

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Мобільний додаток для керування децентралізованою системою опалення

Виконав: студент 2 курсу, групи 2АКІТ-21м
спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології



Андрій ЛІЩУК
Ім'я ПРІЗВИЩЕ

Керівник: д.т.н., професор каф. КСУ
степінь, звання, посада



Володимир ДУБОВОЙ
Ім'я ПРІЗВИЩЕ

« 12 » грудня 2022 р.

Опонент: доцент. каф. АІТ
степінь, звання, посада



Володимир ГАРМАШ
Ім'я ПРІЗВИЩЕ

« 16 » грудня 2022 р.

Допущено до захисту
Зав. кафедри КСУ
В'ячеслав КОВТУН

« 14 » грудня 2022

Вінницький національний технічний університет
 Факультет інтелектуальних інформаційних технологій та автоматизації
 Кафедра комп'ютерних систем управління
 Рівень вищої освіти другий (магістерський)
 Галузь знань – 15 – Автоматизація та приладобудування
 Спеціальність – 151 – Автоматизація та комп'ютерно-інтегровані технології
 Освітньо - професійна програма – Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ
 Завідувач кафедри КСУ



В'ячеслав КОВТУН

"03" жовтня 2022 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ



студенту Ліщук Андрію Романовичу
 (прізвище, ім'я, по батькові)

1. Тема роботи Мобільний додаток для керування децентралізованою системою опалення
 керівник роботи Дубовой Володимир Михайлович
 затверджені наказом ВНТУ від "14" вересня 2022 року №203
2. Термін подання студентом роботи "12" грудня 2022 року
3. Вихідні дані до роботи: кросплатформеність мобільного додатку, зв'язок між пристроями по протоколу BLE, захищеність від злому та зовнішніх перешкод, можливість керування температурним режимом контролера

4. Зміст текстової частини: вступ, аналіз предметної області та огляд аналогів, огляд структури технологій та функціональності мобільного додатку, розробка та тестування мобільного додатку

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): об'єкт, предмет та мета дослідження, задачі дослідження, основні системи управління децентралізованим опаленням, блок-схема порядку дій, обмеження та переваги протоколу Bluetooth Low Energy, процес з'єднання пристроїв BLE, середовище розробки Visual Studio Code, середовище тестування Android Studio, зовнішній вигляд мобільного додатку, тестування програми, економічна частина, висновки


1. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
4	Небава М.І., професор кафедри ЕПВМ		

2. Дата видачі завдання "03" жовтня 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз мов програмування для створення мобільних додатків	04.10.2022	09.10.2022	
2	Огляд та аналіз аналогів керування децентралізованою системою опалення	07.10.2022	13.10.2022	
3	Огляд та аналіз технологій передачі та захисту інформації через протокол BLE	22.10.2022	29.10.2022	
4	Обґрунтування функціональності мобільного додатку	06.11.2022	13.11.2022	
5	Розробка та тестування програмного забезпечення	09.11.2022	25.11.2022	
6	Розрахунок економічної частини	06.12.2022	10.12.2022	
7	Графічні матеріали	18.12.2022	19.12.2022	

Студент 
(підпис)

Андрій ЛІЩУК
(Ім'я ПРІЗВИЩЕ)

Керівник роботи 
(підпис)

Володимир ДУБОВОЙ
(Ім'я ПРІЗВИЩЕ)

АНОТАЦІЯ

УДК 004.42

Ліщук А. Р. Мобільний додаток для керування децентралізованою системою опалення. Магістерська кваліфікаційна робота зі спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології, освітня програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2022. 121 с.

На укр. мові. Бібліогр.: 37 назв; рис.: 24; табл. 7.

У роботі розглянуто проблему в області управління децентралізованої системи опалення, а саме керування електронагрівальними приладами.

Розглянуто різні підходи та прилади керування системою опалення, а саме їх принцип роботи та варіанти використання. Ці підходи було порівняно та проаналізовано.

Розроблено кросплатформений мобільний додаток за допомогою фреймворка React Native, який керує температурою повітря в контролері побудованому на базі плати ESP32. Передача даних реалізована за допомогою протоколу Bluetooth Low Energy.

Ключові слова: протокол BLE(Bluetooth Low Energy), React Native, децентралізована система опалення, мобільний додаток.

ANNOTATION

UDC 004.42

Lischuk A. R. Mobile application for managing a decentralized heating system. Master's thesis on specialty 151 - Automation and computer-integrated technologies, educational program - Intelligent computer systems. Vinnytsia: VNTU, 2022. 121p.

In Ukrainian speech Bibliography: 37 titles; fig.: 24; table 7.

The paper considers the problem in the field of decentralized heating system management, namely the management of electric heating devices.

Different approaches and devices for controlling the heating system are considered, namely their principle of operation and options for use. These approaches were compared and analyzed.

A cross-platform mobile application was developed using the React Native framework, which controls the air temperature in the controller built on the basis of the ESP32 board. Data transfer is implemented using the Bluetooth Low Energy protocol.

Keywords: BLE (Bluetooth Low Energy) protocol, React Native, decentralized heating system, mobile application.

**Відгук
керівника магістерської кваліфікаційної роботи**

студента Ліщука Андрія Романовича група 2АКІТ-21м
на тему: Мобільний додаток для керування децентралізованою системою опалення.

В контексті спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» наукова робота має високий рівень актуальності, що підтверджується в наслідок аналізу наведеної інформації та списку використаних джерел. Актуальність також підтверджується публікацією тез на всеукраїнській науково-технічній конференції «Молодь в науці: дослідження, проблеми, перспективи».

Наукова робота написана лаконічно та зрозуміло, що підтверджується послідовністю виконаних дій: аналіз проблеми, огляд способів вирішення, обґрунтування та аналіз технологій для вирішення проблеми, розробка та тестування програмного забезпечення. Раціональність виконаних дій підтверджується результатами роботи.

Студент показав високі навички в раціональному викладені думок та ефективності в комунікативній роботі, що підтверджується самостійним виконанням усіх завдань по затвердженому графіку не відхиляючись від нього.

Високий ступінь оригінальності підтверджує самостійність виконання поставлених задач. Комунікація із керівником відбувалася періодично, втрати зв'язку не було.

Недоліки: Бажано було використовувати більше наукових праць у списку використаних джерел. Дизайн додатку має сумнівний вигляд. Додаток не протестований на достатній кількості фізичних пристроїв.

Загалом магістерська кваліфікаційна робота відповідає спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», заслуговує на оцінку А, а її автор заслуговує присудження кваліфікації: ступінь вищої освіти магістр, спеціальність «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інтелектуальні комп'ютерні системи».

Керівник магістерської кваліфікаційної роботи

д.т.н., професор каф. КСУ



Дубовой В.М.

**Відгук
опонента на магістерську кваліфікаційну роботу**

студента Ліщука Андрія Романовича група 2АКІТ-21м
на тему: Мобільний додаток для керування децентралізованою системою опалення.

Актуальність наукової роботи зумовлена великою кількістю інформації, яка з'являється останнім часом на відкритих інтернет ресурсах. Актуальність також підтверджується наявністю публікованих тез на всеукраїнській науково-технічній конференції «Молодь в науці: дослідження, проблеми, перспективи».

Обрану тему повністю розкрито. Раціонально оглянуто та проаналізовано досліджувану проблему, та розглянуто способи її вирішення. Способи та технології вирішення поставленої задачі проаналізовано та обґрунтовано в повній мірі.

Проектування функціоналу додатку здійснено на високому рівні. Список використаної літератури обрано правильно та раціонально, що узгоджується із поставленою задачею та вирішенням її.

Графічна частина повністю охоплює всі етапи роботи, ілюстрації та таблиці зображено якісно та доцільно.

Недоліки: В роботі розроблений функціонал є неповним. Інтерфейс мобільного додатку нестандартний та недопрацьований.

Загалом магістерська кваліфікаційна робота відповідає спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», заслуговує на оцінку А, а її автор заслуговує присудження кваліфікації: ступінь вищої освіти магістр, спеціальність «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інтелектуальні комп'ютерні системи».

Опонент на магістерську кваліфікаційну роботу

доцент каф. АІТ



Гармаш В.В.

ЗМІСТ

ВСТУП	10
1 АНАЛІЗ СИСТЕМ КЕРУВАННЯ ТЕМПЕРАТУРОЮ У ПРИМІЩЕННІ ТА ОГЛЯД АНАЛОГІВ.....	12
1.1 Актуальність проблеми управління температурою у приміщенні.....	12
1.2 Аналіз предметної області розробки мобільних додатків.....	14
1.3 Регулятори опалення та їх види	19
1.3.1 Механічні та електричні термостати.....	21
1.3.2 Система «розумний будинок»	27
1.3.3 GSM розетки та GSM блок-модулі.....	33
Висновки до розділу	37
2 АНАЛІЗ ТЕХНОЛОГІЙ ТА ФУНКЦІОНАЛЬНОСТІ МОБІЛЬНОГО ДОДАТКУ	38
2.1 Функціональність мобільного додатку	38
2.2 Принцип роботи Bluetooth Low Energy	41
2.3 Безпека передачі даних протоколу BLE	52
Висновки до розділу	57
3 РОЗРОБКА ТА ТЕСТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ КЕРУВАННЯ КОНТРОЛЕРОМ.....	58
3.1 Вибір та аналіз інструментів розробки.....	58
3.2 Розробка та тестування мобільного додатку	63
Висновки до розділу	67
4 ЕКОНОМІЧНА ЧАСТИНА	69
4.1 Комерційний та технологічний аудит науково-технічної розробки ...	69
4.2 Прогнозування витрат на виконання науково-дослідної (дослідно- конструкторської) роботи.....	73
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	78
Висновки до розділу	84
ВИСНОВКИ.....	85
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	86
ДОДАТКИ.....	90
Додаток А.....	91

Додаток Б	95
Додаток В	101
Додаток Г	107

ВСТУП

Актуальність. Економія природних ресурсів завжди була важливим аспектом для людини. Кожен намагається економити всіма можливими методами, адже від цього залежить, які кошти витратить людина на оплату за користування цими ресурсами.

Мабуть ключовим ресурсом є тепло, особливо в холодні пори року. Людство придумало чимало способів економії на використанні тепла. Найпоширенішим є децентралізована система опалення, яка дозволяє налаштовувати температурний режим під свої бажання. Для обігріву частини приміщення або робочого місця в офісі, як правило використовуються електронагрівальні прилади, які є досить енергозатратними. Для вирішення даної проблеми прийнято рішення у створенні контролера, який буде керувати режимом роботи електронагрівальними приладами. Він буде керуватися та налаштовуватися за допомогою програмного забезпечення.

В даній роботі задача керування контролером буде вирішуватися шляхом створення кросплатформеного мобільного додатку із технологією керування контролером за допомогою протоколу Bluetooth Low Energy.

На даний час уже існують подібні системи, але вони дороговартісні та мають ряд недоліків. Розроблювана система буде позбута цих недоліків та буде коштувати дешевше ніж аналоги.

Об'єктом дослідження є процес дистанційного управління контролером електронагрівального приладу.

Предметом дослідження є методи та технології управління контролером електронагрівального приладу за допомогою протоколу BLE.

Метою дослідження є побудова кросплатформеного мобільного додатку з можливістю керування контролером через протокол Bluetooth Low Energy.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

- дослідження принципу роботи протоколу Bluetooth Low Energy;
- розробка мобільного додатку для управління контролером.

Наукова новизна одержаних результатів:

- побудова мобільного додатку для управління контролером а базі плати Arduino ESP32, через протокол Bluetooth, для керування електронагрівальними приладами.

Практичною цінністю є розробка кросплатформеного додатку для управління контролером, який керує роботою електронагрівальних приладів.

Апробація. Представлені в роботі результати апробовані в результаті участі в конференції Всеукраїнська науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2023)»:

Публікації: Ліщук Андрій Романович, Дубовой Володимир Михайлович «Короткий огляд розробки мобільного додатку для управління децентралізованою системою опалення», «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ», 2022. URL:

1 АНАЛІЗ СИСТЕМ КЕРУВАННЯ ТЕМПЕРАТУРОЮ У ПРИМІЩЕННІ ТА ОГЛЯД АНАЛОГІВ

1.1 Актуальність проблеми управління температурою у приміщенні

На сьогоднішній день проблема опалення помешкання є актуальною як ніколи. Кожна людина намагається оптимізувати використання ресурсів опалення таким чином, щоб зекономити будь яким методом. Для цього існує безліч методів та пристроїв, які допомагають правильно налаштувати використання теплонагрівальних приладів. Наприклад найбільш поширеним пристроями є терморегулятори, які дозволяють легко та швидко налаштувати потрібну температуру в помешканні. Основний недолік таких пристроїв, це їхня дороговизна та складність у встановленні.

Як правило майже усі прилади, які використовуються для управління теплонагрівальними приладами та температурою у помешканні створені для корегування температурного режиму у всьому будинку або ж в окремій кімнаті(у випадку, якщо система опалення - децентралізована). Якщо ж людина хоче налаштувати свій мікроклімат, до прикладу у якійсь частині кімнати, або ж на робочому місці в офісі, то єдиним способом є використання електрообігрівальних приладів. Такі прилади дозволяють створити мікроклімат у місці де вони знаходиться, що є досить зручним зі сторони користувача. Недоліки такого використання, це потреба в постійному нагляді за такими приладами – тобто ручне керування і як наслідок незручність у користуванні та неекономічне використання енергоресурсів, що тягне за собою більшу витрату коштів.

Звичайно ж є деякі обігрівачі із влаштованим блоком віддаленого управління за допомогою технології Wi-Fi, але такі прилади є досить дорогими та нерозповсюдженими, а також використання технології Wi-Fi є недоцільним у деяких місцевостях, а саме там де немає покриття мережі Internet, яка необхідна для роботи такого приладу.

Іншим способом управління є так звані «розумні» розетки, що являють собою розетку, якою можна керувати за допомогою, як правило тої ж технології Wi-Fi, або ж технології Bluetooth. У даного пристрою є та ж проблема при використанні Wi-Fi, що і в обігрівачів із влаштованим блоком віддаленого управління. Якщо ж «розумна» розетка використовує технологію Bluetooth для отримання віддалених команд, то вона також має суттєву проблему, а саме те що користувач буде керувати не температурним режимом, а ввімкненням та вимкненням «розумної» розетки і як наслідок обігрівального елемента, що є недоцільним для налаштування мікроклімату, оскільки неможливо налаштувати потрібну температуру, маючи можливість лише вмикати та вимикати обігрівальний елемент.

Через відсутність альтернатив було прийнято рішення у створенні контролера, який дозволить керувати температурним режимом з використанням технології Bluetooth за допомогою мобільного додатку. Контролер, базується на платі Arduino ESP32, якою можна управляти за допомогою Bluetooth. У платі наявний протокол BLE (Bluetooth Low Energy), що дозволить енергоефективно використовувати контролер у поєднанні з мобільним телефоном. Пристрій являє собою «розумну» розетку із влаштованим датчиком температури повітря, що дозволяє керувати температурним режимом у потоці реального часу. Ця розробка надає можливість гнучкого управління, використовуючи для налаштування як температуру навколишнього середовища так і графік ввімкнень та вимкнень контролера встановленого користувачем. Технологія BLE дозволяє не прив'язуватися до необхідності покриття мережі Internet, що є ще одною суттєвою перевагою.

Ціль даної наукової роботи – розробка кросплатформеного мобільного додатку для керування та налаштування контролера. Обмін інформацією між контролером та мобільним телефоном буде здійснюватися через протокол Bluetooth. Інформація буде містити блоки даних, якими прилади будуть обмінюватися між собою. Основні дані, які будуть передаватися з телефону на

контролер – це дані для управління температурним режимом та його налаштування. В свою чергу інформація про температуру навколишнього середовища, стан та налаштування контролера буде передаватися на мобільний телефон, що дозволить швидко та гнучко налаштувати контролер та отримати інформацію про це. Мобільний додаток дозволить налаштовувати контролер в залежності від побажань користувача, а також отримувати та обробляти інформацію про стан та роботу контролера.

1.2 Аналіз предметної області розробки мобільних додатків

Сьогодні важко собі уявити життя без сучасних гаджетів. Якщо ж комп'ютерна техніка досі є не у всіх, то мобільний телефон, а саме смартфон, є практично у кожної людини. Перевагою смартфона є його мобільність та практичність - ним можна користуватися у будь-якому місці і виконувати різноманітні завдання. Також суттєвою перевагою є те, що смартфон окрім своїх основних функцій – дзвінки та повідомлення, може виконувати безліч інших операцій [1].

Ці операції можуть бути найрізноманітнішими, від пошуку інформації в мережі Інтернет до роботи з навігацією, від спілкування в соціальних мережах до управління пристроями керування різних систем. Використання усіх цих функцій стало можливим завдяки розвитку комп'ютерної техніки та програмування [2].

Розробка додатків на мобільні пристрої, завжди була актуальною, оскільки кількість мобільних пристроїв зростає кожного дня. Усі мобільні пристрої можна поділити на два основних види по операційній системі: Android та iOS [3].

Ці операційні системи суттєво відрізняються одна від одної. Кожна має свої переваги, основні представлені нижче [3, 4].

IOS:

- Чудова продуктивність;

- Високоякісні додатки і магазин додатків;
- Довгострокові оновлення ОС;
- Краща цінність в довгостроковій перспективі.

Android:

- Набагато більший ступінь свободи;
- Настроюваний інтерфейс;
- Велика різноманітність програм;
- Більш доступні телефони в цій категорії.

На кожен операційну систему існує безліч найрізноманітніших додатків, які допоможуть виконати певні завдання та полегшать життя їх користувачу. Розробка таких додатків здійснюється за допомогою спеціально визначених мов програмування [2, 3, 4].

Найпопулярнішою та однією із офіційних мов розробки додатків під операційну систему Android є мова програмування Java. Одна очевидна перевага Java розробників над рештою це те, що API Android дуже схоже на API мови Java, і Android підтримує найважливіші J2SE SDK класи. Ще одна перевага, це те що є можливість використовувати для розробки під Android ті ж інструменти, що і використовує Java. Наприклад, IDE Eclipse, адже Google надає для Eclipse плагін для розробки програм Android. Основні переваги та недоліки мови Java представлені нижче [5, 6].

Переваги:

- Легкість вивчення та розуміння;
- Відсутність прив'язки до платформи;
- Наявність великої стандартної бібліотеки API;
- Наявність безлічі стандартних бібліотек Java;
- Наявність великої опенсорс платформи;
- Хороша підтримка спільноти для допомоги розробникам;
- Легкість та компактність додатків;
- Велика швидкість розробки.

Недоліки:

- Потреба великої кількості пам'яті;
- Громіздкість коду;
- Повільність.

Друга офіційна мова програмування для розробки додатків під систему Android, є Kotlin. Kotlin був спроектований і розроблений компанією JetBrains, чеською компанією, яка відома своєю популярною IDE - IntelliJ IDEA. Kotlin був розроблений для розв'язання деяких проблем на Java. Синтаксис Kotlin простіший, чистіший та призводить до меншого роздування коду. Це дозволяє більше зосередитися на розв'язанні актуальної проблеми, а не на складному синтаксисі. Крім того є можливість використовувати Kotlin і Java разом в одному проекті, і це робить його дійсно потужним. Переваги та недоліки наведені мови Kotlin наведені нижче [5, 6].

Переваги:

- Легкість вивчення після знайомства із мовою Java;
- Лаконічність (менша можливість зробити помилку);
- Гнучкість у роботі з дизайном;
- Можливість використання в бекенд проектах;
- Можливість взаємодії з Java.

Недоліки:

- Складний обчислювальний процес;
- Мала кількість спеціалістів, через молодість мови.

Як і для Android так і для iOS існує офіційна мова розробки додатків і це мова - Objective-C. Мова, створена на початку 80-х років минулого століття, шляхом схрещування C з популярним на той час Smalltalk. Спочатку ця мова сприймалася як проста надбудова, що модифікує деякі синтаксичні конструкції, але після того, як за ліцензування взялася спочатку компанія NeXT, а потім на правах наступника і Apple, Objective-C став однією з найпопулярніших мов для розробки додатків під систему iOS [6, 7].

Переваги:

- Динамічна типізація;

- Розширена документація та спільноти;
- Гнучкість у порівнянні з іншими мовами того часу;
- Простота розуміння синтаксису.

Мінуси:

- Складність прочитання коду та розуміння алгоритмів;
- Динамічна типізація;
- Низька продуктивність;
- Обмежена функціональність.

Іншою офіційною мовою програмування для розробки додатків під операційну систему iOS, віднедавна стала мова Swift. Swift – це відкрита компільована мова програмування, призначена для розробки програм та додатків для iOS/MacOS та інших платформ. З його допомогою можна створювати програмне забезпечення для різних типів пристроїв та обладнання, включаючи смартфони, планшети та десктопи. Основні переваги та недоліки представлені нижче [7].

Переваги:

- Висока продуктивність;
- Простий синтаксис;
- Безпека;
- Вільна ліцензія;
- Зручна робоча частина;
- Сумісність з другою основною мовою програмування на iOS - Objective-C;
- Динамічні бібліотеки.

Недоліки:

- Проблеми сумісності із сторонніми інструментами та IDE;
- Нестача інструментів, через молодість мови.

Як можемо бачити, кожна із цих мов програмування має свої переваги та недоліки. Також кожна із них створена для розробки для конкретної операційної системи і є найкращою у своєму роді. З іншої сторони, те що

Objective-C та Swift створені для розробки під iOS, а Java та Kotlin для розробки під Android, є незручним та не практичним зі сторони розробки додатків під обидві операційні системи із написанням одного коду. На щастя розвиток технологій не стоїть на місці і можливість розробки мобільних додатків одразу для двох операційних систем, тобто написання одного коду, який буде працювати на iOS та Android, стало доступним завдяки кросплатформним мовам програмування. Основні з яких це JavaScript та Python. Обидві мови програмування створені для вирішення певних питів завдань. Наприклад JavaScript орієнтований більше на розробку веб додатків, а Python – на машинне навчання. Однак існує безліч фреймворків, які розширюють можливості та напрямки використання цих мов програмування, що робить їх універсальними. Для розробки на мобільні пристрої також є спеціальні фреймворки. Так для JavaScript – це фреймворк React Native, а для Python – Kivy. Звичайно ж існують інші фреймворки, але наведені вище найпопулярніші та найбільш доступні для розробки. Нижче наведені основні переваги та недоліки наведених фреймворків [8, 9].

React Native

Переваги:

- Зрозумілий розробникам, хто знайомий із фреймворком React;
- Велика кількість JavaScript пакетів з npm;
- Розширена документація та спільноти;
- Велика кількість готових наборів графічного інтерфейсу;
- Зрозумілий інструментарій.

Недоліки:

- Можливість конфлікту між різними пакетами інструментів;
- Неможливість розробки програми без використання нативного коду;

Kivy

Переваги:

- Швидкість розробки, через швидкість та доступність мови Python;
- Велика кількість готових бібліотек;
- Можливість роботи з графікою, без використання нативного коду;
- Використання нативного коду лише, для доступу до деяких функцій телефону;
- Можливість використання Java бібліотек при розробці на Android;
- Доступ до основних функцій телефону без використання нативного коду.

Недоліки:

- Повільний «холодний старт»;
- Особливості роботи зі списками(витрата надлишкової пам'яті).

Як можемо бачити JavaScript та Python мають як переваги так і недоліки, хоча у Python недоліки стосуються специфіки мови, що не є суттєвим. На сьогодні ці дві програмування також все частіше використовуються для розробки мобільних додатків на рівні великих компаній. Гнучкість та кросплатформеність цих мов програмування дозволяє розробляти найрізноманітніші проекти як на iOS так і на Android. Тому при меті розробити додаток одразу для двох операційних систем, при написанні одного коду, слід обирати одну із вищевказаних [8, 9].

1.3 Регулятори опалення та їх види

Від недавніх пір можемо спостерігати тенденцію на зменшення використання природних ресурсів. Ця тенденція зумовлена багатьма факторами, які безпосередньо впливають на рівень та спосіб життя людей. Останнім часом люди намагаються економити всюди де це можливо, від

покупок різних речей до заощадження на використанні природних ресурсів. Основним методом заощадження є скорочення використання ресурсів, а саме зменшення використання електроенергії, води та тепла.

Якщо ж влітку використання ресурсів є не суттєвим, то взимку витрати виростають у декілька разів. Основним ресурсом в холодну пору року є тепло. Особливо коли помешкання людини під'єднане до централізованої системи опалення її комунальні платежі на 60% складаються із оплати за теплоенергію. Вирішенням цієї проблеми є заміна централізованої системи опалення на децентралізовану. Таких систем є безліч, від різноманітних котлів до звичайних обігрівачів. Хоча ці системи допомагають зекономити, але проблема із раціональним і економічним використанням теплоенергії залишається. Саме вирішенням даної проблеми були зацікавлені розробники різних компаній та звичайні люди. За останні роки було придумано та розроблено багато різних систем управління опаленням – так звані регулятори. У більшості випадків це системи керування децентралізованою системою опалення, оскільки вона є економічнішою на відмінну від централізованого опалення, що згадано вище [10].

Контроль опалення – це широкий термін, що охоплює таймери, термостати, сантехніку та електронні компоненти, які допомагають контролювати, коли має бути ввімкнено опалення та яка температура має бути у кімнатах [10, 11].

Елементи керування опаленням варіюються від традиційних механічних стилів, які встановлюються вручну, до елементів керування через Інтернет, які вивчають звички користувача та автоматично регулюють налаштування. Контроль опалення постійно вдосконалюється, допомагаючи контролювати тепло в домівках.

Існує багато різних типів регуляторів опалення. Нижче перераховані деякі загальні типи [10, 11].

- Таймер або програматор - дозволяє контролювати, ввімкнення та вимкнення опалення та гарячої води;

- Механічні кімнатні термостати – автоматичне вмикання та вимикання опалення, при досягненні заданої температури;
- Програмовані кімнатні термостати – контроль часу та температури в одному блоці, що дозволяє встановлювати різні температури для різного часу доби;
- Термостатичні радіаторні вентиля – контроль потоком гарячої води через радіатор, до якого вони встановлені;
- Розумне керування опаленням(система «розумний дім») – керування елементи опалення за допомогою інтернету;
- Термостат рідини – контроль температури рідини у різних варіаціях систем опалення;
- Погодна компенсація – контроль засобами обігріву в залежності від погодних умов;
- Програмовані радіатори – електричні засоби обігріву із можливістю налаштування через інтернет.

1.3.1 Механічні та електричні термостати

Термостат – це пристрій для постійної підтримки температури у приміщенні. Термостат, мабуть, є найважливішим елементом у системі управління теплоенергією, через його основну функцію – контроль. Він дає можливість контролювати, та в окремих випадках взаємодіяти з іншими компонентами, а також отримувати важливу інформацію про стан кожного компонента [12].

Термостати бувають усіх форм і розмірів. Але крім цього, термостати також відрізняються за типом. Є два основних типи — механічний і цифровий. Обидва типи термостатів принципово відрізняються один від одного, особливо принципом роботи [12, 13].

Будь-яка речовина розширюється при нагріванні та стискається при охолодженні. Але, вода є винятком, оскільки вона розширюється як при

нагріванні, так і при замерзанні. Цей принцип називається тепловим розширенням і є принципом роботи механічних термостатів .

Використовуючи теплове розширення, механічні термостати можуть маніпулювати рухом своїх компонентів, щоб завершити електричне коло. Це можна зробити за допомогою біметалічних смуг або сільфонів, наповнених газом, які діють як міст, через який може проходити електрика. Механічні термостати поділяються на кілька видів в залежності від способу активації. Основні види, це активація за допомогою біметалічних стрічок, активація з використанням газонаповнених сільфонів та активація за допомогою воску [13, 14].

1) Термостати на основі біметалічних стрічок

Термостати можуть використовувати два різні шматки металу, як правило, латунь і залізо, які з'єднані болтами для створення біметалічної стрічки. Вона діє як міст між двома кінцями ланцюга. Електрика проходить через стрічку, коли ланцюг завершений, що відбувається, коли «міст опускається» [14, 15].

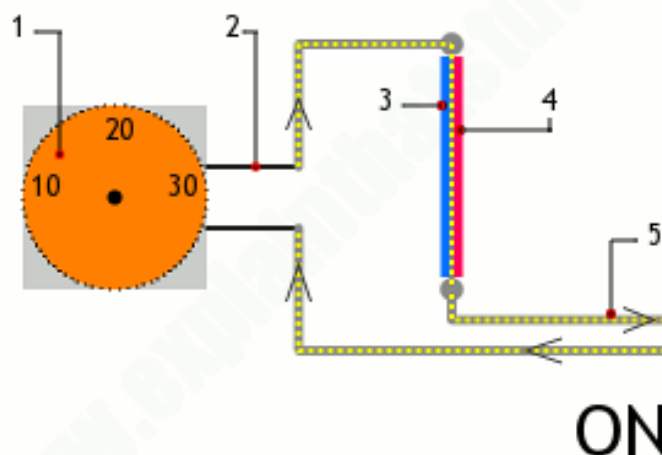


Рисунок 1.1 – Активний стан механічного термостата побудованого на основі біметалічної стрічки

Коли електрика проходить через смужки, ланцюг генерує тепло, що спричиняє розширення металевих смужок. Це, у свою чергу, «піднімає міст» і знову розмикає ланцюг. Через це електрика не може проходити, що також відключає тепло та охолоджує кімнату [15].

Але оскільки тепло відключається від системи, металеві стрічки також починають охолоджуватися. Вони повертаються до своєї первісної форми, що знову замикає ланцюг для вироблення тепла.

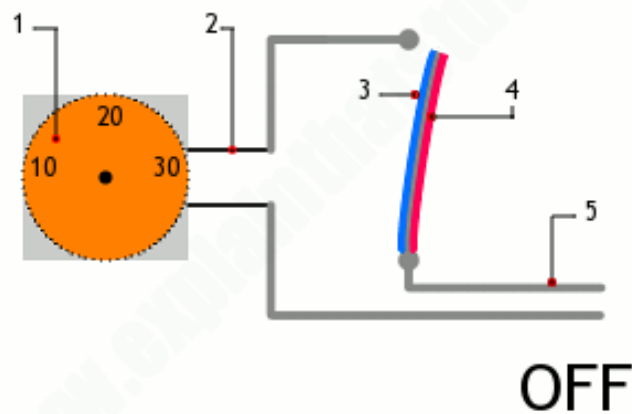


Рисунок 1.2 – Неактивний стан механічного термостата побудованого на основі біметалічної стрічки

Так працюють автоматичні механічні термостати з біметалевими планками. Налаштування циферблата змінить температуру, яка визначає, коли міст піднімається та опускається назад. Але оскільки металам потрібно занадто багато часу, щоб згинатися та виходити з контуру, також знадобиться деякий час, перш ніж термостат відреагує на зміни температури.

2) Термостати на основі газонаповнених сильфонів

Оскільки біметалічні стрічки розширюються та звужуються надто довго, була введена альтернативна конструкція термостата, щоб система легше реагувала на зміни температури. Цей новий дизайн позбавляє

металевих смуг і замість них використовує металеві диски. Використання металевих дисків збільшує площу реагуючої поверхні. Вони також мають виступи, щоб зробити їх еластичними та гнучкими [15, 16].

Між двома металевими дисками лежать заповнені газом сільфони. Газ всередині цих сільфонів має здатність негайно реагувати на зміни температури. Наприклад, коли в кімнаті тепло, газ розширюється, збільшуючи відстань між дисками. І навпаки, коли кімната знову охолоджується, газ стискається, повертаючи металеві диски разом [17].

Постійне перемикання розмикає та замикає ланцюг, коли внутрішній диск натискає на мікроперемикач, розташований у середині термостата.



Рисунок 1.3 – Механічний термостат побудований на основі газонаповнених сільфонів

3) Воскові термостати

Деякі термостати використовують зміну стану речовини з рідини на газ. Мабуть, найпоширенішим прикладом є воскові термостати, найчастіше вони використовуються у вентилях домашніх радіаторів, двигунах автомобілів і душових змішувачах. Вони використовують невелику пробку воску всередині герметичної камери. Під час зміни температури віск плавиться (змінює стан із твердого на рідкий), сильно розширюється та виштовхує з камери стрижень, який щось вмикає чи

вимикає (керує системою охолодження двигуна в автомобілі чи регулює суміш гарячої та холодної води в душі). Воскові термостати, як правило, більш надійні та довше працюють у екстремальних умовах, наприклад у двигуні автомобіля [15, 17, 18].

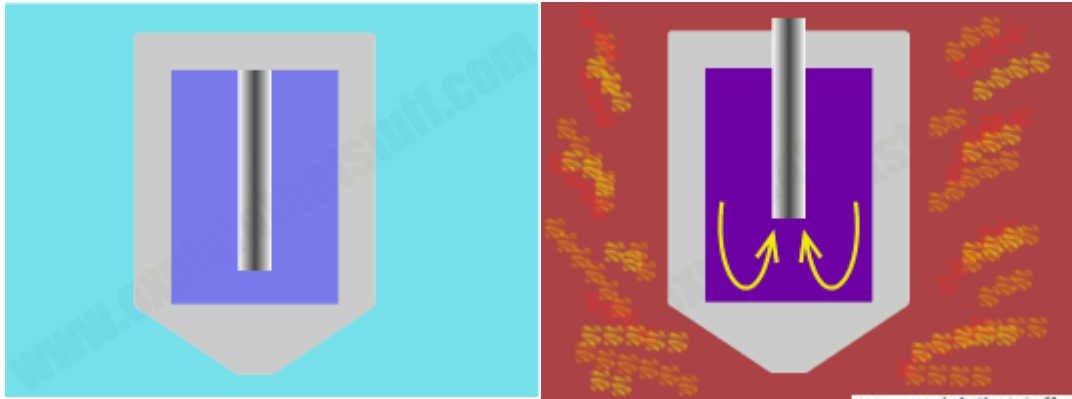


Рисунок 1.4 – Принцип роботи воскового термостату

Механічні термостати використовують внутрішні механічні елементи та термодинамічні принципи, щоб сприймати тепло та запускати нагрівання й охолодження на основі відчуття тепла. Електронні термостати є більш прямими завдяки використанню термодатчиків і схем [17, 18].

Цифрові термостати використовують термістори для вимірювання температури навколишнього середовища в кімнаті. Це тип резистора, опір якого змінюється залежно від температури. Термостат містить мікроконтролер, який вимірює цей опір, який потім перетворює вимірне значення в фактичне показання температури [17].

На відміну від механічних термостатів, цифрові термостати працюють як невеликий комп'ютер. Він має інтерфейс користувача, який надає корисну інформацію, таку як температура та поточні налаштування, і програмується на основі вподобань користувача [16].

В останні роки також були представлені розумні цифрові термостати. Розумні термостати мають вбудовані можливості навчання, які дозволяють пристрою вивчати вподобання користувача та поведінку щодо температури в

приміщенні. Це дозволяє пристрою автоматично встановлювати температуру залежно від часу доби та температури на вулиці. Наприклад, влітку термостат може автоматично вмикати кондиціонер вдень і вимикати його вночі, покращуючи енергоефективність.

Механічні термостати є найдешевшими, але вони також базові та не такі точні, коли вони показують температуру в будинку. На відміну від механічних термостатів, цифрові термостати дорожчі та точніші. Крім того, цифрові термостати можуть мати функції, яких немає в механічних термостатах. Визначною особливістю деяких цифрових термостатів є їхня можливість програмування, що є більш енергоефективним. Нижче наведені основні переваги та недоліки механічного та цифрового термостату[15, 17].

Таблиця 1.1

Порівняння переваг та недоліків механічного та цифрового термостатів

	Переваги	Недоліки
Механічний	Дешевизна	Низький ступінь контролю
	Стабільність роботи при перепадах напруги	Не енергоефективність
	Простота у використанні	Низька швидкість реагування на перепад температури
		Можливість вмісту шкідливих речовин
Цифровий	Високий ступінь контролю	Дорожчий у порівнянні з механічним
	Мінімальне обслуговування	Більша складність встановлення
	Можливість до програмування	
	Простота у використанні	
	Енергоефективність	
	Миттєва швидкість реагування на перепад температури	

1.3.2 Система «розумний будинок»

Найбільш популярна та поширена система управління децентралізованим опаленням, є так званий комплекс – «розумний будинок». «Розумний будинок» - це зручна домашня установка, де приладами та пристроями можна автоматично дистанційно керувати з будь-якого місця, де є підключення до інтернету, за допомогою мобільного або іншого мережевого пристрою. Пристрої в розумному домі з'єднані між собою через інтернет, що дозволяє користувачеві дистанційно контролювати такі функції, як охоронний доступ до дому, температура, освітлення та інші прилади в залежності від конфігурації [20, 21].



Рисунок 1.5 – Приклад системи «розумний будинок»

Пристрої розумного будинку з'єднані один з одним і доступ до них можливий через одну центральну точку — смартфон, планшет або ноутбук. За допомогою однієї системи домашньої автоматизації можна керувати дверними

замками, телевізорами, термостатами, домашніми моніторами, камерами, освітленням і навіть такими приладами, як холодильник. Система встановлюється на мобільному або іншому мережевому пристрої, і користувач може створювати часові розклади для вступу в силу певних змін.

Розумна побутова техніка має навички самонавчання, щоб вони могли вивчати розклад власника будинку та вносити корективи за потреби. Розумні будинки з керуванням освітленням дозволяють власникам будинків зменшити споживання електроенергії та отримати вигоду від економії витрат на енергію. Також однією із основних функцій є керування системою обігріву, що дозволяє раціонально використовувати теплоенергію у домі [20, 22].

Розумні будинки можуть містити бездротові або дротові системи або обидві. Бездротові системи легше встановити. Встановлення бездротової системи домашньої автоматизації з такими функціями, як розумне освітлення, клімат-контроль і безпека, може коштувати кілька тисяч доларів, що робить її дуже економічною.

Недоліком бездротових систем є те, що для такої системи, потрібне потужне покриття Wi-Fi і широкосмугове обслуговування в усьому будинку. Бездротові системи розумного дому, як правило, більш підходять для невеликих будинків або орендованих приміщень через їх менший розмір [20, 23].

Жорсткі системи, з іншого боку, вважаються більш надійними, і зазвичай їх складніше зламати. Крім того, жорсткі системи розумного будинку можна легко масштабувати; тому це часто є методом за замовчуванням при проектуванні нової будівлі або виконанні капітального ремонту [24].

Суттєвий недолік – це те, що жорстка система досить дорога. Встановлення розкішної та жорсткої інтелектуальної системи може коштувати власникам будинків десятки тисяч доларів. Крім того, має бути місце для мережевого обладнання, включаючи кабелі Ethernet.

Пристрої «розумного будинку» дозволяють краще контролювати опалювальні пристрої, зокрема коли продукти вмикаються, вимикаються та

контролюються. Розумні пристрої можуть бути оснащені датчиками температури або вологості, які автоматично вмикаються або вимикаються, якщо відповідають певним критеріям. Ця лінійка інновацій для розумного будинку також поширюється на кондиціонери. Основним приладом керування опаленням та температурою загалом в «розумному будинку» є термостат [24].

Розумні термостати — тенденція, що зростає, допомагаючи власникам будинків заощаджувати витрати на електроенергію та стати справді корисним з використанням опалення та охолодження. Навіть у старих будинках процес встановлення простий і зрозумілий.

Більшість розумних термостатів використовують Wi-Fi для підключення до програми на вашому телефоні. За допомогою програми ви можете планувати, коли вмикається опалення чи охолодження, регулювати температуру помешкання, де б ви не знаходилися, і навіть налаштовувати автоматизацію за допомогою інших розумних пристроїв [25, 26].



Рисунок 1.6 – «Розумний термостат»

Деякі з них є програмованими термостатами, які не підключаються до програми, але можуть планувати процедури нагріву та охолодження через сенсорний дисплей [26].

Розумний термостат полегшує життя. Ви можете запланувати, щоб термостат увімкнув тепло вранці, щоб позбутися холоду. Або ви можете ввімкнути кондиціонер лише за пару годин до того, як прийдете додому з роботи [26].

Якщо на термостаті є програма, з телефону можна змінювати температуру. Це може бути особливо корисно для тих, хто має проблеми з пересуванням.

Розумні термостати можуть бути пристроями безпеки, якщо їх під'єднати до пристрою прослуховування пожежі або розумної димової сигналізації. Під час пожежі термостат автоматично вимикає систему опалення, вентиляції та кондиціонування повітря, щоб запобігти посиленню пожежі через обдування повітрям [27].

Розумні термостати можуть зробити багато корисних речей, яких не може традиційний термостат. Вивчення свого розкладу може заощадити гроші на опаленні та охолодженні, оскільки система опалення, вентиляції та кондиціонування, тепловий насос або піч не працюватимуть, коли вони не потрібні [27, 28].

Різноманітність розумних термостатів дозволяє регулювати температуру за допомогою голосових команд для Amazon Alexa , Google Assistant і Apple Siri. Тож навіть якщо ви не можете знайти свій телефон, до комфорту можна дійти лише одним криком [28].

Багато розумних термостатів також допомагають відстежувати використання енергії. Деякі навіть мають попередні налаштування енергозбереження, які також можуть заощадити ваші гроші на рахунку за електроенергію, керуючи вашим розумним опаленням і охолодженням найбільш рентабельним способом [28, 29].

Основні функції розумних термостатів:

- Вивчення розкладу і внесення змін;

- Підключення до мобільного додатку на своєму телефоні, використання геозонування та внесення налаштувань, при наближенні до дому;
- Слідкування за енергоспоживанням і можливість виявити проблему з системою;
- Встановлення нагадування для планування щорічного технічного обслуговування та заміни повітряних фільтрів;
- Інтеграція розумних колонок у домі, якщо вони є.

Налаштування в поєднанні з постійною аналітикою робить «розумні термостати» надзвичайно корисними.

Встановлення технологічної системи «розумний дім» забезпечує власникам житла зручність. Замість того, щоб керувати приладами, термостатами, освітленням та іншими функціями за допомогою різних пристроїв, домовласники можуть керувати ними всіма за допомогою одного пристрою — зазвичай смартфона або планшета [29, 30, 31].

Оскільки вони підключені до портативного пристрою, користувачі можуть отримувати сповіщення та оновлення щодо проблем у своїх домівках.

Стосовно вартості налаштування розумної системи власники будинків можуть отримати вигоду від значної економії. Прилади та електроніку можна використовувати ефективніше, зменшуючи витрати на електроенергію.

Незважаючи на те, що «розумний дім» пропонує зручність і економію коштів, все ще існують проблеми. Ризики безпеки та помилки продовжують мучити виробників і користувачів технології. Досвідчені хакери, наприклад, можуть отримати доступ до пристроїв розумного дому з підтримкою інтернету [30, 31].

Заходи для зменшення ризиків таких атак включають захист інтелектуальних пристроїв і пристроїв за допомогою надійного пароля, використання шифрування, якщо воно доступне, і підключення до мережі лише надійних пристроїв [33].

Як зазначалося вище, витрати на встановлення інтелектуальної технології можуть становити від кількох тисяч доларів для бездротової системи до десятків тисяч доларів для жорсткої системи. Це висока ціна, яку може собі дозволити далеко не кожен [32].

1.3.3 GSM розетки та GSM блок-модулі

GSM-розетки будуть корисні тим, у кого невеликий будинок, де одна або дві кімнати, і кількість конвекторів потужністю не більше 3-3,5 кВт – допустиме навантаження розетки. Деякі нові моделі дозволяють підключати електроприлади із піковим навантаженням до 4 кВт. Дана розетка дуже проста та зручна у використанні. Достатньо підключити всі обігрівачі до неї та включити прилад у звичайну настінну розетку. У розетці GSM датчик температури вимірює температуру безпосередньо поруч з нею. Тому, якщо є потреба в обігріві багатьох різних кімнат, і всі обігрівачі заведені в одну розетку, то якщо в якійсь кімнаті вже нагріється потрібна температура, а в іншій не встигне, а датчик температури спрацює на відключення приладів, або ще продовжуватиме гріти, поки в тій кімнаті, де встановлена розетка, не встановиться потрібна температура. У GSM-розетках немає акумулятора, і при відключенні електроживлення вона проінформує про відключення, час автономної роботи у неї не більше 3 хвилин, і після цього налаштування скинуться і вона проінформує про включення електроживлення [34, 35].



Рисунок 1.7 – GSM розетка

GSM блок-модуль корисний тим, хто має великий будинок і багато кімнат. Він керує всіма конвекторами у будинку через контактор або силові реле та релейні плати розширення. Для його встановлення потрібні монтажні роботи. У GSM блоці, як і GSM розетці є місце для встановлення SIM-карти. SIM-карту слід вибрати того оператора, який має впевнений прийом у тому місці, де буде встановлено блок. Керується також GSM блок-модуль як і GSM розетка – через SMS-команду або мобільний додаток. Також GSM-модуль оснащується акумулятором, що дозволяє автономно працювати до 12 годин (корисно при частих відключеннях електрики)[34, 35, 36].



Рисунок 1.8 - GSM блок-модуль

Можливості GSM-модуля:

- Підключення до 5 або більше датчиків температури (за допомогою контактора);
- Віддалене керування ввімкненням та вимкненням опалювального обладнання за заданою користувачем програмою;
- Можливість зміни налаштувань терморегулювання у всьому будинку або окремих кімнатах;
- Підтримка температури;
- Контроль наявності напруги мережі 220В;
- Резервний акумулятор;
- Входи для охоронних датчиків та аварійних сигналів;
- Автоматичні звіти про температуру в будинку;
- Контроль справності інженерного обладнання та екстрене сповіщення по SMS про вихід з ладу.

Схема реалізації віддаленого керування конвекторами за допомогою контактора до GSM блок-модуля представлена нижче.

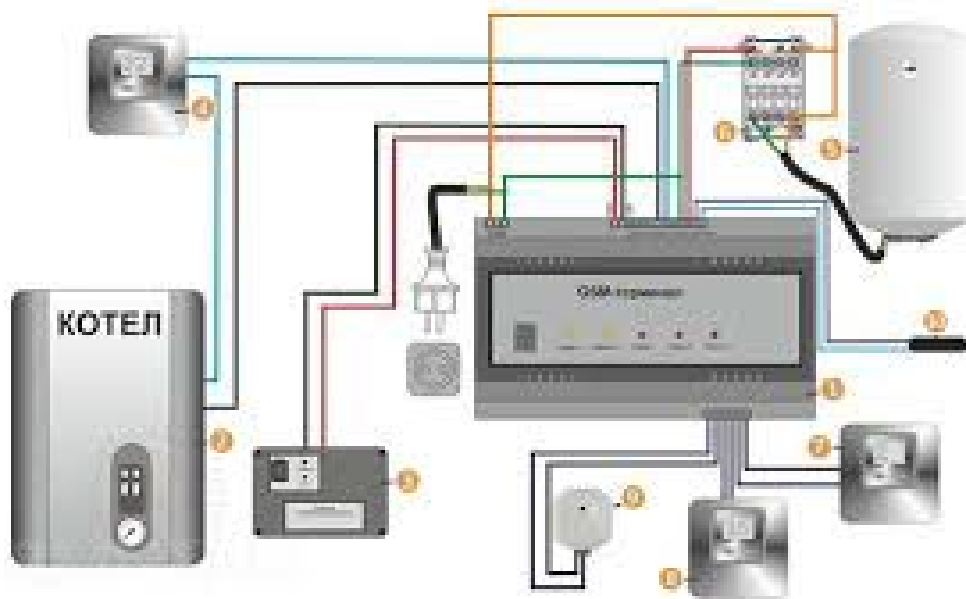


Рисунок 1.9 - Схема реалізації віддаленого керування конвекторами за допомогою контактора до GSM блок-модуля

Конвектори з'єднуються між собою, далі об'єднуються в єдиний центр - електронний блок управління. І через нього віддалено регулюється температура [36].

Після монтажу можна настроїти GSM прилад за допомогою мобільного додатку та встановити автоматичний режим підтримки температури. На телефоні зручно заздалегідь зберегти шаблони SMS-команд, які часто використовуються [37].

Слід зазначити, що вартість даних девайсів порівняно вища, ніж у програмованих та електромеханічних терморегуляторів. І якщо у будинку проживають цілий рік, або будинок використовується постійно, то необхідності віддалено керувати опаленням немає, і можна заощадити на опаленні будинку звичайними повітряними терморегуляторами, які відключатимуть конвектори при досягненні потрібної температури [36, 37].

Як можемо бачити у розглянутих систем управління децентралізованим опаленням є як позитивні так і негативні сторони. Кожна із систем корисна та працездатна. Для їх порівняння можемо виділити кілька основних критеріїв,

які грають важливу роль при виборі системи управління опаленням. Найважливіші параметри при оцінці такі: коштовність системи, зручність використання, складність установки, захищеність від зовнішніх факторів та ефективність(економічність). У таблиці нижче представлені розглянуті вище системи управління опаленням та їх оцінка по основних критеріях по 5-ти бальній шкалі.

Таблиця 1.2 - Порівняння розглянутих систем управління децентралізованим опаленням по основних критеріях

	Дешевизна	Зручність	Простота установки	Захищеність	Ефективність
Механічний термостат	5	2	4	4	2
Цифровий термостат	4	3	4	4	3
«Розумний термостат»	1	5	1	2	5
GSM розетка	3	3	5	3	3
GSM блок-модуль	2	3	2	3	3

Висновки до розділу

Як можемо бачити кожна система має як сильні так і слабкі сторони. Кожна система краща за іншу у певних випадках, і в залежності, за яких умов та де буде розташована система управління опаленням, людина обере одну із визначених. Оскільки кожна із цих систем створена для використання у певних умовах і вона буде найбільш ефективною при дотриманні цих умов.

2 АНАЛІЗ ТЕХНОЛОГІЙ ТА ФУНКЦІОНАЛЬНОСТІ МОБІЛЬНОГО ДОДАТКУ

2.1 Функціональність мобільного додатку

Як і кожен мобільний додаток – розроблений повинен містити конкретну функціональність. Основною функцією, яку повинен виконувати даний мобільний додаток, це передача даних – температури, на контролер. Температура повинна відповідати встановленим межах, для цього в додатку буде створена валідація – перевірка введених даних. Якщо введена температура буде поза межами дозволеної, то відправка інформації на контролер буде неможливою, а також буде виведене повідомлення про неправильно вказану температуру. Якщо ж температура знаходиться у межах дозволеного, то користувач зможе відправити інструкцію для налаштування контролера. Точність температури буде в межах 0.5°C .

Встановлення графіку роботи контролера буде здійснюватися за таким же алгоритмом. Тобто якщо вказаний графік знаходиться в межах дозволеного, то користувач зможе його відправити на обробку до контролера, інакше користувачу доведеться змінити налаштування графіку до дозволених. Графік можна буде встановлювати на кілька днів, що дозволить гнучко налаштовувати температурний режим.

Ще одною важливою функцією є можливість переглядати температурний режим за кілька днів у вигляді графіків. На графіку буде зображена залежність температури від часу. Аналіз цього графіку допоможе користувачу налаштувати температурний режим найкращим чином, оскільки буде доступна інформація про температуру кожної години на протязі кількох днів. Графік буде автоматично оновлюватися, а попередні дні будуть видалятися.

Усі дані, які передаються на контролер, будуть перевірятися за допомогою алгоритму контрольних сум, що дозволить не допустити передачу неправильних або пошкоджених даних. Якщо ж під час передачі даних, або ж

під час режиму очікування – режим протоколу BLE, під час якого дані не передаються, з'єднання за якихось причин було розірване, то буде зроблено кілька спроб з'єднання перш ніж у додатку з'явиться інформація про проблему підключення.

На рисунку 2.1 зображена блок схема порядку дій програмою, при різних умовах.

Як можемо бачити програмою передбачені найважливіші винятки та проблеми, які можуть виникнути під час роботи програми. Таким чином можна передбачити та налаштувати дії програми у відповідь на ту чи іншу непередбачувану ситуацію. В ході розробки та вдосконалення додатку будуть додаватися нові функції для найзручнішого використання користувачем та запобігання неочікуваної поведінки додатку.

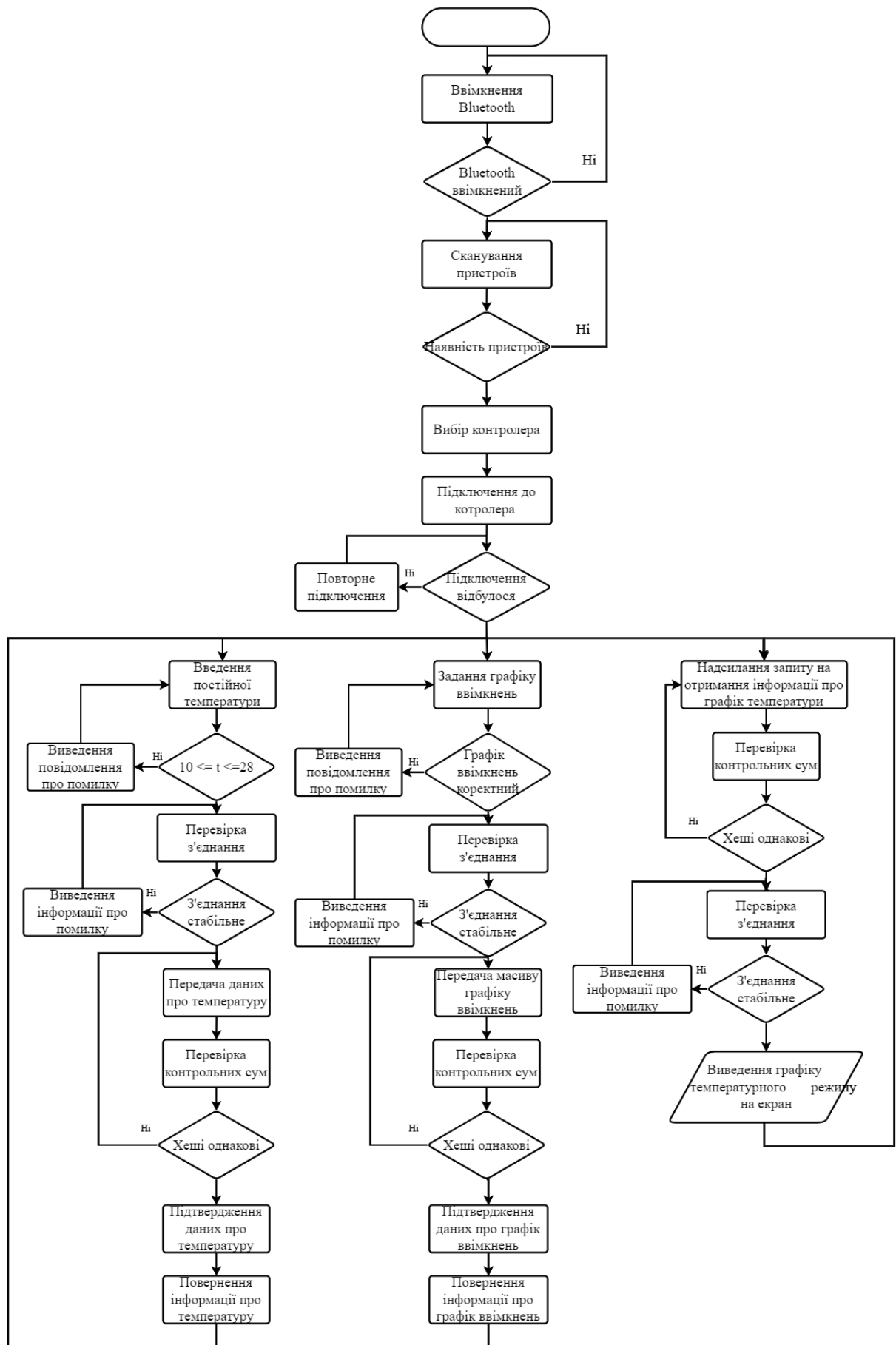


Рисунок 2.1 - Блок схема порядку дій

2.2 Принцип роботи Bluetooth Low Energy

Інтернет речей (IoT) - це система взаємопов'язаних обчислювальних пристроїв, механічних і цифрових машин, об'єктів та людей, яким надано унікальні ідентифікатори (UID) і здатність передавати дані через мережу, не вимагаючи від людини взаємодії з комп'ютером. У всьому світі існує понад 10 мільярдів активних пристроїв IoT, і однією з ключових технологій, що сприяють зростанню цього ринку, є Bluetooth Low Energy (BLE).

Цей гнучкий протокол забезпечує бездротове з'єднання для багатьох найважливіших програм IoT. Його орієнтація на роботу з наднизьким енергоспоживанням дозволяє використовувати його в таких галузях, як побутова електроніка, охорона здоров'я та логістика.

Bluetooth — це бездротова технологія малого радіусу дії, яка працює в неліцензованому ISM 2,4 ГГц для передачі даних і створення персональних мереж (PANs).

Bluetooth - це відносно складний технологічний стандарт, яким керує група, відома як Bluetooth Special Interest Group (SIG). Протягом багатьох років Bluetooth SIG випускала різні версії специфікації Bluetooth. Найпопулярнішими протоколами в специфікації є Bluetooth Classic і Bluetooth Low Energy.

До 2010 року основна увага приділялася вдосконаленню Bluetooth Classic. Але в 2010 році Bluetooth SIG випустила версію 4.0. Ця версія пішла іншим напрямком, ніж попередні версії, додавши новий протокол: Bluetooth Low Energy (BLE).

Незважаючи на те, що BLE має багато спільного зі своїм попередником Bluetooth Classic, Bluetooth Low Energy розглядається як інша технологія, яка спеціально націлена на ринки, де попит на наднизьку потужність, а не на високу пропускну здатність.

Передача даних із радіостанцією Low Energy відбувається короткими пакетами, які не обов'язково мають відправлятися дуже часто. Типовий варіант використання BLE включав би періодичне вмикання каналу радіо, передачу або отримання кількох байтів або кілобайтів даних, а потім вимикання та повернення в режим сну.

Це контрастує з випадками використання Bluetooth Classic, коли радіо не вимикається, як-от підтримка з'єднання протягом тривалого періоду часу, щоб гарантувати найменшу затримку під час вхідного дзвінка, або випадки використання, як-от обмін великим файлом.

Незважаючи на те, що протоколи Bluetooth Classic і Bluetooth Low Energy мають багато подібності, наприклад, вони є протоколами, охопленими специфікацією Bluetooth і обидва працюють у діапазоні ISM 2,4 ГГц, це два різні, несумісні протоколи.

Bluetooth Classic призначений для обробки великої кількості даних, але він також досить суттєво споживає енергію. Передача музики з телефону на навушники через Bluetooth - це традиційний варіант використання Bluetooth Classic.

З іншого боку BLE, розроблений для додатків, які не вимагають обробки великої кількості даних, але вимагають дійсно тривалого часу автономної роботи. Датчик на складі з контрольованою температурою, який можна встановити та забути на місяці чи навіть роки, є типовим прикладом використання протоколу BLE.

Хоча в обох протоколах є більше технічних нюансів, на високому рівні різниця між Bluetooth Classic і Bluetooth Low Energy зводиться до відмінностей у споживанні енергії та пропускній здатності даних.

Specifications	Classic Bluetooth	Bluetooth Low Energy (BLE)
Range	100 m	Greater than 100 m
Data Rate	1-3 Mbps	1 Mbps
Application Throughput	0.7 -2.1 Mbps	0.27 Mbps
Frequency	2.4 GHz	2.4 GHz
Security	56/128-bit	128-bit AES with Counter Mode CBC-MAC
Robustness	Adaptive fast frequency hopping, FEC, fast ASK	24-bit CRC, 32-bit Message Integrity Check
Latency	100 ms	6 ms
Time Lag	100 ms	3 ms
Voice Capable	Yes	No
Network Topology	Star	Star
Power Consumption	1 W	0.01 to 0.5 W
Peak Current Consumption	less than 30 mA	less than 15 mA

Рисунок 2.2 – Основні відмінності між Bluetooth Classic та Bluetooth Low Energy

Як можемо бачити на рисунку 2.2 суттєва різниця між Bluetooth Classic та Bluetooth Low Energy тільки у частоті обміном інформації та споживанням енергії. BLE – має більшу частоту ввімкнення та вимкнення, а також менше споживання енергії на відмінну від Bluetooth Classic.

BLE було розроблено з урахуванням особливостей ринку пристроїв, які вимагають низької потужності та невеликої передачі даних. Щоб задовольнити цей попит, Bluetooth Low Energy забезпечує низку переваг як для розробників, так і для користувачів пристроями BLE.

Основні переваги BLE:

- 1) Низьке енергоспоживання:

BLE забезпечує оптимізоване та низьке енергоспоживання, тримаючи радіоканал якомога довше вимкненим і надсилаючи невеликі обсяги даних на низькій швидкості передачі.

2) Низька вартість:

Модулі та чіпсети BLE мають низьку вартість порівняно з іншими подібними технологіями.

3) Відкритий доступ до документації:

Для більшості інших бездротових протоколів і технологій розробник повинен стати членом офіційної групи або консорціуму для цього стандарту, щоб отримати доступ до специфікації. Стати членом цих груп може коштувати значну суму (до тисяч доларів на рік). З BLE документи специфікації основної версії (4.0, 4.1, 4.2, 5.0, 5.1, 5.2, 5.3 тощо) доступні для безкоштовного завантаження з офіційного веб-сайту Bluetooth.

4) Поширеність у смартфонах:

Це найбільша перевага BLE перед своїми конкурентами, такими як ZigBee, Z-Wave і Thread. Переважна більшість людей у світі володіють смартфонами, і майже всі ці смартфони мають апаратне забезпечення BLE. Це дає розробникам набагато більшу потенційну базу користувачів для їхніх програм.

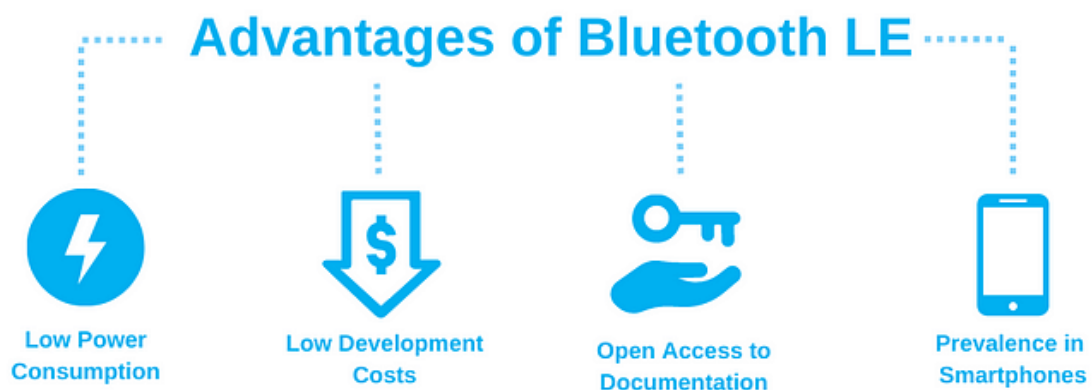


Рисунок 2.3 - Основні переваги протоколу Bluetooth Low Energy

Для розробника розуміння обмежень певної технології так само важливо, як і розуміння її переваг. Таким чином можна визначити, чи підходить певна технологія для розроблюваної програми чи ні.

Основні обмеження BLE:

1) Пропускна здатність даних:

Пропускна здатність даних BLE обмежена фізичною швидкістю передачі даних, тобто швидкістю, з якою радіо передає дані. Ця швидкість залежить від версії Bluetooth, що використовується.

Для версій Bluetooth, раніших за 5.0, швидкість фіксована на рівні 1 Мбіт/с. Проте для Bluetooth 5.0 і пізніших версій швидкість змінюється залежно від використовуваного режиму та РНУ. Швидкість може становити 1 Мбіт/с, як у попередніх версіях, або 2 Мбіт/с, якщо використовується функція високої швидкості.

У разі передачі даних на великий радіус, дії швидкість падає до 500 або 125 Кбіт/с. На прикладному рівні та для кінцевого користувача швидкість передачі даних набагато нижча, ніж швидкість радіопередачі даних.

2) Діапазон:

BLE був розроблений для застосування малої дальності; отже, його діапазон дії обмежений. Є кілька факторів, які обмежують діапазон BLE, зокрема:

- BLE працює в діапазоні ISM 2,4 ГГц, на який сильно впливають перешкоди, які існують навколо нас, як-от металеві предмети, стіни та вода (особливо людські тіла);
- Продуктивність і конструкція антени пристрою BLE;
- Фізичний корпус пристрою Bluetooth Low Energy впливає на продуктивність антени, особливо якщо це внутрішня антена;
- Орієнтація пристрою – фактично пов'язана з розташуванням антени (наприклад, у смартфонах).

3) Вимоги до шлюзу для підключення до Інтернету:

Щоб передати дані з пристрою, що підтримує лише BLE, до Інтернету, потрібен інший пристрій BLE, який має IP-з'єднання, щоб отримати ці дані, а потім, у свою чергу, передавати їх на інший IP-пристрій (або в Інтернет).



Рисунок 2.4 – Основні обмеження протоколу Bluetooth Low Energy

Коли був випущений протокол BLE, він в основному зосереджувався на програмах Інтернету речей (IoT), де невеликі обсяги даних передаються на менших швидкостях. За останнє десятиліття BLE використовувався в різних додатках, від споживчих пристроїв до промислового виробництва. Найпоширеніші напрямки програм використання BLE:

- Домашня автоматизація:

Велика частина ринку домашньої автоматизації частково підтримується використанням Bluetooth Low Energy. Технологія BLE дозволяє використовувати такі пристрої, як розумні світильники, розумні термостати, розумні замки та датчики, які виявляють дим або відчинене вікно.

- Відстеження фізичної активності:

На сьогодні популярними є розумні годинники або годинники для фітнесу, які відстежують пульс, кількість кроків, тощо. Ці пристрої

відстеження здоров'я та активності взаємодіють із програмами на наших телефонах за допомогою BLE. Передача коротких пакетів даних на близькій відстані є ідеальною задачею для технології BLE.

- Аудіопристрої:

Зазвичай Bluetooth навушники використовують Bluetooth Classic, але однією з останніх розробок Bluetooth Low Energy є Bluetooth Low Energy Audio. Цей новий стандарт пропонує кілька переваг порівняно з традиційним Bluetooth, зокрема покращену якість аудіо, менше енергоспоживання та підтримку слухових апаратів.

- Відстеження контактів:

Розумні системи відстеження контактів починають з'являтися, щоб допомогти запобігти поширенню інфекційних захворювань. Замість того, щоб вручну повідомляти дані, які можна використовувати для ідентифікації тих, хто був у тісному контакті з інфікованою людиною, відстеження контактів із підтримкою BLE дозволяє безперервно сканувати «теги» BLE або смартфони для анонімного відстеження точок контакту з людьми.

- Теги для пошуку речей:

Останнім часом зросла кількість подорожей і як наслідок плутанина з багажем, тому деякі люди додають AirTags або Tiles до свого багажу на випадок, якщо він загубиться. Однак ці пристрої точного відстеження з підтримкою BLE можна використовувати не лише для багажу. Можна поставити його на будь-що, що можна втратити, як-от велосипед чи ключі від машини, і відстежити його місцезнаходження за допомогою програми для смартфона.

- Цільова реклама:

Деякі магазини використовують рекламні push-повідомлення для інформування клієнтів, які відвідують їх магазин, про акційні пропозиції. Це завдяки вдосконаленням Bluetooth 5.0 разом із технологією трансляції Bluetooth Beacons.

- Управління запасами:

Керівники складів починають використовувати рішення BLE для відстеження температури та вологості в вантажах, які мають важливе значення, виявлення падінь крихких предметів і навіть планування найбільш оптимальних місць зберігання активів на складі. У цій сфері є значний потенціал для поєднання штучного інтелекту та BLE для створення справді розумного логістичного ланцюжка.

Хоча технологія IoT(Інтернет речей) існує вже деякий час, останніми роками вона набуває серйозного поширення. Технологія IoT допомагає підключати такі речі, як термостати, розумні світильники, людей із вбудованими медичними пристроями, тварин із імплантованими трекерами та автомобілі з датчиками допомоги водієві. Будь-яка людина, тварина або створений людиною об'єкт може стати «рiччю» в IoT. Це означає, що IoT по суті об'єднує людей, процеси та системи.

Щоб зробити всі ці з'єднання можливими, IoT покладається на бездротові технології. ZigBee, Bluetooth Classic і WiFi , які можна використовувати для бездротового підключення пристроїв. Однак BLE часто розглядається як найбільш оптимальна технологія для додатків Інтернету речей через дві основні причини:

- Низьке енергоспоживання. Багато пристроїв IoT живляться від батареї і повинні працювати в польових умовах дуже довго.
- Тип даних, якими обмінюються. BLE оптимізовано для передачі невеликої кількості даних. Це чудово працює для пристроїв Інтернету речей, таких як датчики, яким просто потрібно передавати дані про стан.

Bluetooth Low Energy — це протокол, який як і будь-який інший діє як збірник правил, яких пристрої повинні дотримуватися, щоб спілкуватися чи обмінюватися інформацією.

Важливо розуміти, що загальні функції протоколу розбиваються на менші протоколи, кожен з яких відповідає за спеціалізовану роль. Ці менші

протоколи упаковані в багаторівневу архітектуру, яка називається стеком протоколів або набором протоколів.

Стек протоколу Bluetooth Low Energy який зображений на рисунку 2.5, загалом розбивається на три основні компоненти або підсистеми: додаток, хост і контролер. У середині кожного з цих блоків є окремі шари.

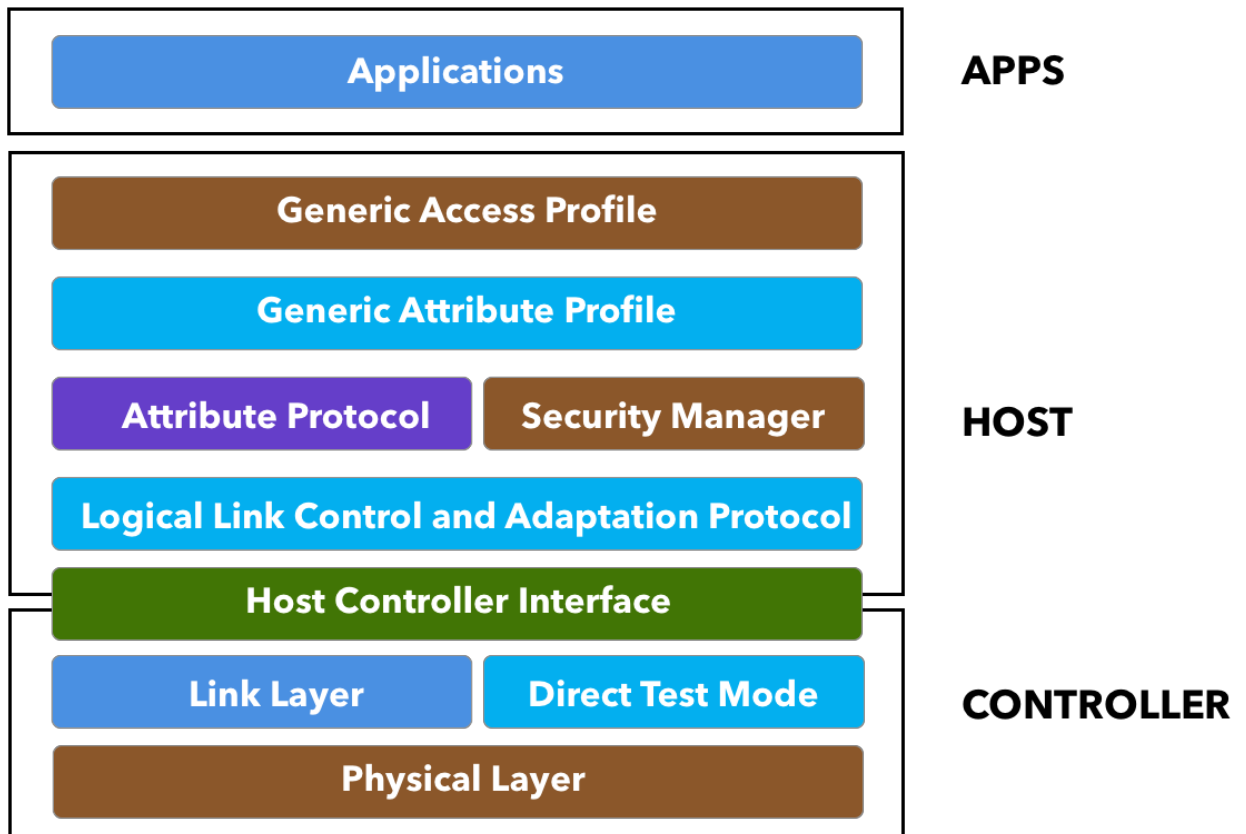


Рисунок 2.5 - Стек протоколу Bluetooth Low Energy

Розглянемо, що означає кожен із рівнів даного протоколу.

- 1) Фізичний рівень (PHY) – відноситься до фізичного радіозв'язку, який використовується для зв'язку та модуляції/демодуляції даних. Він працює в діапазоні ISM (спектр 2,4 ГГц).
- 2) Рівень зв'язку — це рівень, який взаємодіє з фізичним рівнем (радіо) і надає вищим рівням абстракцію та спосіб взаємодії з радіо через проміжний рівень, який називається рівнем HCI. Він відповідає за

керування станом радіо, а також за вимоги до часу для дотримання специфікації Bluetooth Low Energy.

- 3) Рівень інтерфейсу хост-контролера (HCI) — це стандартний протокол, визначений специфікацією Bluetooth, який дозволяє рівню Host обмінюватися даними з рівнем Controller. Ці шари можуть існувати на окремих чипах або на одному.
- 4) Рівень протоколу керування та адаптації логічного з'єднання (L2CAP) — діє як рівень мультиплексування протоколу. Він бере кілька протоколів із верхніх рівнів і розміщує їх у стандартних пакетах BLE. Потім ці пакети передаються на нижні рівні під ним.
- 5) Рівень протоколу атрибутів (ATT) — визначає, як сервер надає свої дані клієнту та як ці дані структуровані.
- 6) Рівень загального профілю атрибутів (GATT) — визначає формат даних, які надає пристрій BLE. Він також визначає процедури, необхідні для доступу до даних, відкритих пристроєм.

У GATT є дві ролі: сервер і клієнт. Сервер — це пристрій, який розкриває дані, якими він керує або містить і можливо, деякі інші аспекти своєї поведінки, якими можуть керувати інші пристрої.

З іншого боку, клієнт — це пристрій, який взаємодіє з сервером з метою зчитування відкритих даних сервера та/або керування поведінкою сервера. Пристрій Bluetooth Low Energy може виступати як сервер і клієнт одночасно.

Щоб зрозуміти рівень GATT, потрібно зрозуміти терміни послуги та характеристики. Послуги — це групування одного або кількох атрибутів (загальний термін для будь-якого типу даних, які надає сервер). Він призначений для групування пов'язаних атрибутів, які задовольняють певні функції на сервері. Наприклад, служба батареї містить одну характеристику, яка називається рівнем батареї. Характеристика завжди є частиною послуги, представляючи частину інформації/даних, які сервер хоче надати клієнту. Наприклад,

характеристика рівня заряду батареї представляє рівень заряду батареї в пристрої, який може зчитувати клієнт.

У BLE існує шість типів операцій над характеристиками:

- Команди;
- Запити;
- Відповіді;
- Сповіщення;
- Показання;
- Підтвердження.

Загальний профіль доступу (GAP) забезпечує структуру, яка визначає, як пристрої BLE взаємодіють один з одним. Він включає:

- Ролі пристроїв BLE;
- Рекламу (трансляція, відкриття, параметри реклами, дані реклами);
- Встановлення з'єднання (ініціювання з'єднань, прийняття з'єднань, параметри з'єднання);
- Безпеку.

В свою чергу ролі пристрою Bluetooth Low Energy поділяються на:

- Транслятор: пристрій, який розсилає рекламу, не отримує пакетів і не дозволяє підключення від інших;
- Спостерігач: пристрій, який слухає інші надсилання рекламних пакетів, але не ініціює підключення до рекламного пристрою.
- Центр : пристрій, який виявляє та прослуховує інші пристрої, які рекламують. Центр також має можливість підключення до рекламного пристрою.
- Периферійний пристрій: пристрій, який рекламує та приймає підключення від центральних пристроїв.

Потрібно враховувати що один пристрій може працювати в кількох ролях одночасно. Наприклад, смартфон може виконувати роль центрального пристрою під час зв'язку з розумним годинником, а також виконувати роль периферійного пристрою під час спілкування з ПК.

Після розгляду основної інформації про характеристики та основні способи використання протоколу Bluetooth Low Energy, можна починати розробку мобільного додатку. Розглянута інформація допоможе у розробці, а саме для розуміння правильності виконання дій спрямованих на використання даного протоколу або під час виникнення проблем та помилок.

2.3 Безпека передачі даних протоколу BLE

Як відомо із попереднього розділу, Bluetooth Low Energy використовується для періодичної передачі малих пакетів даних. Ключовим словом є періодичність: BLE весь час перебуває в режимі сну, за винятком участі в обміні даними, що зменшує загальне споживання енергії. Ця зміна специфікації Bluetooth дозволила виробникам створити безліч невеликих переносних пристроїв, які можуть працювати до кількох років від невеликої батареї. Гарними прикладами пристроїв із підтримкою BLE є фітнес-трекери, вимірювачі артеріального тиску та промислові датчики.

Існує певна топологія мереж Bluetooth Low Energy: пристрої можуть бути центральними або периферійними.

Центральні пристрої — смартфони, планшети, комп'ютери тощо — мають більші можливості обробки та відповідають за керування периферійними пристроями. На центральних пристроях зазвичай працює програмне забезпечення, створене спеціально для взаємодії з периферійними пристроями.

Периферійні пристрої служать датчиками, які збирають дані та надсилають їх на центральні пристрої для обробки. Ці периферійні датчики можуть бути чим завгодно: від переносних медичних моніторів до промислових термостатів: головне, що вони не обробляють дані; тільки збирають.

Практично неможливо зламати модулі BLE в пристроях, які вже перевірили підключення. Однак, щоб підключитися, пристрої повинні

спочатку з'єднатися, і саме в цьому полягає основна вразливість систем з підтримкою BLE.

Процес створення пари відбувається в два-три етапи, під час яких пристрої обмінюються даними, необхідними для автентифікації, такими як ключі та можливості введення/виведення. Третя фаза є необов'язковою та виникає лише тоді, коли пристрої збираються зв'язуватися.

З'єднання — це процес, під час якого пристрої зберігають дані автентифікації, якими вони обмінялися під час першого сполучення, що дозволяє їм запам'ятовувати один одного як безпечні під час повторного підключення в майбутньому.

Як вже згадувалося, під час першого етапу сполучення, пристрої обмінюються основною інформацією про свої можливості, щоб визначити, як продовжити підключення. В основному пристрої ідентифікують себе в мережі та пояснюють, що вони собою являють і що вони можуть робити. Цей обмін не зашифрований.

Друга фаза сполучення присвячена генерації та обміну ключами. Саме в цей момент з'єднання BLE можуть бути підроблені: якщо з'єднання не захищено належним чином, зловмисники можуть отримати контроль над пристроями та даними, які передають пристрої.

Pairing process

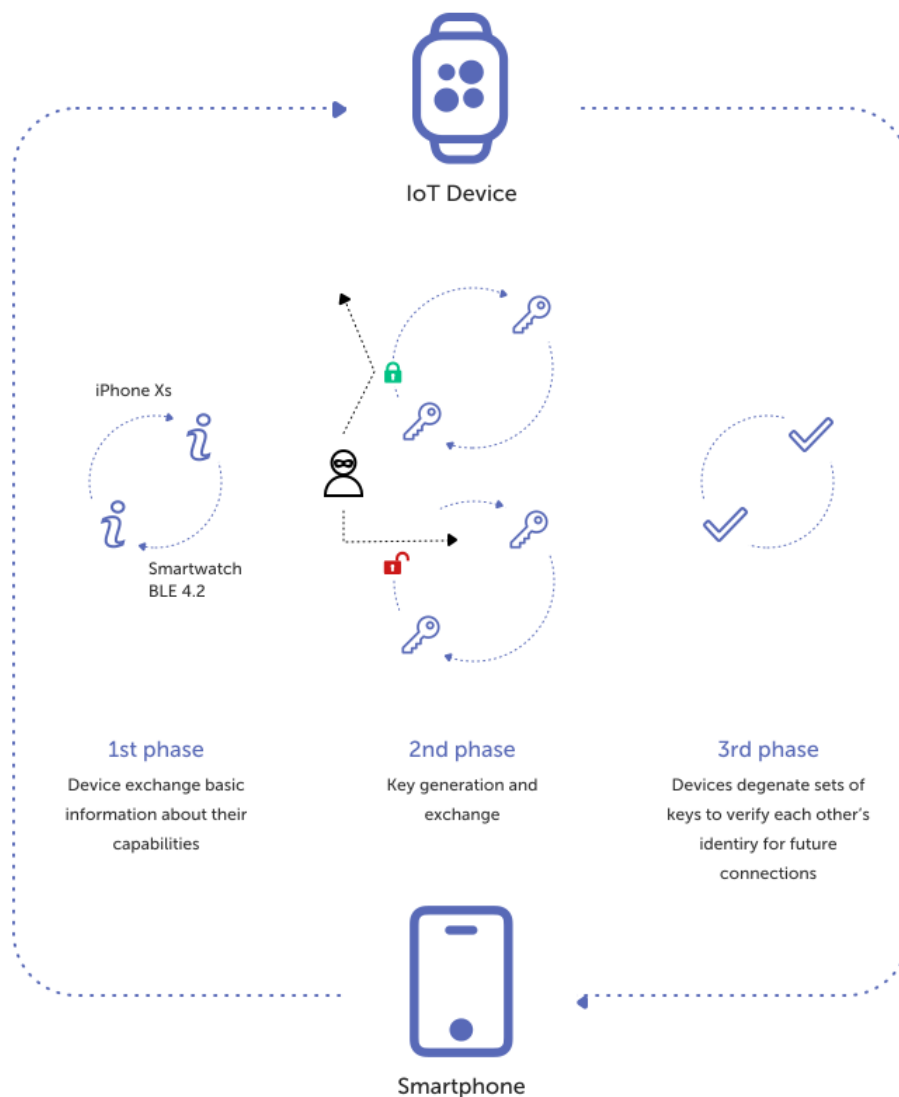


Рисунок 2.6 – Процес з'єднання пристроїв BLE

Оскільки друга фаза є дуже вразливим місцем у з'єднаннях BLE, розробники технології доклали багато зусиль, щоб захистити її. Можна виділити два типи з'єднань BLE — Legacy і Secure, між якими програмісти модулів BLE можуть обирати.

Legacy підключення можуть бути реалізовані для BLE версій 4.0, 4.1 і 4.2. Процес простий: пристрої обмінюються значенням, яке називається тимчасовим ключем (ТК), і використовують його для створення короткострокового ключа (СТК), який потім використовується для авторизації

з'єднання. З'єднання BLE Legacy є незахищеними за умовчанням, але їх можна захистити за допомогою відповідного методу сполучення.

Secure з'єднання були представлені починаючи з BLE 4.2 і несумісні зі старими версіями BLE. Вони реалізують алгоритм еліптичної кривої Діффі–Хеллмана для генерації ключів і вводять більш складний процес автентифікації ключів. Це забезпечує стандартний захист від пасивного прослуховування та дозволяє додатково захистити пристрій за допомогою відповідного методу з'єднання.

Під час необов'язкової третьої фази сполучення, пристрої генерують набори ключів для перевірки ідентичності один одного для майбутніх з'єднань. Це може бути, наприклад, пара ключів визначення підпису підключення, які використовуються для підпису даних, або ключів визначення ідентифікації, які використовуються для створення приватних MAC-адрес і пошуку.

Існує два типи кібератак, які зазвичай пов'язують із зломом модулів Bluetooth Low Energy: пасивне прослуховування та «людина посередині».

Пасивне прослуховування — це атака, яка дозволяє сторонньому пристрою отримати доступ до даних, що передаються між пристроями в мережі BLE. Наприклад, зловмисник може прослухати дані, які промислові периферійні датчики надсилають до центрального блоку, а потім використати їх, щоб з'ясувати інші вразливості безпеки в системі. Модулі BLE, які реалізують безпечні з'єднання BLE, за замовчуванням захищені від пасивного прослуховування.

Атака типу "людина посередині" включає сторонній пристрій, який прикидається одночасно центральним і периферійним і змушує інші пристрої в мережі підключитися до нього. Це може стати проблемою на великих виробничих комплексах, оскільки чужорідний пристрій може ввести в потік неправдиві дані та спричинити збій у роботі цілого виробничого ланцюжка.

Хоча захищені з'єднання BLE забезпечують захист від пасивного прослуховування, атаки типу «людина посередині» можна запобігти лише за допомогою відповідного методу з'єднання.

Оскільки виробники Bluetooth знають про проблеми протоколу зв'язку Bluetooth Low Energy, вони винайшли низку безпечних методів сполучення, які захищають ключі, якими обмінюються пристрої.

З'єднання поза діапазоном (OOB) дозволяє передавати деякі пакети даних через інший бездротовий протокол. Сполучення OOB можна реалізувати як під час другої фази сполучення, щоб ключі, якими обмінюються пристрої, не передавалися через менш безпечний протокол BLE.

Кінцева безпека сполучення OOB залежить від протоколу, який використовується як заміник BLE. Відповідний протокол зв'язку може запропонувати захист як від пасивного прослуховування, так і від атак типу «людина посередині».

Числове порівняння, є ще одним методом сполучення, який передбачає участь користувача. Після автоматичної перевірки своїх значень підтвердження на другому етапі з'єднання за допомогою числового порівняння пристрої використовують випадкове значення, яким вони раніше обмінялися, щоб створити інше шестизначне значення, яке потім порівнюється на обох пристроях. Числовий метод порівняння також забезпечує захист від обох видів атак.

Представлені вище методи захищають з'єднання пристроїв по протоколу BLE, не тільки від злому, а і від некоректного отримання та надсилання даних. Тобто алгоритми цих методів перевіряють надіслані та отримані дані на наявність пошкоджень.

Для додаткового ступеню перевірки пошкодження даних, було прийнято рішення перевіряти дані власноруч за допомогою перевірки контрольних сум.

Контрольна сума призначена для контролю цілісності переданих даних і є певним значенням, яке було розраховане за допомогою спеціальних

алгоритмів. Серед алгоритмів, які найчастіше застосовуються, є CRC32, MD5 та SHA-1.

Контрольна сума - це сума, що виводиться в результаті роботи одного з відомих алгоритмів (надлишковий код). Будь-які дані, які передаються цифровим способом, відправляються фрагментами і нерідкі випадки, коли ці фрагменти губляться дорогою. Хеш призначений для перевірки цілісності даних та виявлення пошкоджених фрагментів.

Спеціальний алгоритм розраховує суму отриманого файлу, якщо вона збігається з контрольною сумою оригіналу, отже, передача пройшла успішно. Якщо файл (або його фрагмент) має хеш, що відрізняється, від оригінального, то з'являється помилка контрольної суми. Вона говорить про порушення цілісності файлу.

Перевірка контрольних сум дозволить збільшити надійність алгоритму передачі даних і виключить можливість втрати або пошкодження файлу. Для перевірки буде використовуватися алгоритм MD5.

Висновки до розділу

В результаті обґрунтування основних функцій, які повинні міститися у розроблюваному мобільному додатку, було розглянуто основні технології, за допомогою яких здійснюється передача та перевірка даних по протоколу BLE. Боло розглянуто основні недоліки технологій та способи їх вирішення.

Побудовано блок схему послідовності дій виконання програми у відповідь на основні проблеми із комунікацією між контролером та мобільним телефоном.

3 РОЗРОБКА ТА ТЕСТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ КЕРУВАННЯ КОНТРОЛЕРОМ

3.1 Вибір та аналіз інструментів розробки

Для розробки спочатку потрібно обрати інструменти за допомогою яких і буде розроблятися мобільний додаток. Основні інструменти такі як: мова програмування, середовище розробки та середовище тестування повинні мати усі необхідні функції для коректної взаємодії між собою.

Для початку потрібно визначитися із мовою програмування. Основні критерії вибору це кросплатформеність мови та наявність потрібних для розробки функцій та бібліотек, зокрема бібліотека для роботи з протоколом BLE. В попередньому розділі було порівняно декілька мов програмування і лише дві підходять для розробки додатку – це мова Python та мова JavaScript, оскільки це популярні кросплатформені мови, які мають фреймворки для розробки мобільних додатків.

Після аналізу та порівняння вибір було зроблено на користь мови програмування JavaScript, оскільки вона має розгорнуту бібліотеку для роботи з BLE, на відміну від мови Python, яка хоч і має подібну бібліотеку, але вона є досить конфліктною та із вузьким функціоналом.

Для розробки мобільних додатків JavaScript має доволі популярний фреймворк, з великим функціоналом та великою кількістю бібліотек, який має назву React Native.

React Native - це фреймворк для розробки кросплатформових додатків для iOS та Android. Для роботи із BLE існує дві основних бібліотеки: «React Native BLE Manager» та «React Native BLE PLX», які мають усі необхідні функції для роботи із Bluetooth Low Energy пристроями.

В рамках цієї роботи було обрано середовище розробки Visual Studio Code, оскільки воно є безкоштовним та є можливість додатково встановити

різні плагіни, які допоможуть швидше орієнтуватися у коді та виправляти помилки.

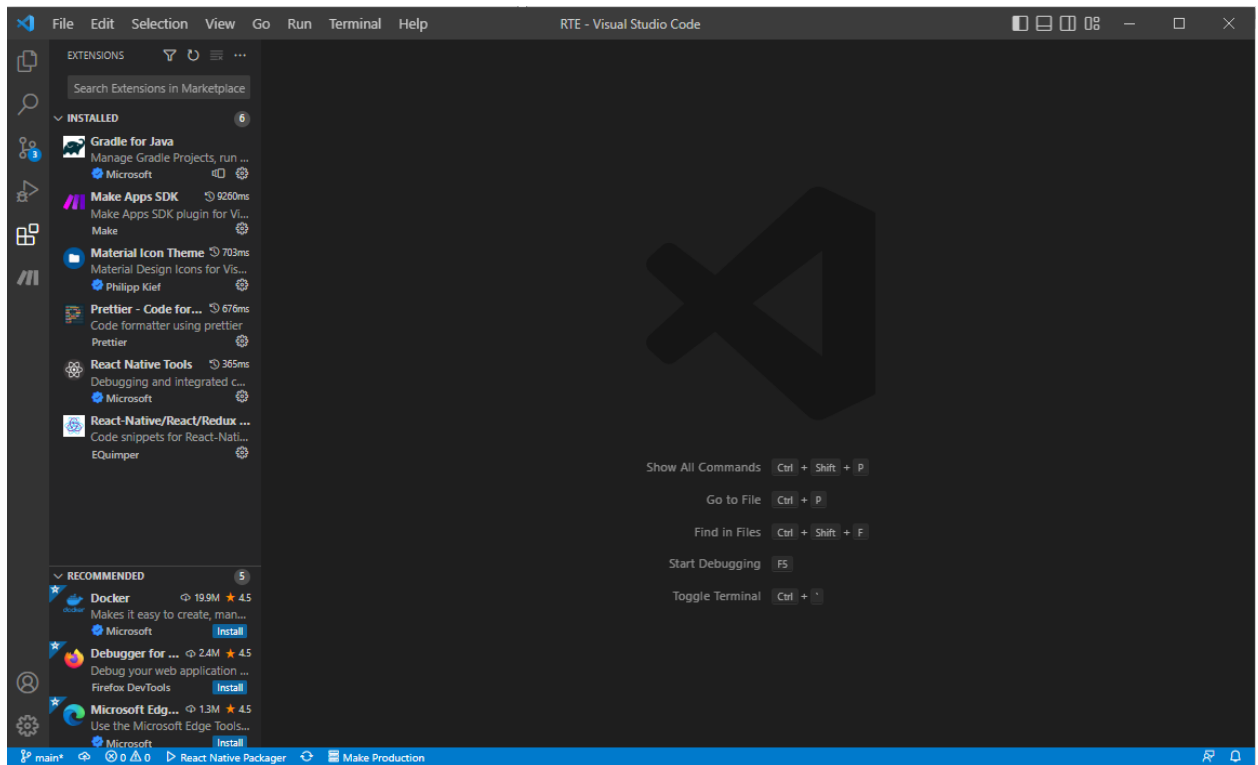


Рисунок 3.1 – Середовище розробки Visual Studio Code

Після встановлення на робочу машину середовища розробки потрібно також оглянути способи тестування та огляду додатку який розробляється. Для мобільної розробки існують програми емулятори, які в реальному часі будуть відображати на макеті телефону додаток який розробляється і при кожному збереженні коду функціонал та інтерфейс буде змінюватися в залежності від того що написано у коді. Такий спосіб є єдиним вірним, оскільки компіляція проекту в додаток для телефону займає велику кількість часу, що є не доцільним.

Оскільки додаток, який розробляється є кросплатформним, то програми емулятори для iOS та Android будуть різними.

Для Android це програма «Android Studio», яка дозволяє тестувати програму тільки на пристроях з відповідною операційною системою.

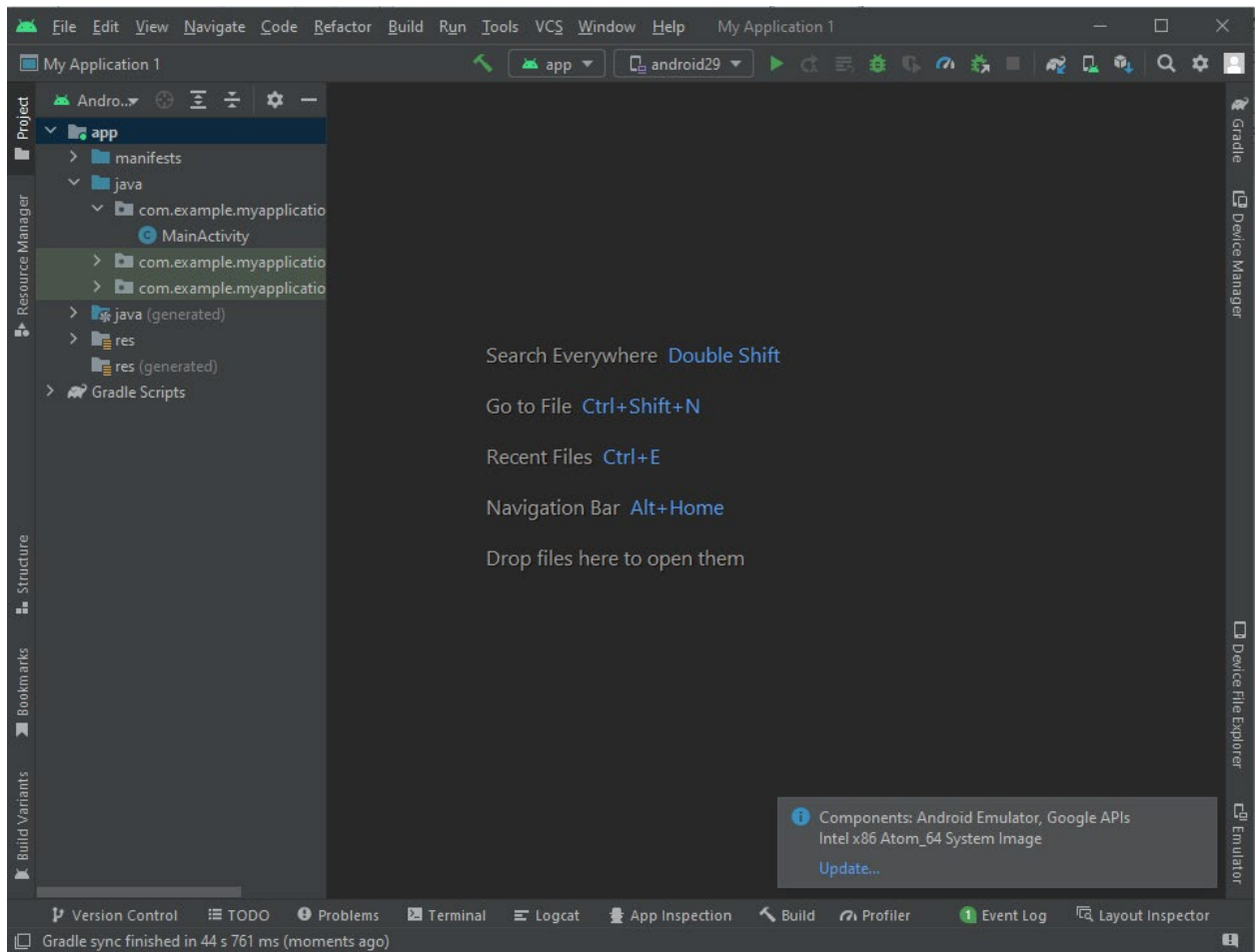


Рисунок 3.2 - Android Studio — інтегроване середовище розробки для платформи Android

Для початку роботи із цією системою, потрібно створити віртуальний пристрій та налаштувати його. В даному випадку було обрано пристрій Pixel 3, який використовує версію системи Android – 10.0, а версію API – 29.



Рисунок 3.3 – Пристрій-емулятор Pixel 3

Для тестування додатку на мобільних телефонах системи iOS потрібно дещо більше інструментів та налаштувань, оскільки запустити емулятор Xcode – який створений спеціально для написання програм для системи iOS та Mac OS, можна тільки в операційній системі Mac OS. Для використання емулятора Xcode було прийнято рішення встановити операційну систему Mac OS, як віртуальну машину в основній операційній системі Windows. Для цього потрібно спочатку встановити програму для створення та керування віртуальними машинами. В даному випадку це програма «Oracle VM VirtualBox». Після встановлення, потрібно завантажити образ операційної системи Mac OS та встановити його, як віртуальну машину. Нарешті після всього вище вказаного потрібно встановити програму Xcode, яка і буде використовуватися як емулятор.

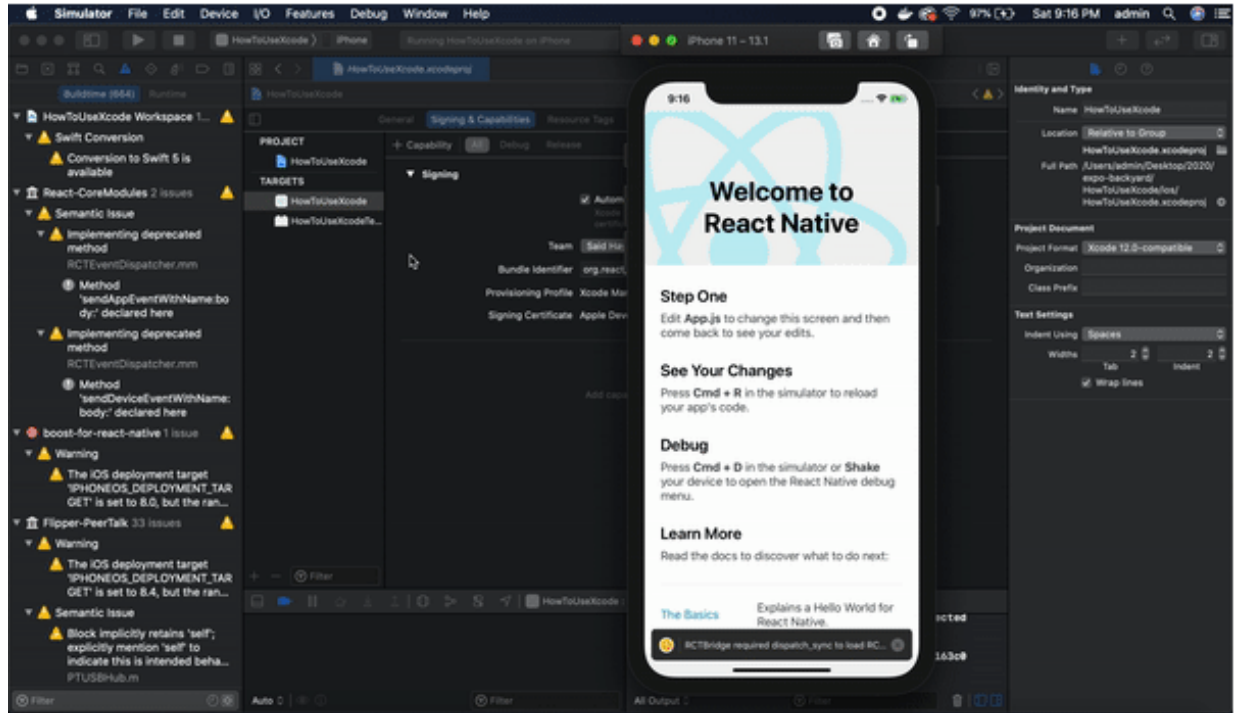


Рисунок 3.4 - Xcode — інтегроване середовище розробки виробництва Apple

Після вибору усіх необхідних інструментів потрібно встановити необхідні бібліотеки та інструменти на робочу машину. Найважливіший інструмент, який необхідно встановити спочатку – це Node.js — платформа з відкритим кодом для виконання високопродуктивних мережових застосунків, написаних мовою JavaScript. Ця платформа дає можливість розробки мовою JavaScript, а також фундаментальний набір інструментів та можливостей для розробки.

Наступним інструментом для встановлення є Expo — це платформа з відкритим вихідним кодом для створення універсальних нативних програм для Android та iOS за допомогою JavaScript та React. Саме ця платформа дозволить створювати кросплатформені проекти, для обох операційних систем мобільних телефонів.

Також по ходу розробки були встановлені додаткові бібліотеки, для розширення функціоналу розроблюваного додатку. Зокрема це бібліотека для

роботи із протоколом BLE – «react-native-ble-plx», бібліотека навігації в додатку – «react-navigation» та бібліотека для роботи із датою та часом – «react-native-modern-datepicker».

Слід також зазначити, що для контролю версій було використано Git — розподілена система керування версіями файлів та спільної роботи. Для збереження коду та змін у ньому в мережі Інтернет було використано сервіс GitHub — вебсервіс для спільної розробки програмного забезпечення.

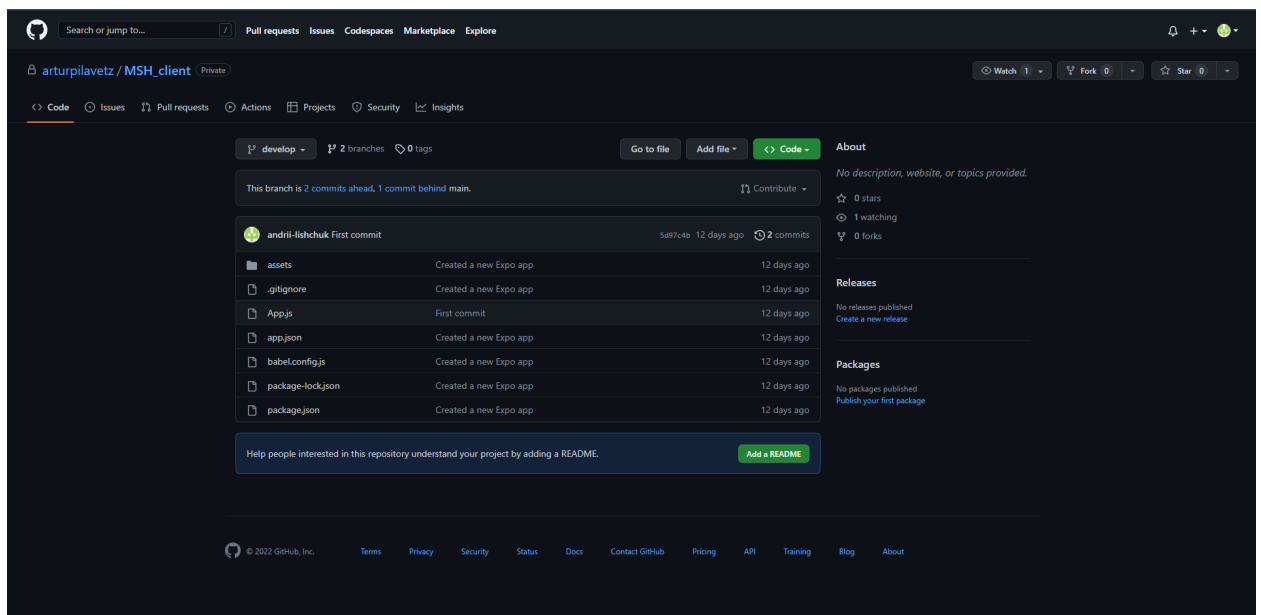


Рисунок 3.5 – Сервіс GitHub із використанням розподіленої системи керування версіями файлів

3.2 Розробка та тестування мобільного додатку

Для початку розробки потрібно визначитися із функціоналом додатку, який він повинен мати:

- 1) Робота із пристроями BLE(сканування, підключення та передача даних на пристрій);
- 2) Базова навігація(бокове меню).

Базова навігація буде представлена у вигляді бокового меню. В якому буде можливість перейти на основні сторінки додатку, поки що це сторінка

з'єднання та передачі інформації про необхідну температуру на контролер. Бокове меню створене за допомогою бібліотеки «react-navigation/drawer», яка дозволяє реалізувати бокове меню, яке буде працювати як в операційній системі Android так і в iOS. Початкове наближення вигляду бокового меню зображене на рисунку 3.6.

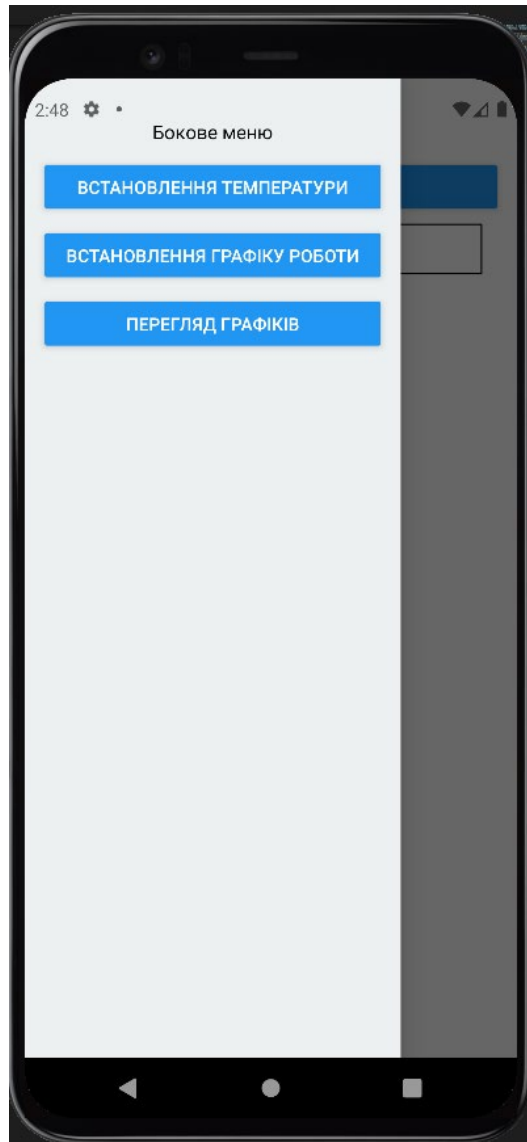


Рисунок 3.6 - Початкове наближення вигляду бокового меню

Бокове меню поки має початковий вигляд, оскільки програма досі знаходиться на стадії розробки. Функціонал буде поступово збільшуватися, в залежності від прошивки контролера. Зараз прошивка дозволяє лише підключатися до контролера та надсилати на нього значення, які в

найближчому майбутньому будуть декодуватися у потрібні для контролю температурою.

Для підключення до контролера необхідно спочатку просканувати доступні пристрої на телефоні. Для чого потрібно натиснути кнопку «Сканувати». Після чого обрати потрібний контролер зі списку доступних. Для цього був розроблений простий інтерфейс для користувача, який зображений на рисунку 3.7.

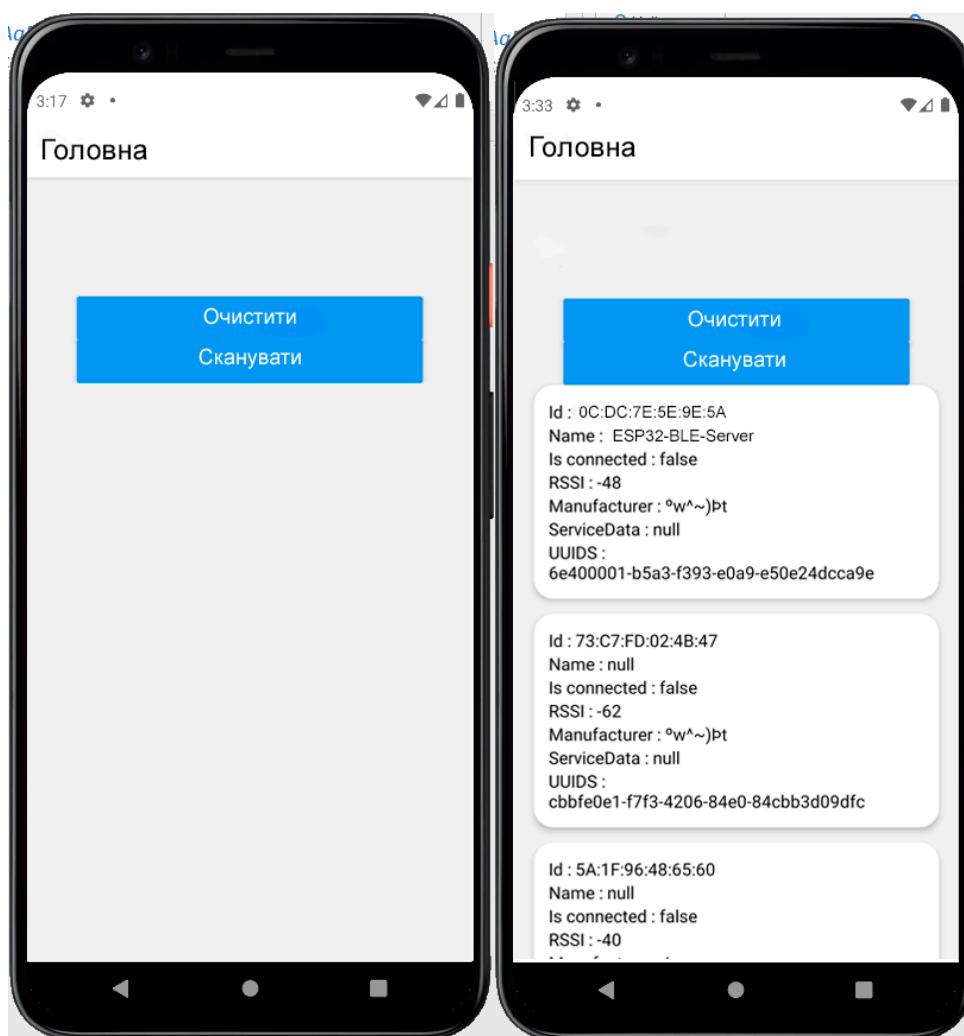


Рисунок 3.7 – Інтерфейс користувача сканування та вибору Bluetooth пристроїв

Як можемо бачити, у списку є потрібний контролер, що означає що бібліотека «react-native-ble-plx» налаштована у проекті правильно. Дана бібліотека може виконувати всі основні функції, які пов'язані із BLE

пристроями. В даному мобільному додатку були використані такі з функцій як: сканування пристроїв, отримання інформації про пристрій(основна інформація про пристрій представлена на рисунку 3.7 – зправа), підключення до пристрою, а також надсилання інформації на пристрій – у даному випадку надсилання бажаної температури для обігріву.

Для надсилання значення на контролер потрібно всього лиш обрати потрібний контролер зі списку, після цього ввести потрібну температуру у поле, яке з'явиться нижче і натиснути кнопку надіслати. Приклад виконання цієї дії зображений на рисунку 3.8.

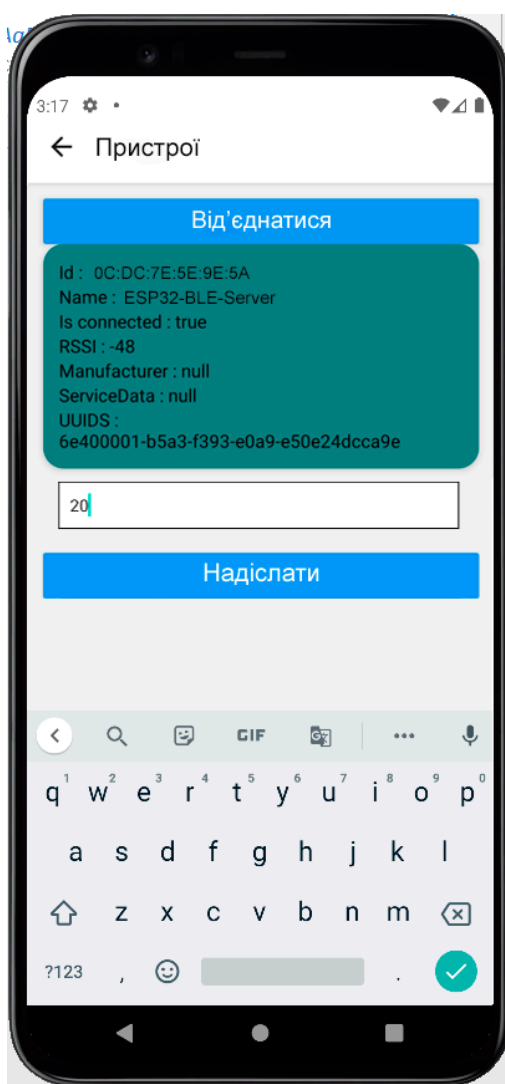


Рисунок 3.8 – Надсилання значення температури на контролер

Для перевірки працездатності необхідно відкрити дисплей порта до якого під'єднаний контролер, через програму Arduino та відстежити передане значення. На рисунку 3.9 можемо бачити, що отримане значення збігається із значенням яке було передане через мобільний додаток, що означає що мобільний додаток налаштований правильно, зокрема модуль для роботи із пристроями через BLE протокол.



Рисунок 3.9 – Дисплей порта контролера

Для розробки додатку були використані лише універсальні блоки з фреймворку React Native для побудови інтерфейсу, а також універсальні вбудовані та додаткові бібліотеки і функції, які працюють як для Android так і для iOS. В майбутньому функціонал, вигляд та дизайн будуть змінюватися в залежності від функціоналу, який буде мати прошивка контролера. Зараз додаток має примітивний вигляд та функціонал, який потрібно додатково протестувати на кількох різних девайсах, оскільки програму поки було протестовано лише на емуляторах та на одному фізичному пристрої на операційній системі Android.

Висновки до розділу

В результаті обґрунтування інструментів для розробки мобільного додатку, було обрано ті, які найбільше підходять для побудови потрібного програмного забезпечення: фреймворк – React Native, середовище розробки – Visual Studio Code, середовище тестування – Android Studio та Xcode, бібліотека для роботи із протоколом BLE – «react-native-ble-plx».

Використовуючи вказані інструменти було побудовано та протестовано кросплатформений мобільний додаток, для комунікації між телефоном та контролером.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту мобільний додаток для керування децентралізованою системою опалення. Особливістю програми є те, що дана технологія полягає в керуванні контролером за допомогою технології Bluetooth, яка дозволяє швидко та доцільно налаштувати температурний режим та мікроклімат у приміщенні.

Аналогами мобільного додатку можуть бути безкоштовні мобільні додатки SmartThings, Amazon Alexa, Smart Life - Smart Living і відповідні їм контролери, якими і дозволяють керувати дані додатки: DEVI Devireg Smart Black – 6000 грн; Ecoset-1823 WiFi – 1500 грн; Nest learning thermostat gen3 – 9000 грн. Перевагою розробленого додатку є те, що він розроблений під контролер (esp32 dev kit v1), вартістю 200 грн., в комплекті з яким використовується силове реле – 100 грн., термодатчик (ds18b20) – 50 грн. та корпус – 50 грн., т.б. вартість всього комплекту складає 400 грн. до якої ми додаємо вартість нематеріального ресурсу у 100 грн., тобто вартість всієї системи для керування децентралізованою системою опалення разом з мобільний додатком складе 500 грн., що значно нижче ніж у конкурентів.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність в реальних умовах
Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Продовження табл. 4.1

9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	3	3	3
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4

Продовження табл. 4.2

Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	3	3
Супровідна документація	3	3	3
Сума	42	40	41
Середньоарифметична сума балів	$(42+40+41) / 3 = 41$		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високою, що досягається за рахунок того, що дана розробка дозволяє зручно та швидко керувати електричними нагрівальними елементами в залежності від потреб користувача, що дозволяє раціонально та економічно використовувати електроенергію. Додаток розробляється під конкретний контролер, який виготовлено іншим розробником, а саме під (esp32 dev kit v1), вартістю 200 грн., в комплекті з яким використовується силове реле – 100 грн., термодатчик (ds18b20) – 50 грн. та корпус – 50 грн., т.б. вартість всього комплекту складає 400 грн. до якої ми додаємо вартість нематеріального ресурсу у 100 грн., тобто вартість всієї системи для керування децентралізованою системою опалення разом з мобільний додатком складе 500 грн., що значно нижче ніж у конкурентів.

4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

4.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів в місяці, 20 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.4.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	23000	1150,00	30	34500,000
Програміст	20000	1000,00	30	30000,000
Всього				64500,00

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

4.2.2 Додаткова заробітна плата розробників, які приймали участь в розробці обладнання.

Додаткова заробітна плата прийнято розраховувати як 10 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 10 \% / 100 \% \quad (4.2)$$

$$З_д = (64500,00 \cdot 10 \% / 100 \%) = 6450,00 \text{ (грн.)}$$

4.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22 \% / 100\% \quad (4.3)$$

$$Н_з = (64500,00 + 6450,00) \cdot 22 \% / 100 \% = 15609,00 \text{ (грн.)}$$

4.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

4.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{T_{\text{в}} \cdot 12} \cdot t_{\text{вик}} \text{ [грн.]} \quad (4.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер

балансова вартість якого становить 20000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,50 міс.

$$A_{обл} = \frac{20000}{2} \times \frac{1,5}{12} = 1250 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 3.5. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

$$A_{н.р.} = Ц_{н.р.} * H_a * \frac{t_{вик.}}{12} \quad (4.5)$$

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $B_{нем.ак.} = 7100$ грн.

Таблиця 4.5 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	20000	2	1,50	1250,000
Офісне обладнання	20000	4	1,50	625,000
Приміщення	650000	20	1,50	4062,500
Всього				5937,50

4.2.6 Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того,

розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi}, \quad (4.6)$$

де V – вартість 1 кВт-години електроенергії для 1 класу підприємства, $V = 6,2$ грн./кВт;

P – встановлена потужність обладнання, кВт. $P = 0,35$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

K_{Π} – коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$V_e = 0,9 \cdot 0,35 \cdot 8 \cdot 30 \cdot 6,2 = 468,72 \text{ (грн.)}$$

4.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (4.7)$$

де H_{ib} – норма нарахування за статтею «Інші витрати».

$$I_e = 64500,00 \cdot 100\% / 100\% = 64500 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та

раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (З_o + З_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.8)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 64500,00 * 153 \% / 100 \% = 98685 \text{ (грн.)}$$

4.2.8 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 64500,00 + 6450,00 + 15609,00 + 5937,50 + 7100 + 468,72 + 64500 + 98685 = 263250,22 \text{ грн.}$$

4.2.9 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються $ЗВ$, визначається за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (4.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 263250,22 / 0,5 = 526500 \text{ грн.}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

4.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.10)$$

де $\pm\Delta C_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

C_o – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $C_o = C_b \pm \Delta C_o$;

C_b – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні нематеріального ресурсу у 100 грн. за одиницю виробу (вартість буде включено в ціну набору терморегуляції автономного опалення, а сам додаток буде скачувати безкоштовно), термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 30 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 50000 шт., протягом другого року – на 75000 шт., протягом третього року на 100000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*30 + (100 + 30) * 50000) * 0,8333 * 0,5 * (1 - 0,18) = 1708333,265 \text{ грн.}$$

$$\Delta\Pi_2 = (0*30 + (100 + 30) * (50000 + 75000)) * 0,8333 * 0,5 * (1 - 0,18) = 5552083,111 \text{ грн.}$$

$$\Delta\Pi_3 = (0*30 + (100 + 30) * (50000 + 75000 + 100000)) * 0,8333 * 0,5 * (1 - 0,18) = 9993749,600 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 17254165,98 грн.

4.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned} ПП = & \\ = & (1708333,265/(1+0,1)^1) + (5552083,111/(1+0,1)^2) + (9993749,600/(1+0,1)^3) = \\ & 1553030,24 + 4588498,439 + 7508451,991 = 13649980,67 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * ZB, \quad (4.12)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв}=2...5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 526500 = 1053000,88 \text{ грн.}$$

Тоді абсолютний економічний ефект E_{abc} або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = ПП - PV, \quad (4.13)$$

$$E_{abc} = 13649980,67 - 1053000,88 = 12596979,79 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR , *Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_g . Для цього використаємо формулу:

$$E_{\epsilon} = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.14)$$

$T_{жс}$ – життєвий цикл наукової розробки, роки.

$$E_{\epsilon} = \sqrt[3]{(1 + 12596979,79/1053000,88)} - 1 = 1,349$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_{\epsilon} > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_{\epsilon}}, \quad (4.16)$$

$$T_{ок} = 1 / 1,349 = 0,74 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,74 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 526500 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,74 роки.

ВИСНОВКИ

В результаті виконання магістерської роботи було проведено аналіз проблеми децентралізованих систем опалення, а також розглянуто найпоширеніші системи, яким надають перевагу користувачі. Через відсутність продукту, який задовольнив би потреби, які стосуються керування децентралізованим обігрівом електронагрівальними приладами, було прийняте рішення, створити програмне забезпечення для контролера, який створюється паралельно іншим розробником.

Було спроектоване та розроблене програмне забезпечення для управління контролером на базі плати Arduino ESP32, яке комунікує із контролером за допомогою протоколу Bluetooth Low Energy. Створений мобільний додаток є кросплатформним, що було досягнуто за допомогою використання фреймворку React Native. Мобільний додаток дає змогу сканувати, під'єднуватися та надсилати дані на контролер. Програмне забезпечення дозволяє передавати контролеру бажану температуру для обігріву.

Мобільний додаток має початкову функціональність, яка буде поступово збільшуватися, в залежності від функціоналу контролера. Дизайн програми є початковим, оскільки ціль даної роботи це насамперед створення мобільного додатку із справним функціоналом передачі даних між девайсами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Об'єктивне порівняння iOS і Android в 2019 році [Електронний ресурс] – Режим доступу: <https://ilounge.ua/ua/review/obektivnoe-sravnenie-ios-i-android-v-2019-godu>
2. The 7 Best Programming Languages to Write & Develop Native Android Apps [Електронний ресурс] – Режим доступу: <https://www.netsolutions.com/insights/best-programming-languages-to-write-develop-android-apps/>
3. Best programming Languages for Android App Development [Електронний ресурс] – Режим доступу: <https://techvidvan.com/tutorials/best-programming-languages-for-android-app-development/>
4. Мови програмування для мобільної розробки [Електронний ресурс] – Режим доступу: <https://code.tutsplus.com/uk/articles/mobile-development-languages--cms-29138>
5. How Android works [Електронний ресурс] – Режим доступу: <https://javarevisited.blogspot.com/2013/06/introduction-of-how-android-works-Java-programmers.html#axzz7g7pOf8DW>
6. Розробка Android: Kotlin і Java [Електронний ресурс] – Режим доступу: <https://seo24.kiev.ua/rozrobka/rozrobka-android-kotlin-i-java/>
7. Язык программирования Swift – особенности, преимущества, недостатки [Електронний ресурс] – Режим доступу: <https://apix-drive.com/ru/blog/useful/jazyk-programirovaniya-swift#swift3>
8. Kivu. Xamarin. React Native. Три фреймворка — один експеримент [Електронний ресурс] – Режим доступу: <https://habr.com/ru/post/420691/>
9. Kivu. Xamarin. React Native. Три фреймворка — один експеримент (часть 3) [Електронний ресурс] – Режим доступу: <https://habr.com/ru/company/developersoft/blog/421571/>
10. Smart Home: Definition, How They Work, Pros and Cons [Електронний ресурс] – Режим доступу: <https://www.investopedia.com/terms/s/smart-home.asp>

11. Что включает в себя система умный дом [Электронный ресурс] – Режим доступа: <https://freehomeabb.ru/info/sistema-umnyy-dom-chto-vkhodit/>
12. Терморегулятор: принцип работы и виды [Электронный ресурс] – Режим доступа: <https://ds-electronics.ru/support/blog/termoregulyatory/termoregulyator-princip-raboty-i-vidy/>
13. Термостат для опалювального котла: принцип роботи, види, схеми підключення [Электронный ресурс] – Режим доступа: <https://sovet-ingenera.com/otoplenie/o-drugoe/termostat-dlya-kotla-otopleniya.html>
14. Thermostats [Электронный ресурс] – Режим доступа: <https://www.explainthatstuff.com/thermostats.html>
15. MECHANICAL VS. DIGITAL THERMOSTAT: IT'S TIME TO GO DIGITAL! [Электронный ресурс] – Режим доступа: <https://blackhawksupply.com/blogs/articles/mechanical-vs-digital-thermostat-it-s-time-to-go-digital>
16. Heating controls [Электронный ресурс] – Режим доступа: <https://energysavingtrust.org.uk/advice/thermostats-and-heating-controls/>
17. How Home Thermostats Work [Электронный ресурс] – Режим доступа: <https://home.howstuffworks.com/home-thermostat.htm>
18. How a Thermostat Works [Электронный ресурс] – Режим доступа: <https://www.hometips.com/how-it-works/how-thermostat-works.html>
19. How does a thermostat work in your home? [Электронный ресурс] – Режим доступа: <https://www.mrcentralheating.co.uk/blog/how-does-a-thermostat-work-at-home/>
20. Система дистанционного управления отоплением загородного дома, дачи или коттеджа. [Электронный ресурс] – Режим доступа: <https://termogorod.ru/stati/distantcionnoe-upravlenie-otopleniem-zagorodnogo-doma>

21. Из чего собрать умный дом в 2020 году: от хаба и до лампочки [Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/mvideo/blog/499706/>
22. Системы управления отоплением – от ручного к погодозависимому [Электронный ресурс] – Режим доступа: <https://teplo.bast.ru/articles/sistemi-upravlenia-otopleniem>
23. Системы управления отоплением Умный дом [Электронный ресурс] – Режим доступа: <https://tetan.ua/technology/smart-home-system/>
24. How Do Smart Thermostats Work? [Электронный ресурс] – Режим доступа: <https://www.safewise.com/smart-home-faq/how-do-smart-thermostats-work/>
25. How Do Smart Thermostats Work? [Электронный ресурс] – Режим доступа: <https://www.thisoldhouse.com/smart-homes/21097136/everything-to-know-about-smart-thermostats>
26. How Do Smart Thermostats Work? [Электронный ресурс] – Режим доступа: <https://www.vivint.com/resources/article/how-do-smart-thermostats-work>
27. РОЗУМНИЙ ДІМ [Электронный ресурс] – Режим доступа: <https://ua.nesrakonk.ru/smart-home/>
28. How Smart Homes Work [Электронный ресурс] – Режим доступа: <https://home.howstuffworks.com/smart-home.htm>
29. How Does a Smart Home Work? [Электронный ресурс] – Режим доступа: <https://www.getlivewire.com/smart-home-automation-work/>
30. Everything You Need to Know About Smart Home Technology [Электронный ресурс] – Режим доступа: <https://www.otelco.com/resources/smart-home-guide/>
31. What Is Home Automation and How Does It Work? [Электронный ресурс] – Режим доступа: <https://www.security.org/home-automation/>
32. Smart home guide for beginners: How to lay a foundation you can build on [Электронный ресурс] – Режим доступа:

<https://www.techhive.com/article/583408/smart-home-guide-for-beginners-how-to-make-your-home-more-convenient-to-live-in.html>

33. Smart home or building (home automation or domotics) [Электронный ресурс] – Режим доступа: <https://www.techtarget.com/iotagenda/definition/smart-home-or-building>

34. Управление отоплением по GSM на даче или в загородном доме [Электронный ресурс] – Режим доступа: <https://kouzi.ru/materials/articles/heating-control-GSM/>


35. Using GSM to control the heating boiler. Heating control systems [Электронный ресурс] – Режим доступа: <https://kayabaparts.ru/en/ispolzovanie-gsm-dlya-upravleniya-otopitelnyim-kotlom-sistemy-upravleniya/>

36. GSM МОДУЛЬ УПРАВЛЕНИЯ КОТЛОМ. ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ КОТЛОМ [Электронный ресурс] – Режим доступа: <https://bio.ukr.bio/ru/articles/3398/>

37. What Is Smart Home Technology? A Guide for Absolute Beginners [Электронный ресурс] – Режим доступа: <https://www.makeuseof.com/what-is-smart-home-technology/>

ДОДАТКИ

Додаток А
ВНТУ

ЗАТВЕРДЖЕНО
Зав. кафедри КСУ ВНТУ,
д.т.н., доцент
 В'ячеслав КОВТУН
" 30 " листопада 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання магістерської кваліфікаційної роботи
«МОБІЛЬНИЙ ДОДАТОК ДЛЯ КЕРУВАННЯ ДЕЦЕНТРАЛІЗОВАНОЮ
СИСТЕМОЮ ОПАЛЕННЯ»

Студент групи 2АКІТ-21м


Підпис

Андрій ЛІЩУК

Керівник д.т.н., професор каф. КСУ


Підпис

Володимир ДУБОВОЙ

Вінниця 2022

1. Назва та галузь застосування

1.1 Назва – Мобільний додаток для керування децентралізованою системою опалення.

1.2 Галузь застосування – Енергетичні системи.

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ від “14” вересня 2022 року №203

3. Мета та призначення розробки.

Метою роботи є розробка мобільного додатку керування децентралізованою системою опалення, через контролер, під операційні системи Android та iOS. Також метою розробки є можливість керування контролером через технологію Bluetooth, що дозволяє керувати електронагрівальними приладами, які підключені через наявний контролер.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Sizing the Opportunity for Auracast™ Broadcast Audio in Public Locations [Електронний ресурс] – Режим доступу: <https://www.bluetooth.com/>
2. React Native. Learn once, write anywhere. [Електронний ресурс] – Режим доступу: <https://reactnative.dev/>

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- Можливість управління температурою контролера;
- Передача даних через протокол BLE;

- Кросплатформеність мобільного додатку;
- Зручність використання мобільного додатку.

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- WINDOWS 10;
- Середовища тестування Android Studio та Xcode;
- Visual Studio Code.

5.2.2. Умови експлуатації системи:

- робота на мобільних операційних системах Android та iOS;
- запобігання непередбачуваного функціонування;
- дані оновлюються і є актуальними.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

1. Аналіз методів, принципів, підходів і засобів реалізації задачі автоматизації процесами в об'єкті управління відповідно до теми кваліфікаційної роботи. Постановка задач дослідження 03.09.2022 р.
2. Удосконалення технології прийняття рішень при автоматизації об'єкту управління «01»_10_2022 р.
3. Визначення технічних характеристик системи «14»_10_2022 р.
4. Розробка програмного забезпечення системи «05»_11_2022 р.

6.2 Графічні матеріали:

1. Розробка блок схеми функціонування системи «03»_12_20__ р.
2. Розробка програмного забезпечення «05»_11_2022 р.
3. Тестування програмного забезпечення «06»_12_2022 р.

7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «28»_11_2022 р.

7.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «16»_12_2022 р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до «23»_12_2022 р.

Додаток Б

ЛІСТИНГ БОКОВОГО МЕНЮ

```
// @flow

import React from 'react';
import {
  PanResponder,
  View,
  Dimensions,
  Animated,
  TouchableWithoutFeedback,
} from 'react-native';
import PropTypes from 'prop-types';
import styles from './styles';

type WindowDimensions = { width: number, height: number };

type Props = {
  edgeHitWidth: number,
  toleranceX: number,
  toleranceY: number,
  menuPosition: 'left' | 'right',
  onChange: Function,
  onMove: Function,
  onSliding: Function,
  openMenuOffset: number,
  hiddenMenuOffset: number,
  disableGestures: Function | bool,
  animationFunction: Function,
  onAnimationComplete: Function,
  onStartShouldSetResponderCapture: Function,
  isOpen: bool,
  bounceBackOnOverdraw: bool,
  autoClosing: bool
};

type Event = {
  nativeEvent: {
    layout: {
      width: number,
      height: number,
    },
  },
};

type State = {
  width: number,
  height: number,
  openOffsetMenuPercentage: number,
  openMenuOffset: number,
  hiddenMenuOffsetPercentage: number,
  hiddenMenuOffset: number,
  left: Animated.Value,
};
```

```

const deviceScreen: WindowDimensions = Dimensions.get('window');
const barrierForward: number = deviceScreen.width / 4;

function shouldOpenMenu(dx: number): boolean {
  return dx > barrierForward;
}

export default class SideMenu extends React.Component {
  onLayoutChange: Function;
  onStartShouldSetResponderCapture: Function;
  onMoveShouldSetPanResponder: Function;
  onPanResponderMove: Function;
  onPanResponderRelease: Function;
  onPanResponderTerminate: Function;
  state: State;
  prevLeft: number;
  isOpen: boolean;

  constructor(props: Props) {
    super(props);

    this.prevLeft = 0;
    this.isOpen = !!props.isOpen;

    const initialMenuPositionMultiplier = props.menuPosition === 'right' ? -1 : 1;
    const openOffsetMenuPercentage = props.openMenuOffset / deviceScreen.width;
    const hiddenMenuOffsetPercentage = props.hiddenMenuOffset / deviceScreen.width;
    const left: Animated.Value = new Animated.Value(
      props.isOpen
        ? props.openMenuOffset * initialMenuPositionMultiplier
        : props.hiddenMenuOffset,
    );

    this.onLayoutChange = this.onLayoutChange.bind(this);
    this.onStartShouldSetResponderCapture = props.onStartShouldSetResponderCapture.bind(this);
    this.onMoveShouldSetPanResponder = this.handleMoveShouldSetPanResponder.bind(this);
    this.onPanResponderMove = this.handlePanResponderMove.bind(this);
    this.onPanResponderRelease = this.handlePanResponderEnd.bind(this);
    this.onPanResponderTerminate = this.handlePanResponderEnd.bind(this);

    this.state = {
      width: deviceScreen.width,
      height: deviceScreen.height,
      openOffsetMenuPercentage,
      openMenuOffset: deviceScreen.width * openOffsetMenuPercentage,
      hiddenMenuOffsetPercentage,
      hiddenMenuOffset: deviceScreen.width * hiddenMenuOffsetPercentage,
      left,
    };

    this.state.left.addListener(({value}) => this.props.onSliding(Math.abs((value -
    this.state.hiddenMenuOffset) / (this.state.openMenuOffset - this.state.hiddenMenuOffset))));
  }

  UNSAFE_componentWillMount(): void {
    this.responder = PanResponder.create({
      onStartShouldSetResponderCapture: this.onStartShouldSetResponderCapture,

```



```

    onMoveShouldSetPanResponder: this.onMoveShouldSetPanResponder,
    onPanResponderMove: this.onPanResponderMove,
    onPanResponderRelease: this.onPanResponderRelease,
    onPanResponderTerminate: this.onPanResponderTerminate,
  });
}

```

```

UNSAFE_componentWillReceiveProps(props: Props): void {
  if (typeof props.isOpen !== 'undefined' && this.isOpen !== props.isOpen && (props.autoClosing ||
  this.isOpen === false)) {
    this.openMenu(props.isOpen);
  }
}

```

```

onLayoutChange(e: Event) {
  const { width, height } = e.nativeEvent.layout;
  const openMenuOffset = width * this.state.openOffsetMenuPercentage;
  const hiddenMenuOffset = width * this.state.hiddenMenuOffsetPercentage;
  this.setState({ width, height, openMenuOffset, hiddenMenuOffset });
}

```

```
/**
```

```
* Get content view. This view will be rendered over menu
```

```
* @return {React.Component}
```

```
*/
```

```
getContentView() {
```

```
  let overlay: React.Element<void, void> = null;
```

```
  if (this.isOpen) {
```

```
    overlay = (
```

```
      <TouchableWithoutFeedback onPress={() => this.openMenu(false)}>
```

```
        <View style={styles.overlay} />
```

```
      </TouchableWithoutFeedback>
```

```
    );
```

```
  }
```

```
  const { width, height } = this.state;
```

```
  const ref = sideMenu => (this.sideMenu = sideMenu);
```

```
  const style = [
```

```
    styles.frontView,
```

```
    { width, height },
```

```
    this.props.animationStyle(this.state.left),
```

```
  ];
```

```
  return (
```

```
    <Animated.View style={style} ref={ref} {...this.responder.panHandlers}>
```

```
      {this.props.children}
```

```
      {overlay}
```

```
    </Animated.View>
```

```
  );
```

```
}
```

```
moveLeft(offset: number) {
```

```
  const newOffset = this.menuPositionMultiplier() * offset;
```

```
  this.props
```

```
    .animationFunction(this.state.left, newOffset)
```

```

    .start(this.props.onAnimationComplete);

    this.prevLeft = newOffset;
  }

  menuPositionMultiplier(): -1 | 1 {
    return this.props.menuPosition === 'right' ? -1 : 1;
  }

  handlePanResponderMove(e: Object, gestureState: Object) {
    if (this.state.left.__getValue() * this.menuPositionMultiplier() >= 0) {
      let newLeft = this.prevLeft + gestureState.dx;

      if (!this.props.bounceBackOnOverdraw && Math.abs(newLeft) > this.state.openMenuOffset) {
        newLeft = this.menuPositionMultiplier() * this.state.openMenuOffset;
      }

      this.props.onMove(newLeft);
      this.state.left.setValue(newLeft);
    }
  }

  handlePanResponderEnd(e: Object, gestureState: Object) {
    const offsetLeft = this.menuPositionMultiplier() *
      (this.state.left.__getValue() + gestureState.dx);

    this.openMenu(shouldOpenMenu(offsetLeft));
  }

  handleMoveShouldSetPanResponder(e: any, gestureState: any): boolean {
    if (this.gesturesAreEnabled()) {
      const x = Math.round(Math.abs(gestureState.dx));
      const y = Math.round(Math.abs(gestureState.dy));

      const touchMoved = x > this.props.toleranceX && y < this.props.toleranceY;

      if (this.isOpen) {
        return touchMoved;
      }

      const withinEdgeHitWidth = this.props.menuPosition === 'right' ?
        gestureState.moveX > (deviceScreen.width - this.props.edgeHitWidth) :
        gestureState.moveX < this.props.edgeHitWidth;

      const swipingToOpen = this.menuPositionMultiplier() * gestureState.dx > 0;
      return withinEdgeHitWidth && touchMoved && swipingToOpen;
    }

    return false;
  }

  openMenu(isOpen: boolean): void {
    const { hiddenMenuOffset, openMenuOffset } = this.state;
    this.moveLeft(isOpen ? openMenuOffset : hiddenMenuOffset);
    this.isOpen = isOpen;

    this.forceUpdate();
  }

```

```

    this.props.onChange(isOpen);
  }

  gesturesAreEnabled(): boolean {
    const { disableGestures } = this.props;

    if (typeof disableGestures === 'function') {
      return !disableGestures();
    }

    return !disableGestures;
  }

  render(): React.Element<void, void> {
    const boundryStyle = this.props.menuPosition === 'right' ?
      { left: this.state.width - this.state.openMenuOffset } :
      { right: this.state.width - this.state.openMenuOffset };

    const menu = (
      <View style={[styles.menu, boundryStyle]}>
        {this.props.menu}
      </View>
    );

    return (
      <View
        style={styles.container}
        onLayout={this.onLayoutChange}
      >
        {menu}
        {this.getContentView()}
      </View>
    );
  }
}

SideMenu.propTypes = {
  edgeHitWidth: PropTypes.number,
  toleranceX: PropTypes.number,
  toleranceY: PropTypes.number,
  menuPosition: PropTypes.oneOf(['left', 'right']),
  onChange: PropTypes.func,
  onMove: PropTypes.func,
  children: PropTypes.node,
  menu: PropTypes.node,
  openMenuOffset: PropTypes.number,
  hiddenMenuOffset: PropTypes.number,
  animationStyle: PropTypes.func,
  disableGestures: PropTypes.oneOfType([PropTypes.func, PropTypes.bool]),
  animationFunction: PropTypes.func,
  onAnimationComplete: PropTypes.func,
  onStartShouldSetResponderCapture: PropTypes.func,
  isOpen: PropTypes.bool,
  bounceBackOnOverdraw: PropTypes.bool,
  autoClosing: PropTypes.bool,
};

```

```
SideMenu.defaultProps = {
  toleranceY: 10,
  toleranceX: 10,
  edgeHitWidth: 60,
  children: null,
  menu: null,
  openMenuOffset: deviceScreen.width * (2 / 3),
  disableGestures: false,
  menuPosition: 'left',
  hiddenMenuOffset: 0,
  onMove: () => {},
  onStartShouldSetResponderCapture: () => true,
  onChange: () => {},
  onSliding: () => {},
  animationStyle: value => ({
    transform: [{
      translateX: value,
    }],
  }),
  animationFunction: (prop, value) => Animated.spring(prop, {
    toValue: value,
    friction: 8,
    useNativeDriver: true,
  }),
  onAnimationComplete: () => {},
  isOpen: false,
  bounceBackOnOverdraw: true,
  autoClosing: true,
};
```

Додаток В

Лістинг передачі даних по протоколу BLE

```

import { BleManager, Device } from 'react-native-ble-plx';

const manager = new BleManager();

const HomeScreen = () => {
  return (
    <SafeAreaView>
      <View style={styles.body}>
        <View style={styles.sectionContainer}>
          <Text style={styles.sectionTitle}>Step One</Text>
        </View>
      </View>
    </SafeAreaView>
  );
}

const styles = StyleSheet.create({
  body: {
    backgroundColor: Colors.red,
  },
  sectionContainer: {
    marginTop: 32,
    paddingHorizontal: 24,
  },
  sectionTitle: {
    fontSize: 24,
    fontWeight: '600',
    color: Colors.black,
  },
});

export { HomeScreen };

const reducer = (
  state: Device[],
  action: { type: 'ADD_DEVICE'; payload: Device } | { type: 'CLEAR' },
): Device[] => {
  switch (action.type) {
    case 'ADD_DEVICE':
      const { payload: device } = action;
      if (device && !state.find((dev) => dev.id === device.id)) {
        return [...state, device];
      }
      return state;
    case 'CLEAR':
      return [];
    default:
      return state;
  }
};

const HomeScreen = () => {

```

```

    // reducer to store and display detected ble devices
    const [scannedDevices, dispatch] = useReducer(reducer, []);

    const [isLoading, setIsLoading] = useState(false);

    const scanDevices = () => {
      // display the Activityindicator
      setIsLoading(true);

      // scan devices
      manager.startDeviceScan(null, null, (error, scannedDevice) => {
        if (error) {
          console.warn(error);
        }

        // if a device is detected add the device to the list by dispatching the action into the reducer
        if (scannedDevice) {
          dispatch({ type: 'ADD_DEVICE', payload: scannedDevice });
        }
      });

      // stop scanning devices after 5 seconds
      setTimeout(() => {
        manager.stopDeviceScan();
        setIsLoading(false);
      }, 5000);
    };

    return (
      /* ...Header with title */

      /* Clear the device list to reset it */
      <Button
        title="Clear devices"
        onPress={() => dispatch({ type: 'CLEAR' })}
      />

      {isLoading ? (
        <ActivityIndicator color='teal' size={25} />
      ) : (
        <Button title="Scan devices" onPress={scanDevices} />
      )}
    );
  }
}

// screens/HomeScreen.tsx
return (
  <SafeAreaView style={styles.body}>
    <View style={styles.body}>
      <View style={styles.sectionContainer}>
        <Text style={styles.sectionTitle}>Step One</Text>
      </View>
    </View>

    <FlatList
      keyExtractor={({item}) => item.id}
      data={scannedDevices}
      renderItem={({item}) => <DeviceCard device={item} />}
      contentContainerStyle={styles.content}
    />
  )
)

```

```

    </SafeAreaView>
  );

// components/DeviceCard.tsx

type DeviceCardProps = {
  device: Device;
};

const DeviceCard = ({ device }: DeviceCardProps) => {
  const navigation = useNavigation<StackNavigationProp<RootStackParamList>>();

  const [isConnected, setIsConnected] = useState(false);

  useEffect(() => {
    // is the device connected?
    device.isConnected().then(setIsConnected);
  }, [device]);

  return (
    <TouchableOpacity
      style={styles.container}
      // navigate to the Device Screen
      onPress={() => navigation.navigate('Device', { device })}>
      <Text>{`Id : ${device.id}`}</Text>
      <Text>{`Name : ${device.name}`}</Text>
      <Text>{`Is connected : ${isConnected}`}</Text>
      <Text>{`RSSI : ${device.rssi}`}</Text>
      /* Decode the ble device manufacturer which is encoded with the base64
algorithm */
      <Text>{`Manufacturer : ${Base64.decode(
        device.manufacturerData?.replace(/[/=]/g, ""),
      )}`}</Text>
      <Text>{`ServiceData : ${device.serviceData}`}</Text>
      <Text>{`UUIDS : ${device.serviceUUIDs}`}</Text>
    </TouchableOpacity>
  );
};

const styles = StyleSheet.create({
  container: {
    backgroundColor: 'white',
    marginBottom: 12,
    borderRadius: 16,
    shadowColor: 'rgba(60,64,67,0.3)',
    shadowOpacity: 0.4,
    shadowRadius: 10,
    elevation: 4,
    padding: 12,
  },
});

export { DeviceCard };

const DeviceScreen = ({
  route,
  navigation,

```

```

}: StackScreenProps<RootStackParamList, 'Device'> => {
  const { device } = route.params;

  const [isConnected, setIsConnected] = useState(false);
  const [services, setServices] = useState<Service[]>([]);

  const disconnectDevice = useCallback(async () => {
    navigation.goBack();
    const isDeviceConnected = await device.isConnected();
    if (isDeviceConnected) {
      await device.cancelConnection();
    }
  }, [device]);

  useEffect(() => {
    const getDeviceInformations = async () => {
      const connectedDevice = await device.connect();
      setIsConnected(true);
      const allServicesAndCharacteristics = await
connectedDevice.discoverAllServicesAndCharacteristics();
      const discoveredServices = await allServicesAndCharacteristics.services();
      setServices(discoveredServices);
    };

    getDeviceInformations();

    device.onDisconnected(() => {
      navigation.navigate('Home');
    });

    return () => {
      disconnectDevice();
    };
  }, [device, disconnectDevice, navigation]);

  return (
    <ScrollView contentContainerStyle={styles.container}>
      <Button title="disconnect" onPress={disconnectDevice} />
      <View>
        <View style={styles.header}>
          <Text>`Id : ${device.id}`</Text>
          <Text>`Name : ${device.name}`</Text>
          <Text>`Is connected : ${isConnected}`</Text>
          <Text>`RSSI : ${device.rssi}`</Text>
          <Text>`Manufacturer : ${device.manufacturerData}`</Text>
          <Text>`ServiceData : ${device.serviceData}`</Text>
          <Text>`UUIIDS : ${device.serviceUUIIDS}`</Text>
        </View>
        { /* Displays a list of all services */
          &&
          services.map((service) => <ServiceCard service={service} />)}
      </View>
    </ScrollView>
  );
};

const ServiceCard = ({ service }: ServiceCardProps) => {

```



```

const [descriptors, setDescriptors] = useState<Descriptor[]>([]);
const [characteristics, setCharacteristics] = useState<Characteristic[]>([]);
const [areCharacteristicsVisible, setAreCharacteristicsVisible] = useState(
  false,
);

```

```

useEffect(() => {
  const getCharacteristics = async () => {
    const newCharacteristics = await service.characteristics();
    setCharacteristics(newCharacteristics);
    newCharacteristics.forEach(async (characteristic) => {
      const newDescriptors = await characteristic.descriptors();
      setDescriptors((prev) => [...new Set([...prev, ...newDescriptors])]);
    });
  };
};

```

```

  getCharacteristics();
}, [service]);

```

```

return (
  <View style={styles.container}>
    <TouchableOpacity
      onPress={() => {
        setAreCharacteristicsVisible((prev) => !prev);
      }}>
      <Text>`UUID : ${service.uuid}`</Text>
    </TouchableOpacity>

    {areCharacteristicsVisible &&
      characteristics &&
      characteristics.map((char) => (
        <CharacteristicCard key={char.id} char={char} />
      ))}
    {descriptors &&
      descriptors.map((descriptor) => (
        <DescriptorCard key={descriptor.id} descriptor={descriptor} />
      ))}
  </View>
);
};

```

```

const CharacteristicCard = ({ char }: CharacteristicCardProps) => {
  const [measure, setMeasure] = useState("");
  const [descriptor, setDescriptor] = useState<string | null>("");

```

```

  useEffect(() => {
    char.descriptors().then((desc) => {
      desc[0]?.read().then((val) => {
        if (val) {
          setDescriptor(Base64.decode(val.value));
        }
      });
    });
  });
};

```

```

// read on the characteristic
char.monitor((err, cha) => {
  if (err) {

```

```

        console.warn('ERROR');
        return;
    }
    setMeasure(decodeBleString(char?.value));
});
}, [char]);

// write on a characteristic the number 6 (e.g.)
const writeCharacteristic = () => {
    // encode the string with the Base64 algorithm
    char
        .writeWithResponse(Base64.encode('6'))
        .then(() => {
            console.warn('Success');
        })
        .catch((e) => console.log('Error', e));
};

return (
    <TouchableOpacity
        key={char.uuid}
        style={styles.container}
        onPress={writeCharacteristic}>
        <Text style={styles.measure}>{measure}</Text>
        <Text style={styles.descriptor}>{descriptor}</Text>
        <Text>{'isIndicatable : ${char.isIndicatable}'}</Text>
        <Text>{'isNotifiable : ${char.isNotifiable}'}</Text>
        <Text>{'isNotifying : ${char.isNotifying}'}</Text>
        <Text>{'isReadable : ${char.isReadable}'}</Text>
        <TouchableOpacity>
            <Text>{'isWritableWithResponse : ${char.isWritableWithResponse}'}</Text>
        </TouchableOpacity>
        <Text>{'isWritableWithoutResponse : ${char.isWritableWithoutResponse}'}</Text>
    </TouchableOpacity>
);
};

```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

Мобільний додаток для керування децентралізованою системою опалення
(тема)

1. Об'єкт, предмет та мета дослідження;
2. Задачі дослідження;
3. Основні системи управління децентралізованим опаленням;
4. Блок схема порядку дій;
5. Обмеження та переваги протоколу Bluetooth Low Energy;
6. Процес з'єднання пристроїв BLE;
7. Середовище розробки Visual Studio Code;
8. Середовище тестування Android Studio;
9. Зовнішній вигляд мобільного додатку;
10. Тестування програми;
11. Економічна частина;
13. Висновки.

Студент групи 2АКІТ-21м


Підпис

Андрій ЛІЩУК
Ім'я ПРІЗВИЩЕ

Керівник д.т.н., професор каф. КСУ


Підпис

Володимир ДУБОВОЙ
Ім'я ПРІЗВИЩЕ

Мобільний додаток для керування децентралізованою системою опалення

ДИПЛОМНА РОБОТА
(ОСВІТНЬО-КВАЛІФІКАЦІЙНИЙ РІВЕНЬ МАГІСТР)

СТУДЕНТА ГРУПИ 2АКІТ-21М
ЛІЩУКА АНДРІЯ РОМАНОВИЧА

КЕРІВНИК РОБОТИ
ДУБОВОЙ ВОЛОДИМИР МИХАЙЛОВИЧ
Д.Т.Н., ПРОФЕСОР КАФ. КСУ

Об'єкт, предмет та мета дослідження

Об'єктом дослідження є процес дистанційного управління контролером електронагрівального приладу.

Предметом дослідження є методи та технології управління контролером електронагрівального приладу за допомогою протоколу BLE .

Метою є дослідження є побудова кросплатформеного мобільного додатку з можливістю керування контролером через протокол Bluetooth Low Energy.

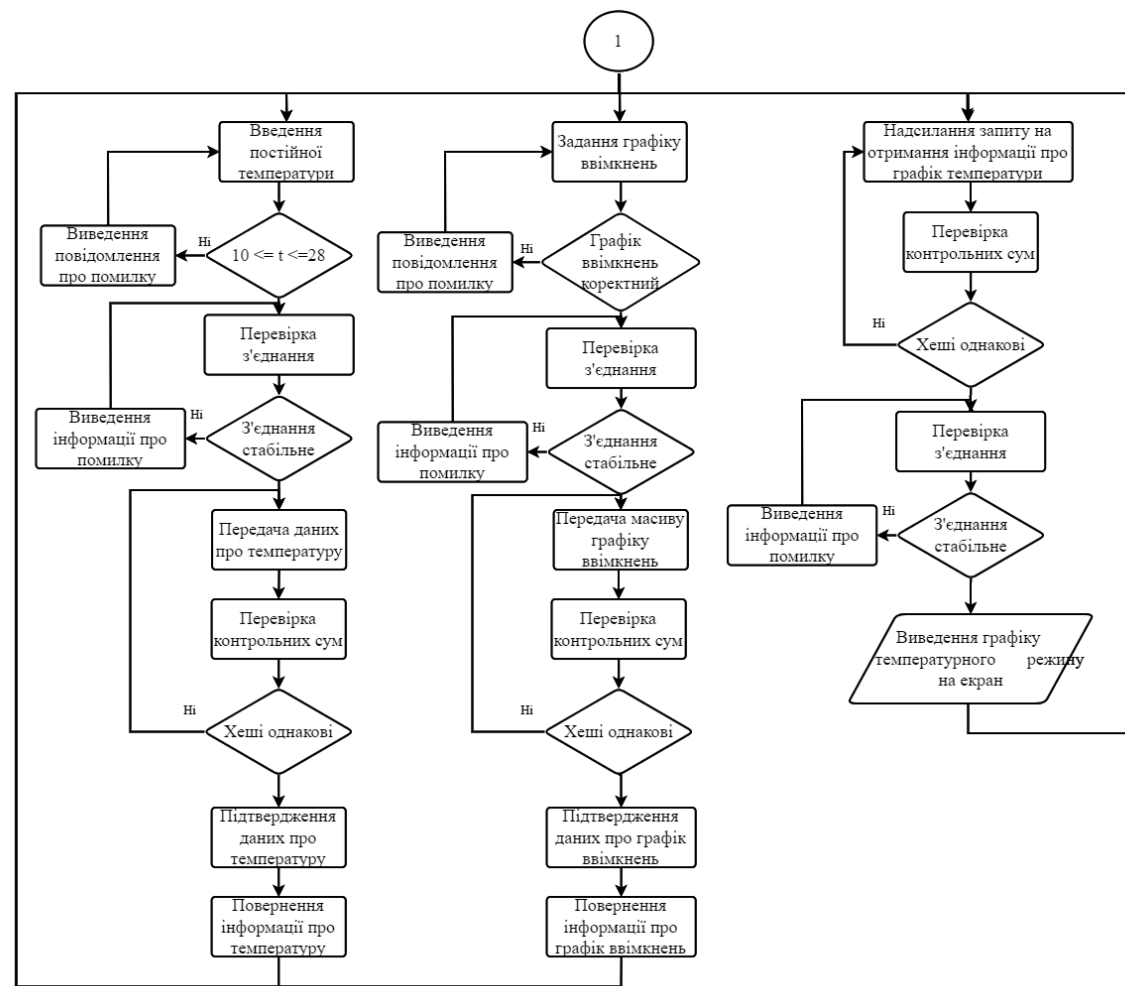
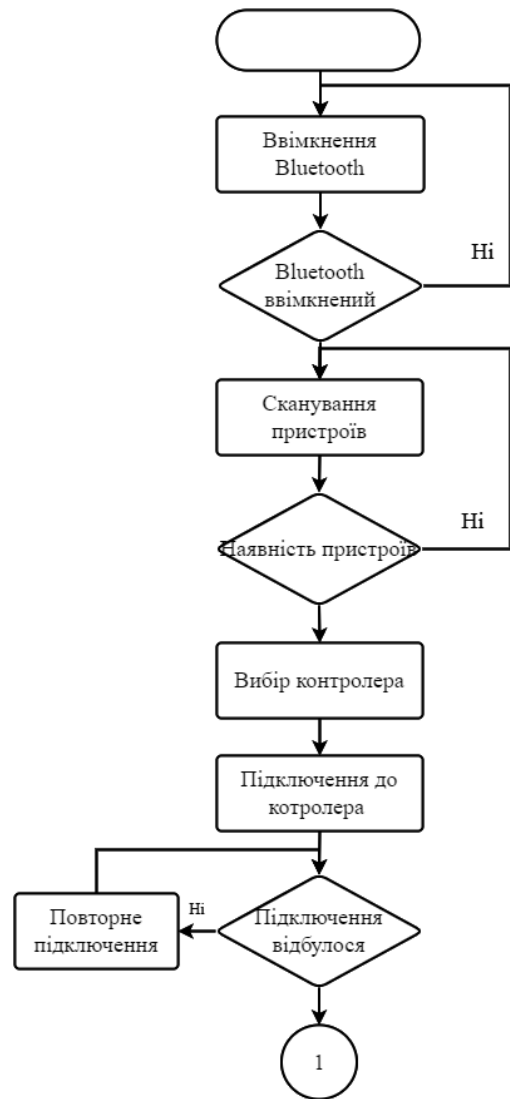
Задачі дослідження

- аналіз систем керування температурою у приміщенні та огляд аналогів;
- аналіз технологій побудови мобільного додатку;
- огляд функціональності програми;
- розробка та тестування мобільного додатку;

Основні системи управління децентралізованим опаленням

	Дешевизна	Зручність	Простота установки	Захищеність	Ефективність
Механічний термостат	5	2	4	4	2
Цифровий термостат	4	3	4	4	3
«Розумний термостат»	1	5	1	2	5
GSM розетка	3	3	5	3	3
GSM блок-модуль	2	3	2	3	3

Блок схема порядку дій



Обмеження та переваги протоколу Bluetooth Low Energy

Specifications	Classic Bluetooth	Bluetooth Low Energy (BLE)
Range	100 m	Greater than 100 m
Data Rate	1-3 Mbps	1 Mbps
Application Throughput	0.7 -2.1 Mbps	0.27 Mbps
Frequency	2.4 GHz	2.4 GHz
Security	56/128-bit	128-bit AES with Counter Mode CBC-MAC
Robustness	Adaptive fast frequency hopping, FEC, fast ASK	24-bit CRC, 32-bit Message Integrity Check
Latency	100 ms	6 ms
Time Lag	100 ms	3 ms
Voice Capable	Yes	No
Network Topology	Star	Star
Power Consumption	1 W	0.01 to 0.5 W
Peak Current Consumption	less than 30 mA	less than 15 mA

Advantages of Bluetooth LE



Low Power Consumption



Low Development Costs



Open Access to Documentation



Prevalence in Smartphones

Limitations of BLE



Data Throughput

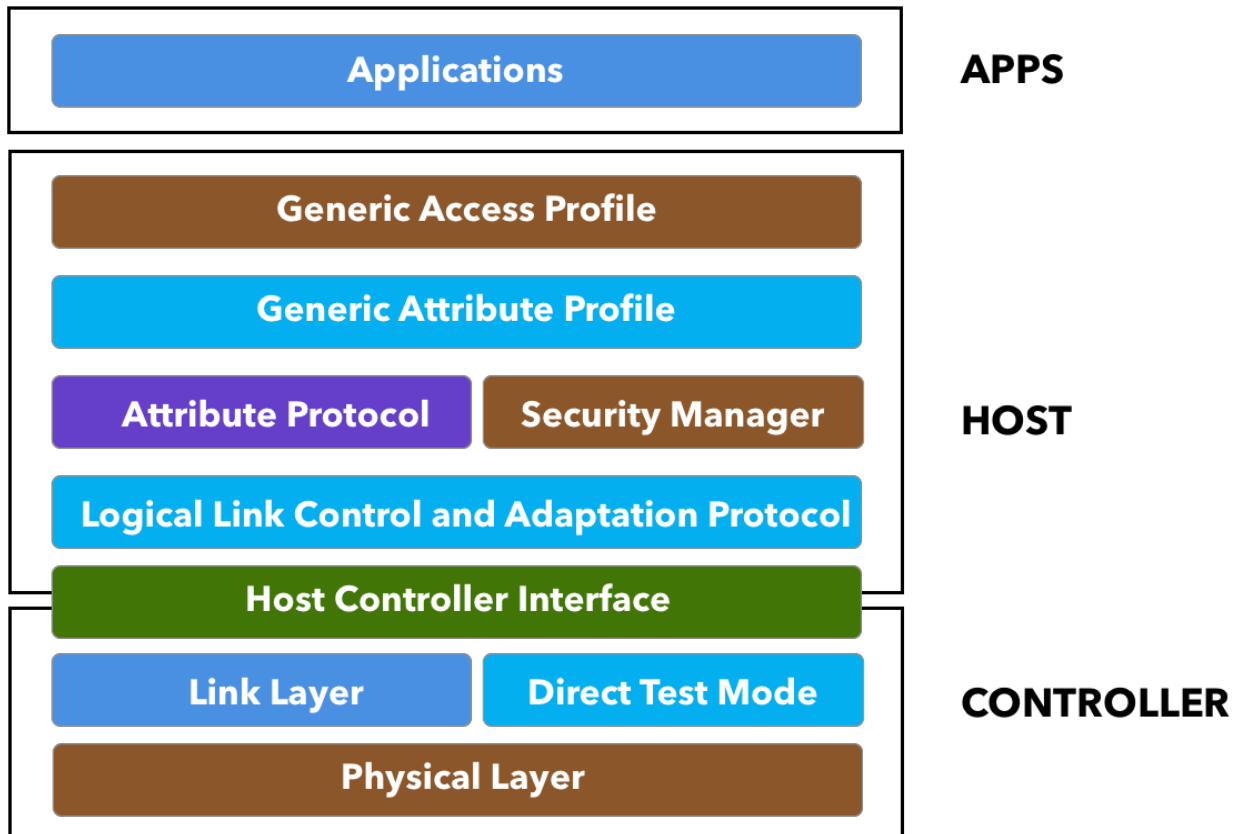


Range

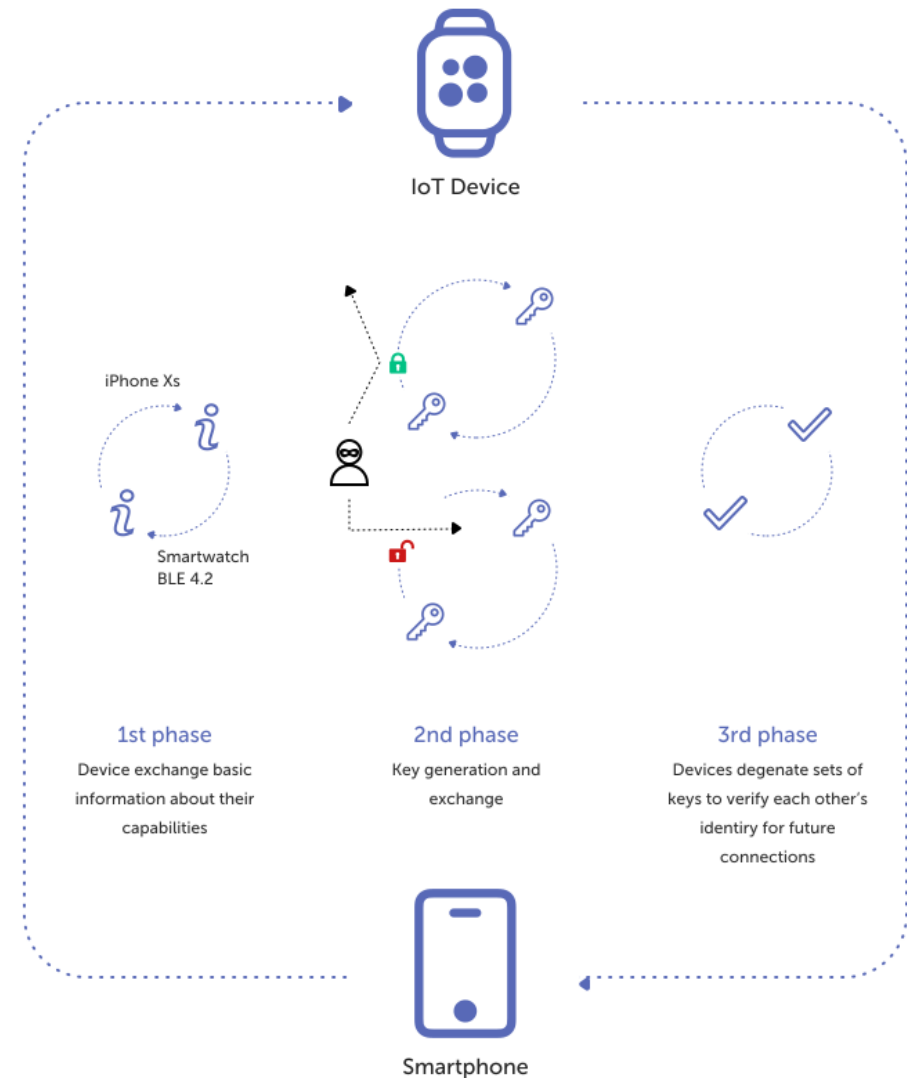


Gateway Required

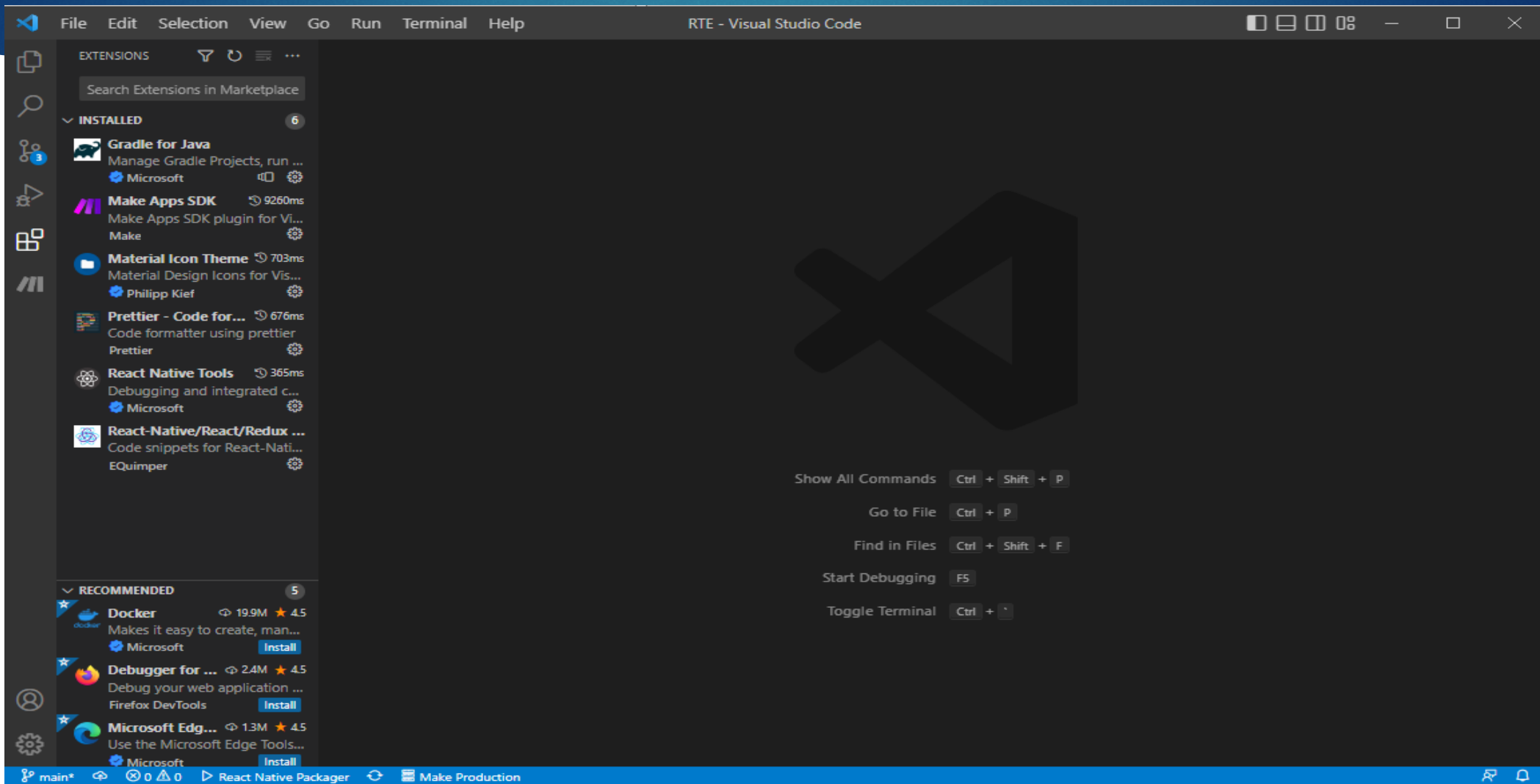
Процес з'єднання пристроїв BLE



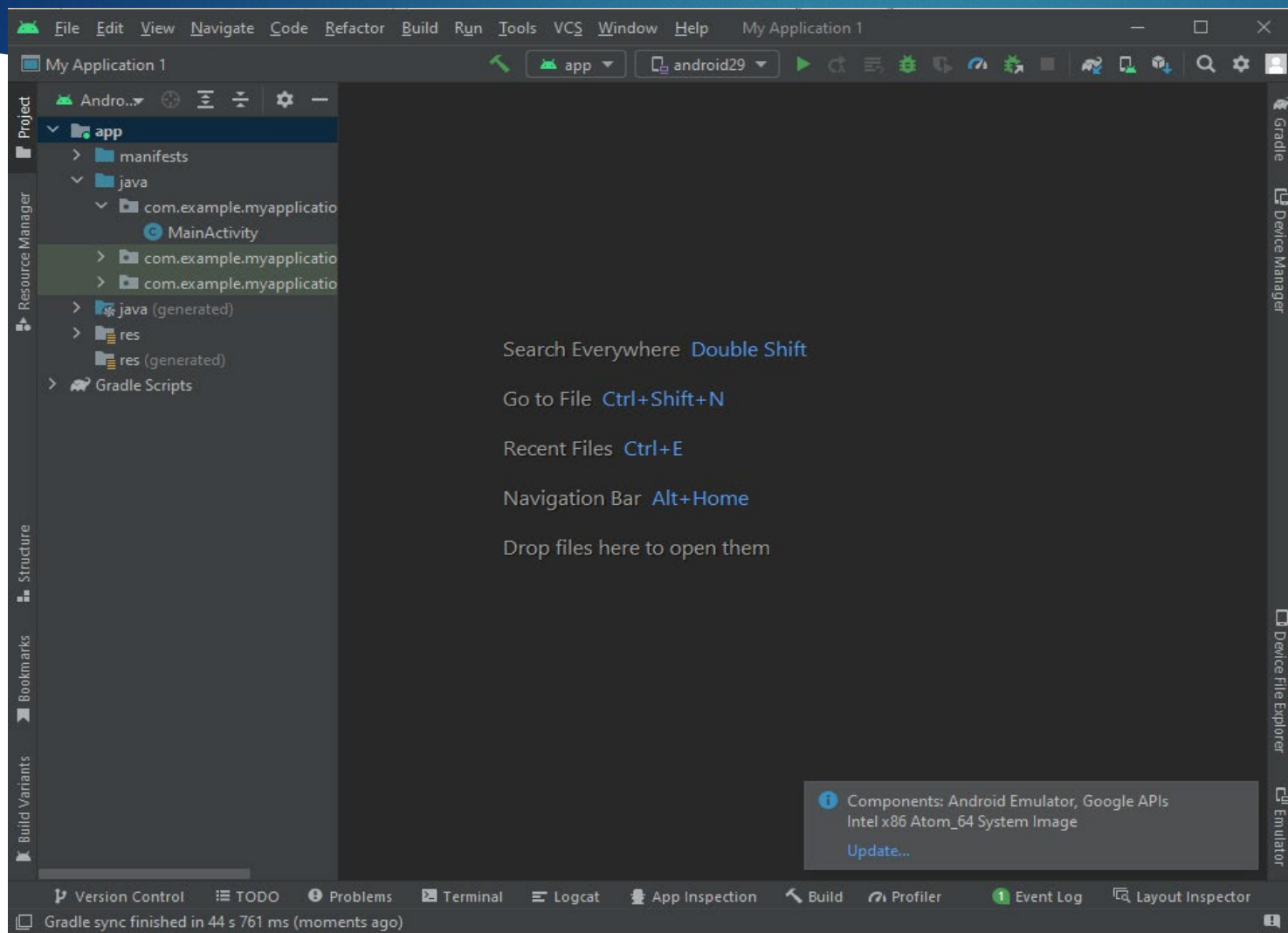
Pairing process



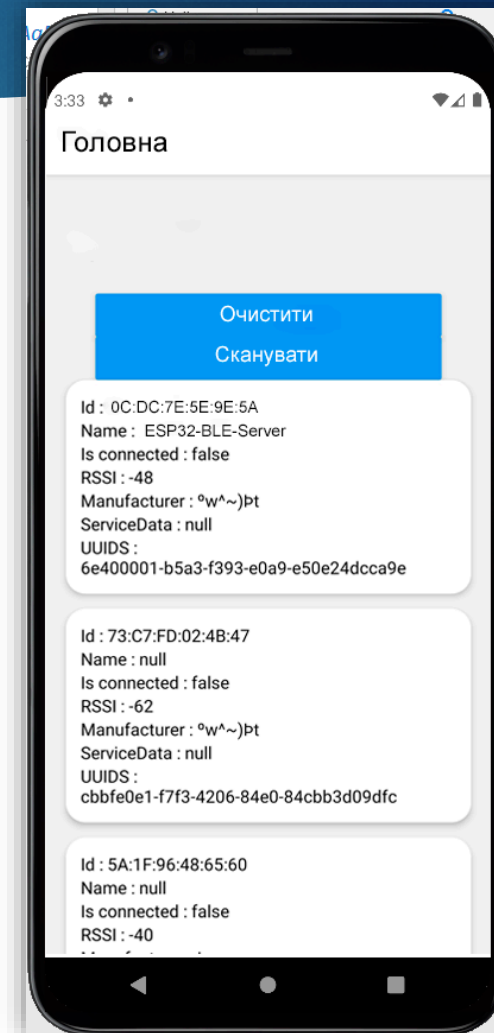
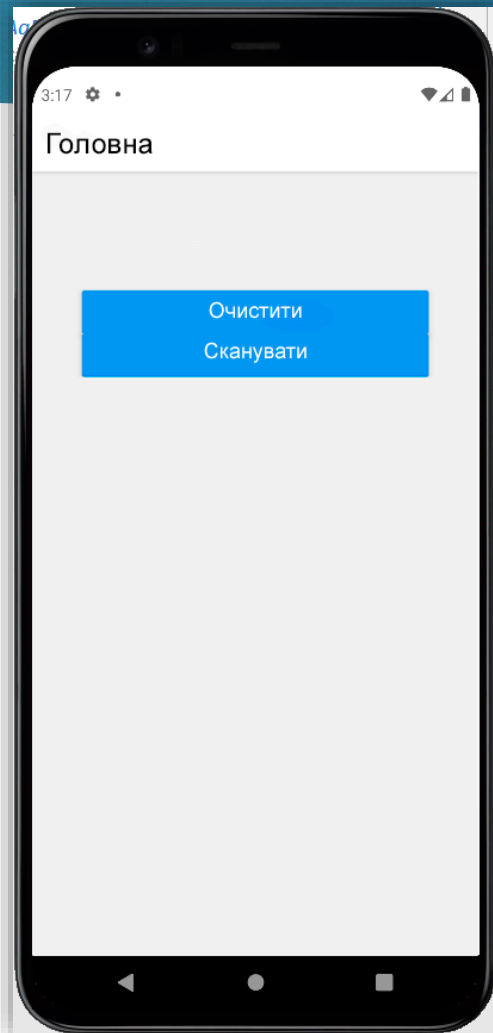
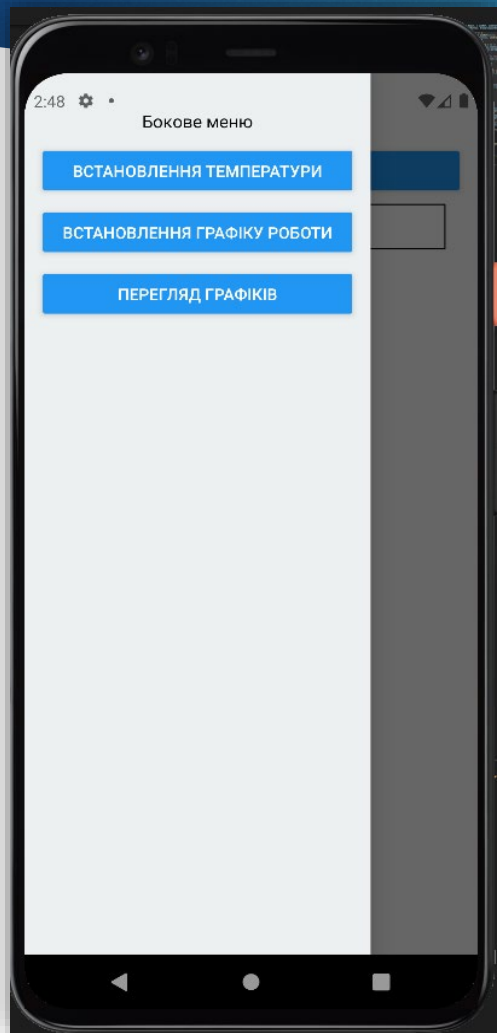
Середовище розробки Visual Studio Code



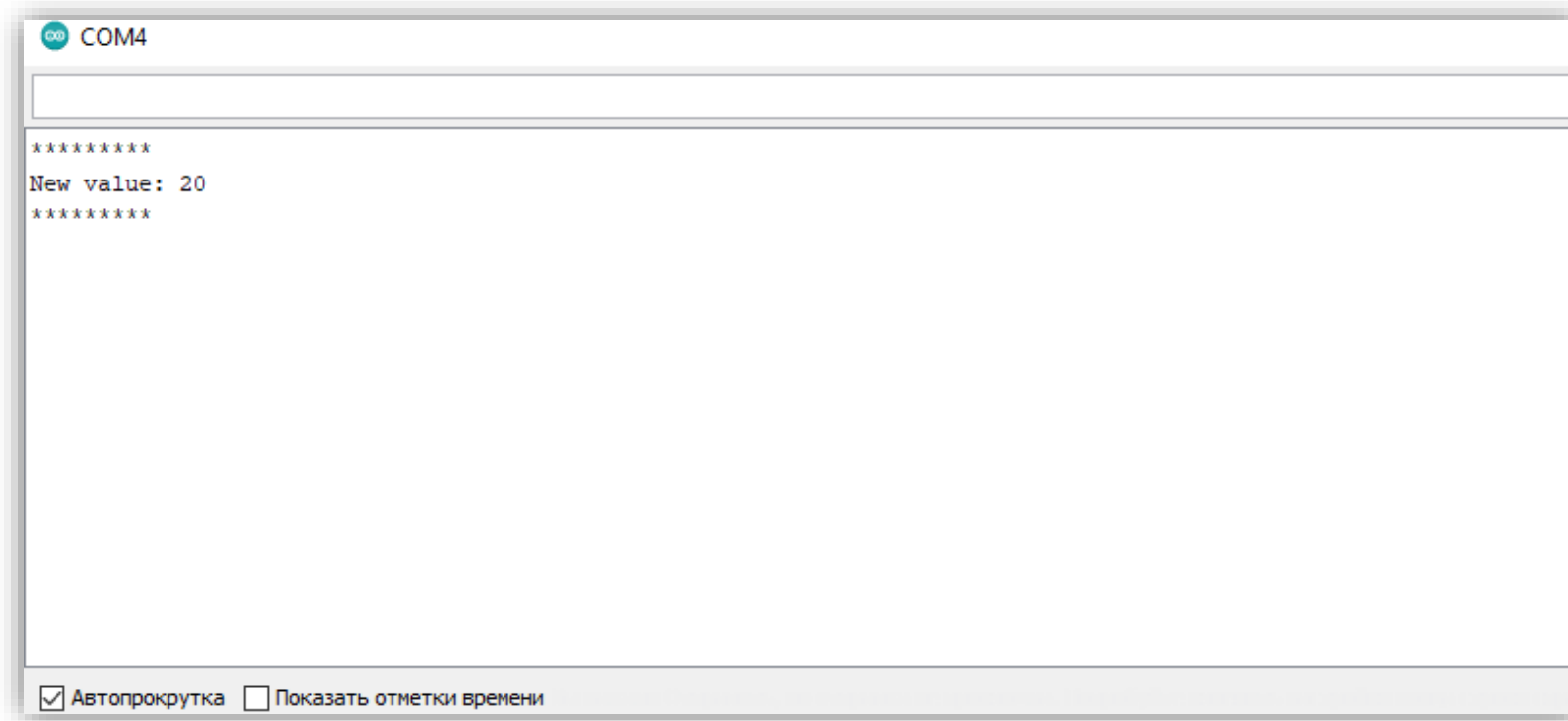
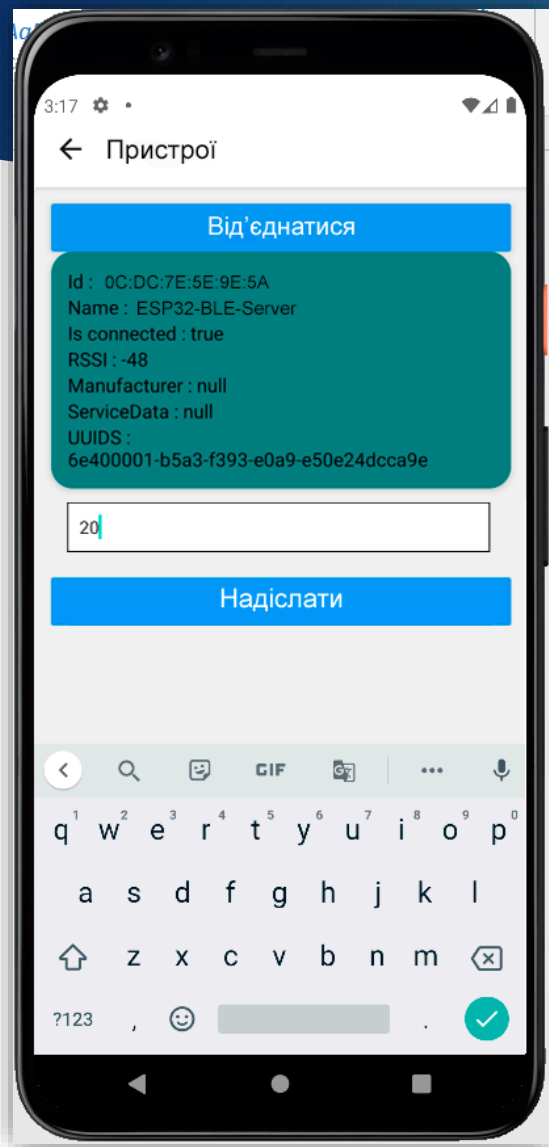
Середовище тестування Android Studio



ЗОВНІШНІЙ ВИГЛЯД МОБІЛЬНОГО ДОДАТКУ



Тестування програми



Економічна частина

- ▶ Витрати на розробку нового програмного продукту - 526500 грн.;
- ▶ Спрогнозований чистий приведений дохід - 12596979,79 грн.;
- ▶ Період окупності - 0,74 роки.

Висновки

- ▶ У магістерській роботі розглянуто переваги та недоліки основних мов програмування мобільних додатків;
- ▶ Оглянуто та порівняно кілька систем керування децентралізованим опаленням;
- ▶ Опрацьовано та обґрунтовано функціональність розроблюваного мобільного додатку;
- ▶ Розглянуто технології для створення додатків із використанням технології Bluetooth Low Energy;
- ▶ Розроблено та протестовано мобільний додаток для керування децентралізованою системою опалення;
- ▶ Проаналізовано розробку з економічної сторони.

Дякую за увагу

**ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: «Мобільний додаток для керування децентралізованою системою опалення»

Тип роботи: Магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ КСУ, ФШТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 99% Схожість 1%

Аналіз звіту подібності (відмітити потрібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку  Галушак А.В.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи  Ліщук А.Р.
(підпис) (прізвище, ініціали)

Керівник роботи  Дубовой В.М.
(підпис) (прізвище, ініціали)