USB-портом комп'ютера, і в разі під'єднання до ПК дає змогу Ардуіно визначатися як віртуальний COM-порт (для цього операційній системі Windows знадобиться відповідний .inf-файл, на відміну від OSX і Linux, де розпізнавання плати як COM-порту відбувається автоматично). До пакета програмного забезпечення Ардуіно входить спеціальна програма SerialMonitor, що дає змогу зчитувати і надсилати на Ардуіно прості текстові дані. Під час передачі даних через мікросхему ATmega8U2/ATmega16U2 під час USB-з'єднання з комп'ютером, на платі блиматимуть світлодіоди RX і TX. (Під час послідовного передавання даних за допомогою виводів 0 і 1, без використання USB-перетворювача, ці світлодіоди не задіюються).

Бібліотека SoftwareSerial дає змогу реалізувати послідовний зв'язок на будь-яких цифрових виводах Mega2560.

У мікроконтролері ATmega2560 також реалізовано апаратну підтримку послідовних інтерфейсів TWI і SPI. У програмне забезпечення Ардуіно входить бібліотека Wire, що дає змогу спростити роботу з шиною TWI [18].

Захист в Arduino Mega 2560 передбачає наявність відновлюваних запобіжників, що захищають USB-порт комп'ютера від коротких замикань і перевантажень. Незважаючи на те, що більшість комп'ютерів мають власний захист, такі запобіжники забезпечують додатковий рівень захисту. Якщо від USB-порту споживається струм понад 500 мА, запобіжник автоматично розірве з'єднання до усунення причин короткого замикання або перевантаження.

2.6. Серверна частина

Використання Ethernet Shield W5100 як сервер для проєкту розумного будинку. Це рішення може бути досить ефективним для малих проєктів "Розумного будинку", де не потрібно обробляти великі обсяги даних та важкі обчислення на стороні сервера. Ethernet Shield W5100 дозволяє створити веб-сервер, який може приймати запити від контролера та мобільного додатку та надсилати їм відповіді [16].

За допомогою цього Ethernet Shield є можливість забезпечити доступ до різних функцій проєкту "Розумного будинку" через локальну мережу. Це включає в себе керування освітленням, обігрівом, кондиціонером та іншими елементами будинку.

Алгоритм функціонування програми мікроконтролера складається з масштабної послідовності етапів:

1. При ініціалізації мікроконтролера, програма організовує реєстрацію веб-сервера за допомогою Ethernet Shield W5100 в локальній мережі. Цей процес включає встановлення IP-адреси та MAC-адреси для шилду, що надає йому унікальний ідентифікатор у мережі [17].

2. Надається пріоритет ініціалізації всіх глобальних змінних, які мають регулювати функціональність різних аспектів розумного будинку. Крім того, визначаються змінні для налаштування таймерів і частоти оновлення інформації із серверу. Одночасно створюється структура (буфер) для утримання отриманої інформації з серверу [18].

3. На варті стоїть ініціалізація Serial Port з метою відстеження внутрішніх процесів мікроконтролера. Також важливим етапом є ініціалізація пінів, при чому кожному присвоюється функціональне призначення - вхід або вихід для керованих елементів. Одночасно ініціалізується буфер для подальшого використання.

4. Відбувається запуск веб-сервера, вбудованого у Ethernet Shield W5100, в локальній мережі за використанням раніше зареєстрованих IP-адрес та MAC-адрес. Додаткова затримка в 1 секунду встановлюється, щоб шилд повністю ініціалізувався та успішно відправив POST-запит до сервера бази даних.

5. Мікроконтролер звертається до функції, яка включає алгоритм надсилання запитів на сервер. Ця функція відповідає за отримання даних з MySQL бази даних та їх занесення в відповідні змінні в буфері. Після цього дані зчитуються з буферу та призначаються відповідним змінним для керованих елементів.

6. Мікроконтролер активує вивід під номером 13, який виступає індикатором готовності до роботи, та виводить текстову лінію "READY" в Serial Port, щоб повідомити про свою готовність.

7. Після успішної ініціалізації, мікроконтролер готовий до нормальної роботи. У випадку виникнення помилок, вивід під номером 13 не активується, і в Serial Port виводиться повідомлення "FALSE update".

8. Мікроконтролер готовий до функціонування, і що 10 секунд він перевіряє дані на сервері стосовно стану кожного функціонального елемента розумного будинку.

9. При натисканні кнопки керування освітленням, програма аналізує стан кнопки та попередній стан елемента освітлення. Якщо кнопку натиснули, і елемент був вимкнений, то стан елемента змінюється на включений, і навпаки. Це дозволяє відмовитися від стандартних перемикачів і зробити керування освітленням більш інтуїтивним.

10. Будь-які зміни стану елемента та змінні FLAG автоматично надсилаються на сервер та зберігаються в базі даних MySQL. Одночасно існує можливість дистанційного керування елементами.

11. Функція sendpin використовується для відправки стану окремого елемента розумного будинку на сервер. Вона приймає номер піна, його стан та попередній стан і, в разі успішної відправки, виводить "Sendpin TRUE", або "Sendpin FALSE" у випадку невдалої спроби.

12. Функція updatemass застосовується для повного оновлення станів всіх елементів розумного будинку. Приймаються номери пінів, які потрібно оновити. Функція відправляє запит на сервер, отримує стани елементів і оновлює їх на мікроконтролері.

13. Функція sendmass використовується для відправки всіх станів елементів розумного будинку на сервер. Вона приймає дані про стани елементів і надсилає їх на сервер.

Це детальний опис алгоритму роботи програми мікроконтролера для керування різними елементами розумного будинку та моніторингу їх стану з додатковими вдосконаленнями та розширеними можливостями.

2.7. База даних

База даних (БД) складеться з таких таблиць:

- login;
- scenarios;
- scenarios_delete;
- heat_hable;

–info.

На рис. 2.2. показано структуру архітектури БД та серверного КОМПОНЕНТА.

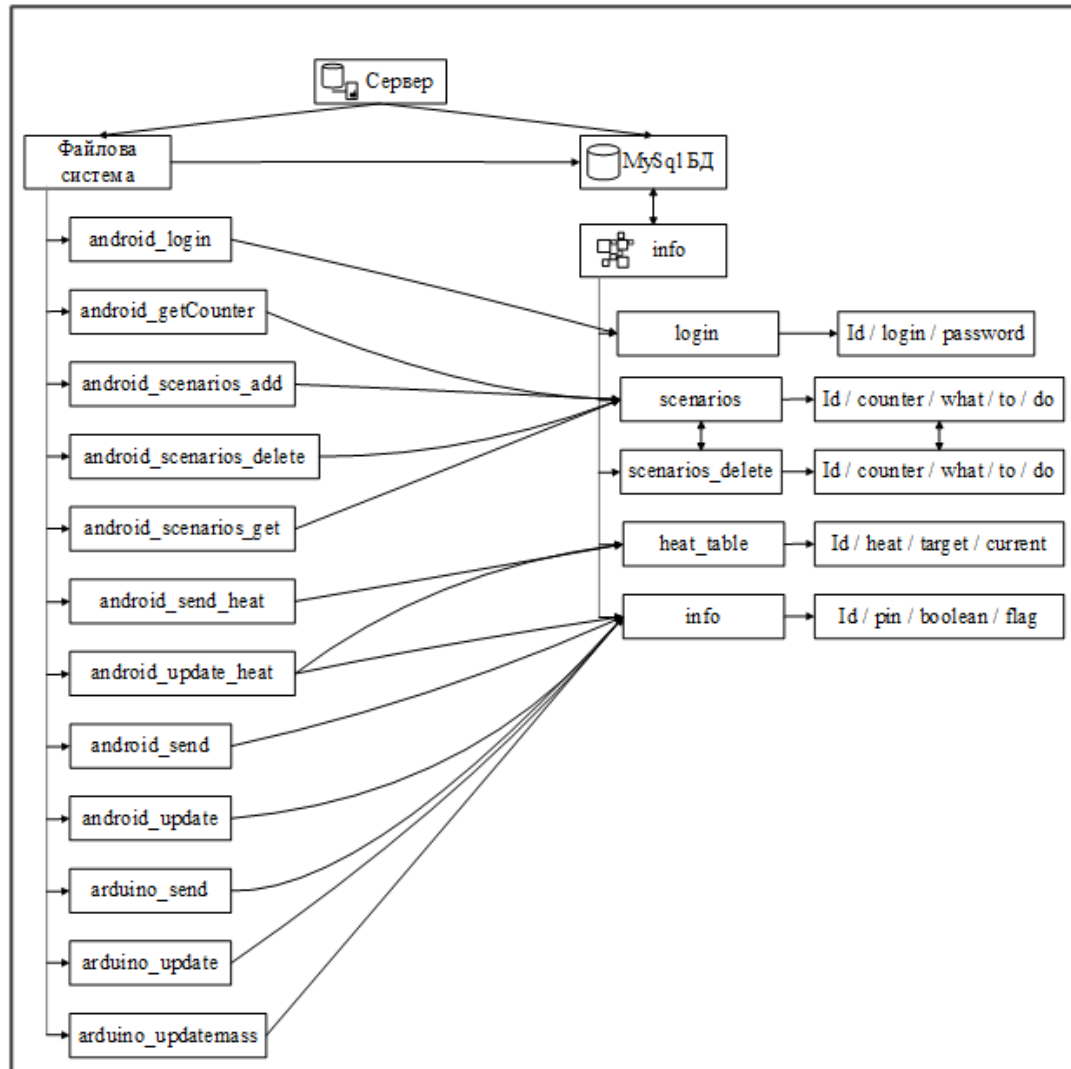


Рис 2.2. Схема архітектури БД та серверної частини

Ще один компонент в системі "Розумний будинок" представляє собою файловою структуру, що охоплює всі файли, розміщені на сервері. Ці файли функціонують з метою забезпечення взаємодії між користувачами інформаційної системи (ІС) і БД.

Призначенням файлової системи є наступні компоненти:

1. Файл `android_login.php`: ця скриптова одиниця відповідає за обробку запитів, отриманих від мобільних клієнтів, щодо авторизації в системі. Вона проводить перевірку введених користувачем логіну та паролю та порівнює їх з відповідними записами у таблиці "login".
2. Файл `android_getCounter.php`: цей компонент використовується для отримання кількості записів у таблиці "scenarios". Ця інформація необхідна для правильного відображення сценаріїв в мобільному додатку [19].
3. Файл `android_scenarios_add.php`: даний файл відповідає за можливість додавання нових сценаріїв до таблиці "scenarios" [19].
4. Файл `android_scenarios_delete.php`: цей компонент забезпечує можливість видалення сценаріїв з таблиці "scenarios" [19].
5. Файл `android_scenarios_get.php`: використовується для отримання всіх сценаріїв для користувачів Android [19].
6. Файл `android_send_heat.php`: ця скриптова одиниця відповідає за оновлення інформації про цільову температуру для нагрівальних приладів, яка зберігається в таблиці "heat_table" [19].
7. Файл `android_update_heat.php`: цей файл використовується для оновлення інформації про нагрівальні прилади в мобільному додатку [20].
8. Файл `android_send.php`: відповідає за можливість оновлення даних в таблиці "info" через Android-додаток [20].
9. Файл `android_update.php`: використовується для отримання інформації від Android-додатку [20].
10. Таблиця "login" включає такі стовпці: id, login, password. В цій таблиці зберігаються дані, необхідні для авторизації в додатку. За доступ до цієї таблиці відповідає скрипт у файлі "android_login.php" [20].
11. використовується для зберігання і управління сценаріями. Файли "android_getCounter.php," "android_scenarios_add.php," "android_scenarios_delete.php," і "android_scenarios_get.php" дозволяють працювати з цією таблицею [21].

12. Таблиця "heat_table" включає стовпці: id, heat, target, current. Вона використовується для зберігання інформації про температурні режими нагрівальних приладів. Файли "android_send_heat.php" і "android_update_heat.php" взаємодіють з цією таблицею через Android-додаток [21].

13. Таблиця "info" включає стовпці: id, pin, boolean, flag. В ній зберігається інформація про стан електроприладів у системі. Файли "android_send.php" і "android_update.php" відповідають за зміну та отримання даних про ці прилади через Android-додаток [21].

14. Скрипти "arduino_send.php," "arduino_update.php," і "arduino_updatemass.php" використовуються для взаємодії з мікроконтролером Arduino та оновлення даних у таблиці "info" [21].

15. Таблиця "scenarios" включає стовпці: id, counter, what, to, do. У цій таблиці містяться різні параметри, необхідні для коректної роботи мобільного додатка для керування розумним будинком. Кожен рядок таблиці включає унікальний ідентифікатор рядка в базі даних (id), порядковий номер сценарію для відображення його в мобільному додатку (counter), порядковий номер конкретного елемента розумного будинку (what), інформацію щодо необхідних дій (to), а також вказівки щодо того, коли чи скільки разів ці дії повинні бути виконані (do). Доступ до цієї таблиці контролюється за допомогою кількох PHP-скриптів, які містяться в файлах: android_getCounter, android_scenarios_add, android_scenarios_delete, android_scenarios_get [21].

Ці скрипти відповідають за взаємодію з БД і забезпечують правильну обробку та передачу інформації між мобільним додатком та джерелом даних.

Схема архітектури системи «Розумний будинок» (Додаток А).

2.8 Висновки

У розділі було проведено детальний аналіз та опис всіх етапів створення програмного забезпечення для систем опалення в розумних будинках:

1. Архітектура програмного продукту: обрана архітектура дозволяє забезпечити гнучкість та масштабованість системи, щоб вона відповідала різноманітним умовам та вимогам користувачів.

2. Вибір платформи: визначено оптимальну платформу для розробки, враховуючи особливості розумних будинків та їхні вимоги.

3. Адаптивний алгоритм: розроблено адаптивний алгоритм, який ефективно реагує на зміни в умовах навколишнього середовища, забезпечуючи оптимальне використання ресурсів.

4. Оптимізація витрат ресурсів: застосовано оптимізаційні алгоритми для зменшення витрат енергії та інших ресурсів, забезпечуючи при цьому зручність користувачів.

5. Безпека системи: враховано питання безпеки для запобігання можливим загрозам та забезпечення конфіденційності даних користувачів.

У розділі поєднано теоретичні аспекти та практична реалізація, спрямована на створення ефективних та інтелектуальних систем опалення в розумних будинках.

РОЗДІЛ 3

РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ І ПРОГРАМНОГО ЗАБЕЗПЕННЯ КЕРУВАННЯМ ТЕМПЕРАТУРНИМ РЕЖИМОМ БУДИНКУ

3.1. Розробка математичної моделі оптимізації витрат енергії для опалення

Розглянемо ММ розрахунку температури всередині будинку. Введемо позначення:

T_{inside} – температура усередині будинку;

$T_{outside}$ – зовнішня температура;

P_{heat} – потужність опалювальної системи;

C_{heat} – тепловмісткість;

R – опір теплопередачі через стіни та вікна;

S – площа будинку;

Q – кількість одиниць тепла, необхідних для підігріву одного кубометра повітря на один градус Цельсія;

Δt – інтервал часу;

$T_{desired}$ – бажана температура всередині приміщення.

Математична модель може бути такою:

$$\Delta T_1 = \frac{(T_{outside} - T_{inside}) \cdot R \cdot S \cdot \Delta t}{(Q \cdot 3600)},$$

де ΔT_1 – зміна температури у будинку протягом інтервалу часу Δt .

Потужність опалювальної системи буде залежати від різниці між бажаною температурою і температурою усередині будинку:

$$P_{heat} = (T_{desired} - T_{inside}) \cdot C_{heat},$$

де C_{heat} – це фізична властивість, яка визначає кількість теплової енергії, необхідної для підвищення температури приміщення на 1 градус Цельсія. C_{heat} вимірюється в джоулях на градус Цельсія ($J/^\circ C$) або кілоджоулях на градус Цельсія ($kJ/^\circ C$).

Завдання оптимізації полягатиме у знаходженні оптимальної потужності опалювальної системи, яка б забезпечила бажану температуру в будинку, мінімізуючи вартість опалення,

мінімізувати:

$$P_{\text{heat}} = (T_{\text{desired}} - T_{\text{inside}}) \cdot C_{\text{heat}},$$

з обмеженнями:

$$\Delta T1 = \frac{(T_{\text{outside}} - T_{\text{inside}}) \cdot R \cdot S \cdot \Delta t}{(Q \cdot 3600)},$$

$$T_{\text{inside}} + \Delta T1 \leq T_{\text{desired}},$$

$$T_{\text{inside}} + \Delta T1 \geq T_{\text{min}},$$

де T_{min} – мінімальна прийнятна температура у будинку.

Візьмемо конкретний приклад, однокімнатну квартиру в Дніпрі, в якій я проживаю і проведемо розрахунок для оптимізації системи опалення на основі її даних.

Площа однокімнатної квартири становить 30 м². Далі необхідно зауважити, що стіни будинку побудовані з силікатної цегли, також присутні вікна в квартирі. Тепер, необхідно розрахувати опір теплопередачі вікон та стін, для цього використовуємо таку формулу:

$$R = R_{\text{скла}} + R_{\text{стін}} = \frac{d}{\lambda},$$

де d – товщина виражена в метрах,

λ – коефіцієнт теплопровідності, виражений у Вт/м·К.

Оскільки, коефіцієнт теплопровідності скла є сталою величиною і становить – 1 Вт/м·К, а товщина скла в моїх вікнах дорівнює 0.5 см, це 0.005 м, то можемо розрахувати дані [25]:

$$R_{\text{скла}} = 0.005 / 1 = 0.005 \text{ м}^2 \cdot \text{К} / \text{Вт}.$$

Тепер, візьмемось до розрахунку опору теплопередачі стіни. Як було вищевказано, вона побудована з силікатної цегли, коефіцієнт теплопровідності якої, становить – 0.7 Вт/м·К, а її товщина дорівнює 0.5 м, отже:

$$R_{\text{стіни}} = 0.5 / 0.7 = 0.715 \text{ м}^2 \cdot \text{К} / \text{Вт},$$

$$R = 0.005 + 0.715 = 0.72 \text{ м}^2 \cdot \text{К} / \text{Вт}.$$

Для того, щоб знайти кількість одиниць тепла, необхідних для підігріву одного кубометра повітря на один градус Цельсія, знадобиться використовувати теплоту (теплоємність) повітря.

Зазвичай теплоту повітря вимірюють у джоулях на кількість повітря на один градус Цельсія ($J/m^3\text{°C}$). Для стандартних умов і нормального атмосферного тиску, це значення близько 1.006 кілоджоулів на кубічний метр на градус Цельсія.

Отже, формула для розрахунку кількості тепла (Q), необхідного для підігріву одного кубометра повітря на ΔT градусів Цельсія виглядає так:

$$Q = (\text{теплота повітря}) * (\text{об'єм повітря}) * (\text{зміна температури}),$$

де $T_{\text{outside}} = 11 \text{ °C}$;

$T_{\text{inside}} = 22 \text{ °C}$;

$S = 30 \text{ м}^2$;

$h = 2.7 \text{ м}$;

"теплота повітря" вимірюється у $J/m^3\text{°C}$;

"об'єм повітря" вимірюється у кубічних метрах (м^3);

"зміна температури" вимірюється в градусах Цельсія (°C).

Теплота повітря визначається його температурою і масою, і може бути обчислена за допомогою специфічної теплоємності повітря [23]. Формула для розрахунку теплоти (K) такого об'єму повітря виглядає наступним чином:

$$K = m * c * \Delta T,$$

де K – теплота (J);

m – маса повітря (kg);

c – специфічна теплоємність повітря;

ΔT_2 – різниця температури повітря (в градусах Цельсія).

Специфічна теплоємність повітря (c) становить приблизно 1005 джоулів на кілограм на градус Цельсія [22]. Тоді проведемо обчислення:

$$\Delta T_2 = t_{\text{inside}} - t_{\text{outside}} = 22 - 11 = 11 \text{ °C}.$$

Отже, щоб розрахувати теплоту повітря, потрібно знати масу повітря (m), специфічну теплоємність повітря (c) і зміну температури (ΔT).

Об'єм квартири залежить від висоти приміщення. Щоб знайти об'єм, потрібно помножити площу квартири на її середню висоту.

Таким чином, для квартири площею 30 квадратних метрів об'єм буде:

$$V = S \cdot h = 30 \cdot 2,7 = 81 \text{ м}^3.$$

Для розрахунку маси повітря потрібно знати його об'єм та густину. Маса (m) може бути обчислена за допомогою наступної формули:

$$m = V \cdot \rho,$$

де m – маса повітря;

V – об'єм повітря;

ρ – густина повітря (зазвичай в кількості маси на одиницю об'єму, наприклад, кількість кілограмів на кубічний метр або грамів на літр).

Густина повітря зазвичай становить близько 1.225 кілограмів на кубічний метр при нормальних умовах (22+ °C і 1 атмосферний тиск) [24]. Проведемо обчислення:

$$m = 81 \cdot 1.225 = 99.225 \text{ кг.}$$

Тепер порахуємо:

$$Q = m \cdot c \cdot \Delta T = 99.225 \cdot 1005 \cdot 11 = 1097932 \text{ J.}$$

Тепер можемо порахувати зміну температури у будинку протягом інтервалу часу Δt :

де $\Delta t = 24 \text{ год} = 1440 \text{ хв} = 86400 \text{ с,}$

$$\Delta T_1 = \frac{(T_{\text{outside}} - T_{\text{inside}}) \cdot R \cdot S \cdot \Delta t}{(Q \cdot 3600)} = \frac{(11 - 22) \cdot 0.72 \cdot 30 \cdot 86400}{(1097932 \cdot 3600)} = -5.2 \text{ }^\circ\text{C.}$$

Специфічна теплоємність C_{heat} для опалювального обладнання може бути розрахована за допомогою такої формули:

$$C_{\text{heat}} = \left(\frac{Q}{m \cdot \Delta T_2} \right),$$

де C_{heat} – специфічна теплоємність ватт на градус Цельсія ($\text{W}/^\circ\text{C}$);

Q – кількість тепла, яка необхідна для опалювального обладнання (ватт, W);

m – маса матеріалу обладнання (кількість матеріалу, який необхідно опалити, кількість повітря тощо) в кількості одиниць маси (наприклад, кількість кілограмів);

ΔT_2 – зміна температури в градусах Цельсія.

Ця формула допомагає визначити, скільки тепла потрібно, щоб підігріти певну масу повітря на певну зміну температури. Після того, як ми знайдемо C_{heat} , можемо використовувати його для подальших розрахунків в системі опалення "Розумного будинку".

$$C_{heat} = (1097932 / (99.225 \cdot 11)) = 1006 \text{ W/}^\circ\text{C}.$$

Тепер, знаючи тепловмісткість, рахуємо потужність опалювальної системи:

$$P_{heat} = (T_{desired} - T_{inside}) \cdot C_{heat} = (27.5 - 22) \cdot 1006 = 5533 \text{ W}.$$

Тепер, необхідно зробити обмеження:

$$\Delta T_1 = \frac{(T_{outside} - T_{inside}) \cdot R \cdot S \cdot \Delta t}{(Q \cdot 3600)},$$

$$T_{inside} + \Delta T_1 \leq T_{desired},$$

$$T_{inside} + \Delta T_1 \geq T_{min},$$

де T_{min} – мінімальна прийнятна температура у будинку;

$T_{desired} = 27.5 \text{ }^\circ\text{C}$;

$T_{min} = 16.5 \text{ }^\circ\text{C}$.

Обмеження мають наступний вигляд:

$$\Delta T_1 = (11 - 22) \cdot 0.72 \cdot 30 \cdot 86400 / (1097932 \cdot 3600),$$

$$22 + (-5.2) \leq 27.5 ,$$

$$22 + (-5.2) \geq 16.5 .$$

Також, знаючи, потужність опалювальної системи та сьогоднішні тарифи, можна порахувати вартість опалення квартири на місяць:

$$\text{Вартість опалення} = 5.533 \cdot 2.62 \cdot 30 = 435 \text{ грн.}$$

За допомогою оптимізації системи опалення та ефективному використанню ресурсів, вартість опалення на місяць становитиме 435 грн.

Тепер необхідно розробити програмний код, для цього я використаю JavaScript. Цей код розраховує специфічну теплоємність опалювального обладнання (C_{heat}) та потужність опалювальної системи (P_{heat}) на основі вхідних даних та розрахунків:

```
// Вхідні дані
const T_inside = 22; // температура усередині будинку (°C)
const T_outside = 11; // зовнішня температура (°C)
const S = 30; // площа будинку (м²)
const h = 2.7; // висота приміщення (м)
const Q = 1097932; // кількість одиниць тепла для підігріву 1 м³ повітря на
1 °C (J/м³°C)
const Δt = 86400; // інтервал часу (с)
const T_desired = 27.5; // бажана температура всередині приміщення (°C)
const T_min = 16.5; // мінімальна прийнятна температура у будинку (°C)
// Розрахунок опору теплопередачі
const R_glass = 0.005; // опір теплопередачі скла (м²*°C/W)
const R_wall = 0.72; // опір теплопередачі стін (м²*°C/W)
const R = R_glass + R_wall; // загальний опір теплопередачі (м²*°C/W)
// Розрахунок об'єму приміщення
const V = S * h; // об'єм приміщення (м³)
// Розрахунок маси повітря
const airDensity = 1.225; // густина повітря (кг/м³)
const m = V * airDensity; // маса повітря (кг)
// Розрахунок зміни температури
const ΔT1 = (T_outside - T_inside) * R * S * Δt / (Q * 3600); // зміна
температури приміщення (°C)
// Перевірка меж температур
if (T_inside + ΔT1 <= T_desired && T_inside + ΔT1 >= T_min) {
// Розрахунок специфічної теплоємності
const ΔT2 = T_inside - T_outside; // різниця температур
```

```

const c_p = 1005; // специфічна теплоємність повітря (J/kg°C)
const C_heat = Q / (m * ΔT2); // специфічна теплоємність опалювального
обладнання (W/°C)
// Розрахунок потужності опалювальної системи
const P_heat = (T_desired - T_inside) * C_heat; // потужність опалювальної
системи (W)
console.log("Специфічна теплоємність опалювального обладнання
(C_heat):", C_heat, "W/°C");
console.log("Потужність опалювальної системи (P_heat):", P_heat, "W");
} else {
console.log("Необхідно змінити параметри системи опалення для
досягнення бажаної температури.");
}

```

Цей програмний код на основі математичної моделі дозволить системі розумного будинку оптимізувати використання опалювальної системи, забезпечуючи комфортні умови для мешканців та ефективне використання ресурсів.

Тепер необхідно підключити датчики до нашої плати Arduino Mega 2560 (рис 3.1.).

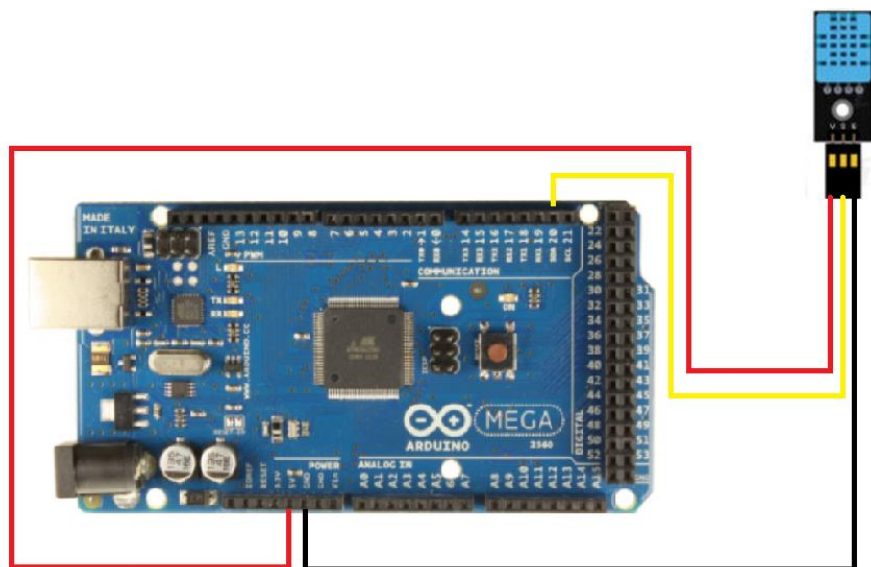


Рис 3.1. Arduino Mega 2560 з датчиками

3.2. Підключення до Arduino

Модуль обладнаний трьох піновим роз'ємом стандарту 2.54 мм до яких підключається [11]:

V – підключається до виводу GND (температура);

R – підключається до виводу + 5V (живлення)

Y – підключається до цифрового входу (SDA 20).

Програмний код для зчитування та передачі інформації датчиком в Arduino:

```
Бібліотека необхідна для роботи з модулем dht11
#include <dht11.h> // Додаємо бібліотеку DHT11
dht11 DHT; // Оголошення змінної класу dht11
#define DHT11_PIN 4 // Датчик DHT11 підключений до цифрового піна
номер 4
void setup() {
  Serial.begin(9600); // Швидкість роботи порту
  Serial.println("DHT TEST PROGRAM "); // Виводимо текст
  Serial.print("LIBRARY VERSION: "); // Виводимо текст
  Serial.println(DHT11LIB_VERSION);
  Serial.println(); // Пустий рядок
}
void loop() {
  int chk;
  // Моніторинг помилок
  chk = DHT.read(DHT11_PIN); // Зчитуємо дані
  switch (chk) {
  case DHTLIB_OK:
    break;
  case DHTLIB_ERROR_CHECKSUM:
    Serial.println("Checksum error, \t");
```

```

break;
case DHTLIB_ERROR_TIMEOUT:
Serial.println("Time out error, \t");
break;
default:
Serial.println("Unknown error, \t");
break;
}
// Виводимо значення вологості та температури
Serial.print("Вологість = ");
Serial.print(DHT.humidity, 1);
Serial.print(", Температура = ");
Serial.println(DHT.temperature, 1);
delay(1000);
}

```

Оскільки, при розрахунках було доведено, що:

$$C_{heat} = (1097932 \text{ J} / (99.225 \text{ кг} * 11 \text{ }^\circ\text{C})) = 1006 \text{ W}/^\circ\text{C}.$$

Зазвичай при стандартних умовах ця фізична величина дорівнює 1005 W/°C, то в подальших розрахунках при побудові графіку залежності, буде використовуватись специфічна теплоємність повітря у звичайних умовах.

Щоб порахувати потужність опалювальної системи та графічно зобразити залежність P_{heat} від T_{inside} (рис. 3.2.), використаємо цю формулу:

$$P_{heat} = (T_{desired} - T_{inside}) * C_{heat}.$$

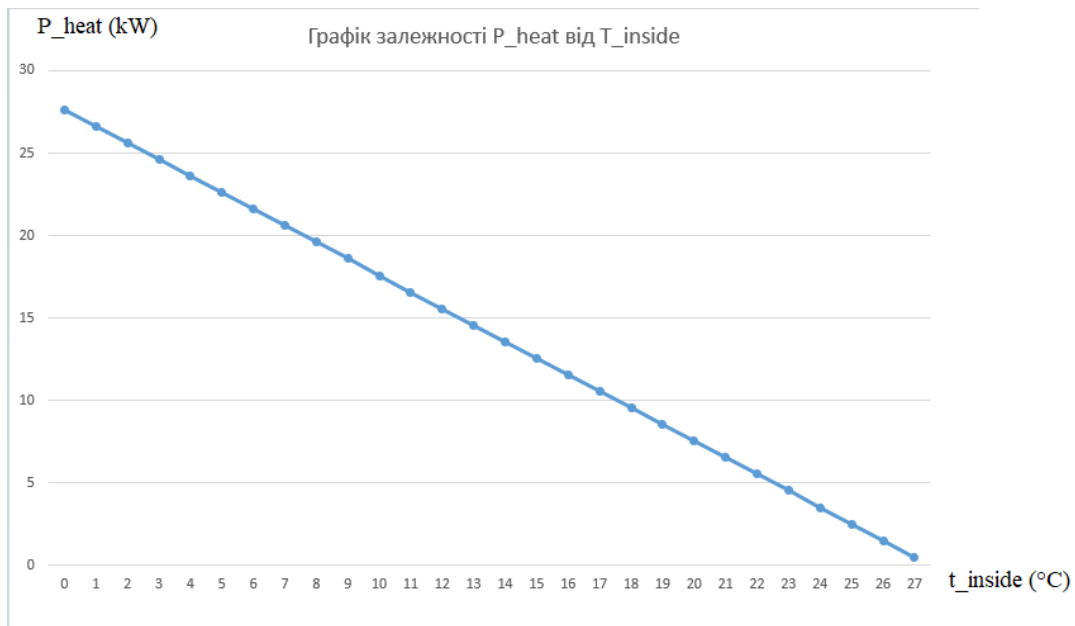


Рис. 3.2. Графік залежності потужності опалювальної системи від температури всередині будинку

Розглянувши графік заженості, також, враховуючи оптимізацію системи опалення задля зменшення витрат ресурсів, можемо зробити висновок, що при збільшенні температури всередині будинку на 1°C, потужність опалювальної системи зменшується приблизно на 1 kW.

3.3 Висновки

В розділі було проведено роботу, спрямовану на створення надійної та ефективної системи керування температурним режимом будинку:

1. Математична модель: розроблена математична модель є досконалим інструментом для передбачення та оптимізації температурного режиму в будинку. Вона враховує велику кількість факторів, таких як розміри будинку, теплові втрати, ефективність системи опалення, та забезпечує точний аналіз та контроль.

2. Програмне забезпечення керування: реалізовано програмне забезпечення для керування температурним режимом.

3. Потенційні переваги: створена система має потенційні переваги у сфері енергоефективності та зручності в управлінні будинком. Вона сприяє економії ресурсів та створює комфортні умови для мешканців.

Можна зазначити, що розробка математичної моделі та програмного забезпечення керування температурним режимом будинку є ключовим кроком у створенні інтелектуальних та ефективних систем опалення в сучасних розумних будинках.

ВИСНОВОК

У результаті виконання кваліфікаційної роботи було розроблене програмне забезпечення розрахунку системи опалення в проєкті розумний будинок. Це має великий вплив на розвиток розумних будинків. В сучасному світі, де економія використання енергії набуває все більшого значення, розробка ефективних систем опалення для розумних будинків стає надзвичайно важливою. Кваліфікаційна робота охоплює широкий спектр аспектів, включаючи розробку архітектури БД та файлової системи, впровадження веб серверу та технологій IoT, розробку мобільного додатку, математичну модель, оптимізацію ресурсів та аналіз енергетичної ефективності.

За допомогою MySQL була реалізована БД та файлова система. БД збирає дані з датчику dht11 та передає на сервер. Серверна частина складається з Ethernet Shield W5100, який може приймати запити від контролера та мобільного додатку та надсилати їм відповіді.

За допомогою Ethernet Shield була реалізована можливість забезпечити доступ до різних функцій проєкту "Розумного будинку" через локальну мережу. Це включає в себе керування освітленням та температурним режимом будинку.

Технології IoT в кваліфікаційній роботі були реалізовані за допомогою мікропроцесору Arduino Mega 2560 rev 3 та датчику dht11, який вимірює температуру та передає дані через Ethernet Shield W5100 до БД.

За допомогою JS був розроблений мобільний додаток для ОС Android. В мобільному додатку була розроблена система керування елементами будинку, була реалізована аутентифікація користувача та моніторинг даних.

Була визначена ММ, яка дозволяє точно врахувати різноманітні чинники, що впливають на температуру в будинку. Врахування зовнішніх умов, розмірів будинку, опору теплопередачі та ефективності опалювальної системи дозволяє прогнозувати та керувати кліматом з максимальною точністю.

Оптимізація витрат енергії: націлені розрахунки спрямовані на оптимізацію вартості опалення, забезпечуючи мінімальні витрати енергії при

досягненні бажаної температури в приміщенні. Це не лише економічно вигідно, але й допомагає зменшити вплив на навколишнє середовище.

Розрахунок опору теплопередачі: застосування формул для розрахунку опору теплопередачі через вікна та стіни стає ключовим аспектом ефективності системи. Завдяки цьому можливо точно визначити тепловтрати та покращити ізоляцію будинку, що сприяє ще більш ефективному використанню енергії.

Аналіз енергетичної ефективності: детальний аналіз теплоти повітря, специфічної теплоємності та інших факторів забезпечує точний розрахунок енергетичної ефективності опалювальної системи. Це дозволяє нам зрозуміти, як система витрачає та виробляє енергію в різних умовах.

Вплив на розвиток галузі та важливість автоматизації: оптимізація ресурсів та використання математичних моделей в розумних системах опалення є кроком до зменшення витрат енергії та екологічного впливу. Автоматизовані системи контролю дозволяють підтримувати комфорт в будинку, ефективно використовуючи енергію та ресурси.

Необхідність подальших досліджень: перспективи досліджень включають розвиток більш точних та адаптивних математичних моделей. Врахування зовнішніх чинників та вдосконалення систем автоматизації сприятимуть максимальній ефективності та збереженню ресурсів.

Сучасні технології ШІ відкривають нові перспективи для оптимізації ресурсів у системах розумного дому. Надзвичайно важливою є їх роль у покращенні ефективності опалювання та зменшенні енерговитрат. Наші висновки вказують на необхідність подальших досліджень та впровадження ШІ у такі системи. Застосування алгоритмів машинного навчання може значно поліпшити адаптивність систем опалення до індивідуальних умов користувача та зовнішнього середовища.

Розумні системи можуть вчитися з плином часу, аналізуючи споживання енергії, зовнішні чинники та представлення користувача щодо комфорту. ШІ може автоматично адаптувати роботу систем опалення в режимі реального часу, реагуючи на зміни в погоді, розклад роботи та інші параметри.

Використання ІІ в системах розумного дому також може значно полегшити взаємодію користувача з системою. Голосові асистенти та інтерфейси, базовані на мові, забезпечують зручний та інтуїтивний спосіб керування. Це робить процес налаштування параметрів опалення більш доступним та зрозумілим для користувача.

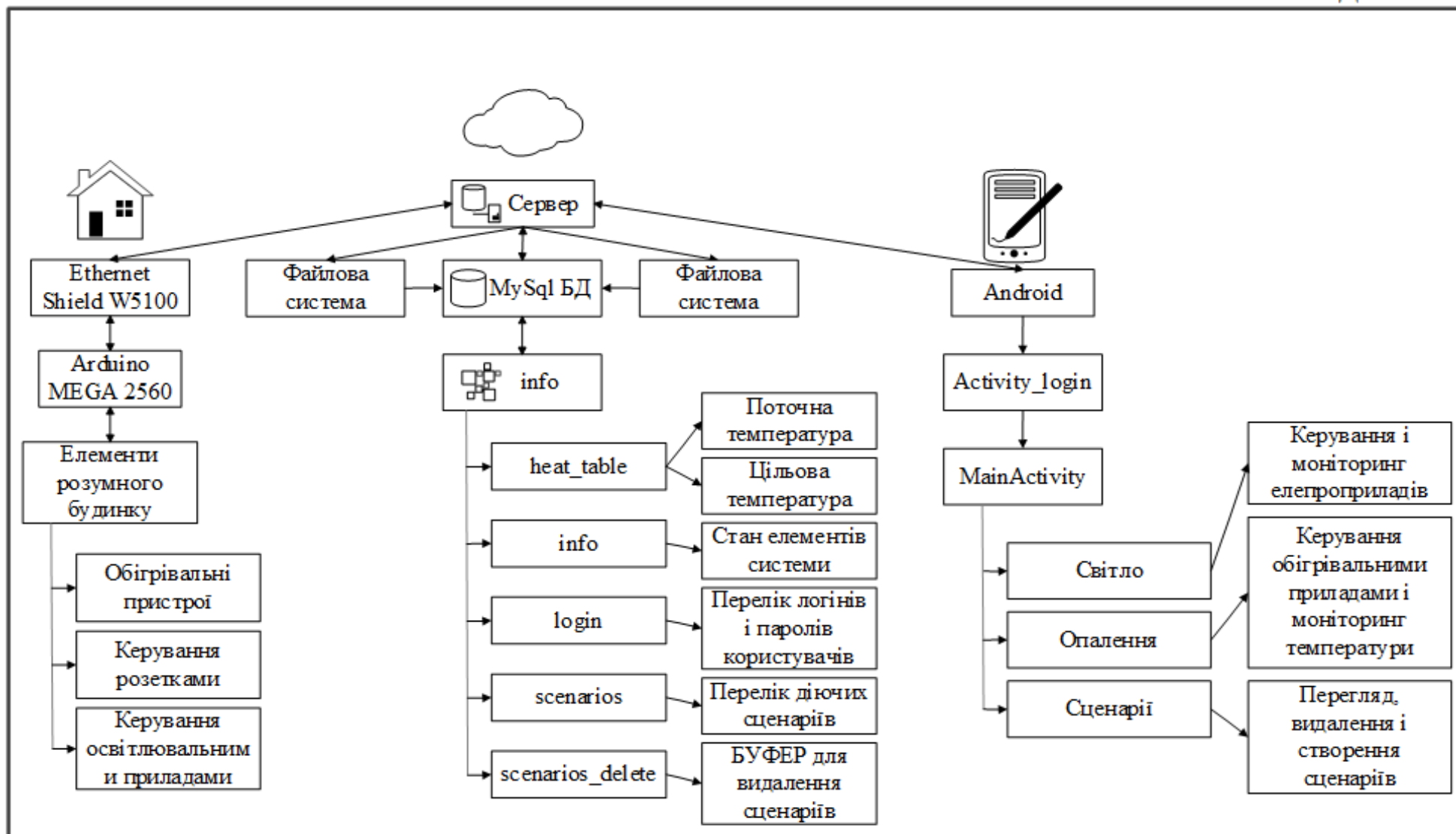
Такий підхід дозволяє створити справжньо реактивні системи опалення, які адаптуються до змінних умов та індивідуальних уподобань. Це не тільки сприятиме збереженню енергії, але і покращить комфорт та якість життя кожного мешканця розумного будинку.

Висновки роботи засвідчують важливість вивчення та впровадження нових підходів у розробці систем опалення для сучасних будинків. Здатність точно прогнозувати, оптимізувати та ефективно використовувати енергію стане ключем до створення більш екологічного та комфортного житла у майбутньому.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. S. S. Rajput. Smart Home Energy Management System. – Chapter 2 – 2019 – P. 32-33.
2. E. L. da Silva. Predictive Maintenance and Energy Management in Smart Homes. – Chapter 4 – 2018 – P. 17.
3. A. G. Cohn. An Intelligent Agent-based Architecture for Smart Home Control. – Chapter 3 – 2001 – P. 6-7.
4. P. Kumar. IoT-Based Smart Home. – Chapter 6 – 2017 – P. 3-5.
5. M. B. Shadmand. Energy Management for Smart Homes Using Renewable Energy Resources and Domestic Load Shifting. – Chapter 3 – 2017 – P.7.
6. J. G. Vink . Smart Homes and Beyond: Icost 20. – Chapter 5 – 2020 – P.9.
7. D. Briere. Home Automation For Dummies. – Chapter 2 – 2015 – P.13-14.
8. Zigbee Alliance – [Електронний ресурс] – <https://csa-iot.org/all-solutions/zigbee/>
9. KNX Association– [Електронний ресурс] – <https://www.knx.org/knx-en/for-professionals/index.php>
10. IEEE Xplore – [Електронний ресурс] – <https://ieeexplore.ieee.org/Xplore/home.jsp>
11. Arduino. Вимірювання температури – [Електронний ресурс] – <https://arduino.ua/ru/art9-arduino-izmerenie-temperatyri-s-ispolzovaniem-datchikov-ds18b20-na-neskolkih-onewire-shinah->
12. Android Architecture – [Електронний ресурс]– <https://www.geeksforgeeks.org/android-architecture>
13. What is a Native Mobile App Development? – [Електронний ресурс] – <https://mdevelopers.com/blog/what-is-a-native-mobile-appdevelopment>
14. Arduino Mega 2560 Rev3 і проекти на його основі – [Електронний ресурс] – <https://evo.net.ua/ru/arduino-mega-2560-rev3-i-proekty-na-ego-osnove/>

15. Arduino Kit – [Електронний ресурс] –
<https://arduinokit.com.ua/ua/p1080109370-arduino-mega-2560.html>
16. Ethernet Shield (W5100) – [Електронний ресурс] –
<https://hobbytech.com.ua/product/plata-rasshireniya-ethernet-shield-w5100/>
17. Ethernet модуль – [Електронний ресурс] –
https://wiki.tntu.edu.ua/Ethernet_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8C
18. В.А. Шахнова. Мікропроцесори та мікропроцесорні комплекти інтегральних мікросхем. – Том 2 –1988 – 48с.
19. SQL Server 2022 – [Електронний ресурс] –
<https://learn.microsoft.com/ru-ru/sql/t-sql/queries/select-group-by-transact-sql?view=sql-server-ver16>
20. PostgreSQL – [Електронний ресурс] –
<https://www.postgresql.org/docs/16/index.html>
21. API MySQL – [Електронний ресурс] –
<https://www.php.net/manual/ru/book.mysql.php>
22. Теплоємність повітря – [Електронний ресурс] –
<https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BF%D0%BB%D0%BE%D1%94%D0%BC%D0%BD%D1%96%D1%81%D1%82%D1%8C>
23. Теплоота повітря – [Електронний ресурс] –
https://uk.wikipedia.org/wiki/%D0%9A%D1%96%D0%BB%D1%8C%D0%BA%D1%96%D1%81%D1%82%D1%8C_%D1%82%D0%B5%D0%BF%D0%BB%D0%BE%D1%82%D0%B8
24. Густина повітря – [Електронний ресурс] –
https://uk.wikipedia.org/wiki/%D0%93%D1%83%D1%81%D1%82%D0%B8%D0%BD%D0%B0_%D0%BF%D0%BE%D0%B2%D1%96%D1%82%D1%80%D1%8F
25. Опір теплопередачі – [Електронний ресурс] – <https://dci-group.com.ua/ua/shho-take-opir-teploperedachi>



ЛІСТИНГ ПРОГРАМИ

Проманий код для застосунку Android:

```
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class SmartHomeControlActivity extends AppCompatActivity {
private static final String TAG = "SmartHomeControlActivity";
// Метод для перевірки аутентифікації користувача
private boolean isUserAuthenticated(String username, String password) {
SharedPreferences preferences = getSharedPreferences("users", MODE_PRIVATE);
String savedPassword = preferences.getString(username, "");
return savedPassword.equals(password);
}
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_smart_home_control);
// Ініціалізація кнопок для керування системою
Button turnOnLightsButton = findViewById(R.id.btnTurnOnLights);
Button turnOffLightsButton = findViewById(R.id.btnTurnOffLights);
Button adjustTemperatureButton = findViewById(R.id.btnAdjustTemperature);

// Додавання обробників подій для кнопок
turnOnLightsButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Логіка для увімкнення світла
Log.i(TAG, "Користувач увімкнув світло");
Toast.makeText(SmartHomeControlActivity.this, "Світло увімкнено",
Toast.LENGTH_SHORT).show();
}
});
turnOffLightsButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Логіка для вимкнення світла
Log.i(TAG, "Користувач вимкнув світло");
Toast.makeText(SmartHomeControlActivity.this, "Світло вимкнено", Toast.LENGTH_SHORT).show();
}
});
adjustTemperatureButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// Логіка для налаштування температури
Log.i(TAG, "Користувач налаштував температуру");
Toast.makeText(SmartHomeControlActivity.this, "Температура налаштована",
Toast.LENGTH_SHORT).show();
}
}
```

```

});
// Додавання обробників подій для аутентифікації
Button loginButton = findViewById(R.id.btnLogin);
loginButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
EditText usernameEditText = findViewById(R.id.editTextUsername);
EditText passwordEditText = findViewById(R.id.editTextPassword);
String username = usernameEditText.getText().toString();
String password = passwordEditText.getText().toString();
if (isUserAuthenticated(username, password)) {
// Аутентифікація успішна, дозвіл на керування системою
turnOnLightsButton.setEnabled(true);
turnOffLightsButton.setEnabled(true);
adjustTemperatureButton.setEnabled(true);
Log.i(TAG, "Користувач увійшов в систему");
} else {
// Помилка аутентифікації, виведення повідомлення
Toast.makeText(SmartHomeControlActivity.this, "Неправильне ім'я користувача або пароль",
Toast.LENGTH_SHORT).show();
Log.w(TAG, "Неуспішна спроба входу в систему");
}
}
});
}
}
}
}

```

Програмний код розрахунку: (C_heat) (P_heat) на основі вхідних даних та розрахунків:

```

// Вхідні дані
const T_inside = 22; // температура усередині будинку (°C)
const T_outside = 11; // зовнішня температура (°C)
const S = 30; // площа будинку (м²)
const h = 2.7; // висота приміщення (м)
const Q = 1097932; // кількість одиниць тепла для підігріву 1 м³ повітря на 1 °C (J/m³°C)
const Δt = 86400; // інтервал часу (с)
const T_desired = 27.5; // бажана температура всередині приміщення (°C)
const T_min = 16.5; // мінімальна прийнятна температура у будинку (°C)
// Розрахунок опору теплопередачі
const R_glass = 0.005; // опір теплопередачі скла (м²*°C/W)
const R_wall = 0.72; // опір теплопередачі стін (м²*°C/W)
const R = R_glass + R_wall; // загальний опір теплопередачі (м²*°C/W)
// Розрахунок об'єму приміщення
const V = S * h; // об'єм приміщення (м³)
// Розрахунок маси повітря
const airDensity = 1.225; // густина повітря (кг/м³)
const m = V * airDensity; // маса повітря (кг)
// Розрахунок зміни температури
const ΔT1 = (T_outside - T_inside) * R * S * Δt / (Q * 3600); // зміна температури приміщення (°C)

```



```

// Перевірка меж температур
if (T_inside + ΔT1 <= T_desired && T_inside + ΔT1 >= T_min) {
// Розрахунок специфічної теплоємності
const ΔT2 = T_inside - T_outside; // різниця температур
const c_p = 1005; // специфічна теплоємність повітря (J/kg°C)
const C_heat = Q / (m * ΔT2); // специфічна теплоємність опалювального обладнання (W/°C)
// Розрахунок потужності опалювальної системи
const P_heat = (T_desired - T_inside) * C_heat; // потужність опалювальної системи (W)
console.log("Специфічна теплоємність опалювального обладнання (C_heat):", C_heat, "W/°C");
console.log("Потужність опалювальної системи (P_heat):", P_heat, "W");
} else {
console.log("Необхідно змінити параметри системи опалення для досягнення бажаної температури.");
}
}

```

Програмний код для зчитування та передачі інформації датчиком в

Arduino:

```

Бібліотека необхідна для роботи з модулем dht11
#include <dht11.h> // Додаємо бібліотеку DHT11
dht11 DHT; // Оголошення змінної класу dht11
#define DHT11_PIN 4 // Датчик DHT11 підключений до цифрового піна номер 4
void setup() {
Serial.begin(9600); // Швидкість роботи порту
Serial.println("DHT TEST PROGRAM "); // Виводимо текст
Serial.print("LIBRARY VERSION: "); // Виводимо текст
Serial.println(DHT11LIB_VERSION);
Serial.println(); // Пустий рядок
}
void loop() {
int chk;
// Моніторинг помилок
chk = DHT.read(DHT11_PIN); // Зчитуємо дані
switch (chk) {
case DHTLIB_OK:
break;
case DHTLIB_ERROR_CHECKSUM:
Serial.println("Checksum error, \t");
break;
case DHTLIB_ERROR_TIMEOUT:
Serial.println("Time out error, \t");
break;
default:
Serial.println("Unknown error, \t");
break;
}
// Виводимо значення вологості та температури
Serial.print("Вологість = ");
Serial.print(DHT.humidity, 1);
Serial.print(", Температура = ");
Serial.println(DHT.temperature, 1);
delay(1000);
}

```

ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файла	Опис
Пояснювальні документи	
Derkach_121m_22_2_diploma.docx	Пояснювальна записка до магістерської роботи. Документ Word.
Derkach_121m_22_2_diploma.pdf	Пояснювальна записка до магістерської роботи. Документ PDF.
Презентація	
Derkach_121m_22_2_presentation.pptx	Презентація до магістерської роботи