

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему

**“ Дослідження та програмна реалізація системи управління
розумним будинком з впровадженням системи захисту від кібератак”**

Виконав здобувач вищої освіти
II курсу, групи КН22М-1
ОПП «Комп'ютерна інженерія»
спеціальності 122 «Комп'ютерні науки»
_____ Повзун В.А.
« ____ » _____ 2023р.

Керівник проекту
кандидат технічних наук, доцент
_____ Босько В.В.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь магістр
Галузь знань 12 “Комп’ютерні науки”
Спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерні науки”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
“ _____ ” _____ 20__ року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Повзуну Володимиру Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи управління розумним будинком з впровадженням системи захисту від кібератак*

2. Керівник роботи *Босько Віктор Васильович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 32-13 від 17.08.23

3. Строк подання роботи до захисту *04.08.2023 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є програмне дослідження та програмна реалізація систем управління розумним будинком*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання. 7. Економічна ефективність

2. Перегляд аналогічних існуючих систем. розробленої програми.

3. Опис і обґрунтування проектних рішень. 8. Заходи з охорони праці та техніки

4. Етапи програмування системи. безпеки.

5. Впровадження системи в промислову експлуатацію. 9. Висновки.

6. Наукова новизна

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна 1 аркуш

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

Діаграма процесів процесів 1 аркуш

Показники економічної ефективності 1 аркуш

6. Консультанти по роботі, із зазначенням розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В., к.т.н., доцент	09.11.2023 р.	17.11.2023 р.
Охорона праці	Оришака О.В., к.т.н., доцент	03.11.2023 р.	21.11.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	12.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	18.10.2023 р.	
3.	Розробка моделі компонента	23.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	32.10.2023 р.	
6.	Програмування алгоритмів	11.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	16.11.2023 р.	
9.	Оформлення ПЗ	18.11.2023 р.	
10.	Попередній захист роботи	04.12.2023 р.	

Дата видачі завдання
«__»_____20 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
«__»_____20 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Повзун В.А. Дослідження та програмна реалізація системи управління розумним будинком з впровадженням системи захисту від кібератак. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній магістерській роботі розроблено програмне забезпечення, яке призначено для системи управління розумним будинком з підсистемою безпеки передачі даних.

Метою розробки є дослідження та програмна реалізація системи управління розумним будинком з підсистемою безпеки передачі даних та захистом від кібератак.

Об'єктом дослідження є процес забезпечення управлінням «розумним будинком».

Предметом дослідження є методи забезпечення управління системами розумного будинку.

Методи дослідження базуються на методах теорії кодування, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи управлінням розумним будинком з підсистемою безпеки передачі управляючих команд.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10/11.

Програму розроблено в середовищі C++ та JS з використанням MySQL та PostgreSQL БД.

Ключові слова: комп'ютерна інженерія, відеонагляд, smart house.

ABSTRACT

Povzun V.A. Research and software implementation of a smart home management system with the introduction of a cyber attack protection system. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this master's thesis, the software was developed, which is intended for the management system of a smart home with a data security subsystem.

The purpose of the development is the research and program implementation of the smart home management system with the data security subsystem.

The object of the study is the process of providing smart home management.

The subject of the study is methods to ensure the management of smart home systems.

Research methods are based on methods of coding theory, methods of mathematical statistics, methods of software development.

The result of the work - software implementation of the smart home management system with the security control command transmission

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

User friendly interface developed. Instructions for working with software are given.

The program can be used on the PC architecture of IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in C ++ and JS using MySQL and PostgreSQL databases.

Keywords: computer engineering, CCTV, smart house.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, СИМВОЛІВ ТА СПЕЦІАЛЬНИХ ТЕРМІНІВ.....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	10
1.1 Призначення системи.....	7
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ.....	9
2.1 Огляд існуючих систем.....	9
2.2 Обґрунтування вибору методів розробки.....	30
2.3 Розгорнута постановка завдання	31
3 ОПИС І ОБґРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ.....	33
3.1 Опис функціонування системи.	33
3.2 Розробка структурної схеми	40
3.3 Розробка функціональної схеми.....	42
3.4 Розробка діаграми процесів.....	45
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ І ПРОГРАМНИХ РІШЕНЬ..	47
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	71
4.2 Захист розробленого програмного забезпечення.....	75
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	77
6 НАУКОВА НОВИЗНА	80
7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	81
7.1 Техніко економічне обґрунтування теми магістерської роботи	81
7.2 Розрахунок трудомісткості розробки програмної продукції.....	83
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	85

ВКРМ-122.23.0018.00.00.ПЗ				
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>
<i>Розроб.</i>		<i>Повзун В.А</i>		
<i>Перевір.</i>		<i>Босько В.В</i>		
<i>Н. Контр.</i>		<i>Коваленко А.С</i>		
<i>Затверд.</i>		<i>Смірнов О.А.</i>		
<i>Дослідження та програмна реалізація системи управління розумним будинком з впровадженням системи захисту від кібератак</i>				
		<i>Піт</i>	<i>Арк</i>	<i>Апквіліє</i>
		М	1	
ЦНТУ КН-22М-1				

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	89
7.5 Визначення собівартості розробки та ціни програмної продукції.	93
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.	97
7.7 Визначення експлуатаційних витрат.....	97
7.8 Визначення економічної ефективності програмної продукції.....	99
7.9 Висновки.	101
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ.....	102
8.1 Аналіз умов праці програміста.	102
8.2 Заходи профілактики при роботі з комп'ютерною технікою.	104
8.3 Розрахунок занулення глухозаземленої нейтралі.....	106
8.4 Висновки.	111
9 ОСНОВНІ ВИСНОВКИ.....	113
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	115

КБГІЗ - 2023

ПЕРЕЛІК СКОРОЧЕНЬ, СИМВОЛІВ ТА СПЕЦІАЛЬНИХ ТЕРМІНІВ

- GPS – (Global Positioning System) система для отримання місцезнаходження об'єкта.
- PKI – (Public Key Infrastructure) – інфраструктура відкритих ключів.
- GSM – (Group Special Mobile) – глобальна система мобільного зв'язку.
- SDL – (Software Development Life Circle) – методика створення програмних продуктів.
- RF – (Radio Frequency) – радіочастоти;
- Smart Switch – розумна розетка, пристрій для моніторингу та контролю електроприладів;
- Smart Grid – електрична мережа, яка включає в себе різноманітні оперативні та енергозберігаючі заходи, такі як розумні лічильники, відновлювані джерела енергії та ресурси забезпечення енергоефективності;
- ВДЕ – відновлювані джерела енергії.

ВСТУП

Актуальність теми. З початку нового тисячоліття людство крокує в епоху нових технологічних відкриттів, одним з яких є побутова автоматизація. Час сучасної людини має величезну цінність і такі системи автоматизації як "розумний будинок" істотно економлять цей життєво-важливий ресурс. Включити кондиціонер, вимкнути світло в вітальні, активувати нічну сигналізацію - це лише маленький перелік дій, які можна покласти на систему "розумного будинку". Але такі пристрої мають один мінус - велику ринкову вартість. Тому розробка відносно дешевої системи, з аналогічними можливостями отримує все більше актуальності.

З 2010 року такі системи як "розумний дім" отримали великого розголосу в сучасному суспільстві, бо вони допомагають заощадити пару важливих ресурсів людського життя - час і гроші. Інноваційні розробки подібного роду спрямовані не тільки на підвищення зручності життя, але і на поліпшення енергозбереження приміщень. Простим прикладом такої автоматизації служить освітлення вашого холодильника. Коли дверцята відкриті - освітлення активно, дверцята зачинені - освітлення вимикається. Таким чином можна виключити неефективне використання електроприладів і опалювальних систем, а якщо врахувати, що ціна на енергоресурси постійно зростає - це дозволить отримати істотну економію як енергії, так і грошових коштів.

Більш сучасні проекти будинку майбутнього передбачають наявність цілої системи модулів, розташованих по всьому будинку. Кожен пристрій є повноцінним комп'ютером, об'єднаним в загальну мережу. Практично кожен крок власника контролюється цією системою. Завдяки пристрою можна налаштувати навколишнє оточення на свій смак і практично миттєво змінити інтер'єр. Ще один варіант передбачає практично повне виключення людини від управління харчовими запасами. Все здійснюється автоматично.

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
						4
Вим.	Арк.	№ докум.	Підпис	Лат		

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи управління розумним будинком з підсистемою безпеки передачі даних.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- огляд існуючих систем управління розумним будинком з підсистемою безпеки передачі даних;
- дослідження системи систем управління розумним будинком;
- програмна реалізація системи управління розумним будинком з підсистемою безпеки передачі даних.

Об'єктом дослідження є процес управління розумним будинком.

Предметом дослідження є методи реалізації систем управління розумним будинком.

Методи дослідження базуються на методах теорії кодування, методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- удосконалено систему управління розумним будинком з впровадженням підсистеми безпеки передачі даних;
- проведено огляд технологій зв'язку в системах «Розумний дім»;
- розроблено вітчизняний продукт управління розумним будинком з підсистемою безпеки передачі даних, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі для реалізації систем управління технологіями розумний будинок.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		5

параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи управління розумним будинком з підсистемою безпеки передачі даних, є актуальною задачею, яка потребує вирішення у даній магістерській роботі.

КБГПЗ-2023

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1. Призначення системи

Системи "Розумний будинок" Smart Home (домашня автоматизація,) представляють собою один з перспективних напрямків розвитку інформаційних та комунікаційних технологій. Ці системи функціонально пов'язують всі електроприлади будівлі, якими можна керувати централізовано користувачем з пульта-дисплею або автоматично за допомогою певних алгоритмів.

Метою даної роботи є дослідження та програмна реалізація системи управління розумним будинком з підсистемою безпеки передачі даних та захисту від кібератак. Робота включає такі етапи:

- аналіз технологій і стандартів у галузі розумних будинків та безпеки передачі даних;
- розробка архітектури системи, що включатиме управління різними пристроями та підсистемами безпеки;
- реалізація програмного забезпечення для управління розумним будинком і забезпечення безпеки даних;
- тестування та валідація системи;
- впровадження системи у реальному середовищі та налагодження.

Ця робота спрямована на створення ефективної та безпечної системи управління розумним будинком, яка враховує сучасні вимоги технологічного розвитку та кібербезпеки.

1.2. Область застосування

Система управління розумним будинком з підсистемою безпеки передачі даних яка:

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		7

- реалізує основні концепції у сфері розумного будинку;
- має певний специфічний функціонал, що відсутній в інших системах;
- є простою, і дешевою і надійною за рахунок мінімалістичного інтерфейсу;
- легка для розгортання для будь-якого користувача;
- за замовчуванням працює в автономному режимі, але може бути налаштована на роботу з хмарою.

Отже, виходячи з вищевказаного, дослідження та програмна реалізація системи управління розумним будинком з підсистемою безпеки передачі даних (захистом від кібератак) є актуальною задачею, яка потребує вирішення у даній магістерській роботі.

КБГПЗ-2023

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1. Огляд існуючих систем

Smart Home (розумний будинок або будівля) представляє собою приміщення, будинок або комерційну споруду, де електричні пристрої мають функціональний зв'язок один з одним. Вони можуть бути підключені до комп'ютерної мережі, що дозволяє їх управління з комп'ютера та забезпечує віддалений доступ через Інтернет. Інтеграція інформаційних технологій у домашніх умовах дозволяє системам та пристроям координувати виконання функцій, враховуючи задані програми та зовнішні умови.

Створення розумного будинку вимагає професійного планування та програмування від компаній, що спеціалізуються на розробці проектів розумних будинків. Програми для алгоритмів багатокімнатного розумного будинку залежать від потреб мешканців та ситуацій, пов'язаних зі змінами середовища чи безпеки. Розумний будинок відзначається можливістю дистанційного керування, де людина може створити певну обстановку одним натисканням кнопки. Система, аналізуючи навколишнє середовище та параметри кімнати, самостійно виконує команди програміста з відповідними налаштуваннями. Побутову техніку, встановлену в розумному будинку, можна об'єднати в домашню мережу Universal Plug'n'Play з доступом в Інтернет.

Незважаючи на потенційні переваги розумного будинку, такі технології мають серйозні недоліки і ризики. Основні з них включають:

- чутливість систем підключення до Інтернету до останнього доступу;
- новизну технології, яка все ще перебуває на початковій стадії розвитку.

Існує багато спеціалізованих платформ і протоколів для локальних мереж "розумного будинку", які взаємодіють з пристроями за різними мовами. Багато з них, на жаль, не є відкритими, але зараз починають з'являтися системи з

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
						9
Вим.	Арк.	№ докум.	Підпис	Лат		

відкритим кодом для використання з різноманітним обладнанням. Схематично типову схему інтеграції таких систем наведено на рисунку 1.1.

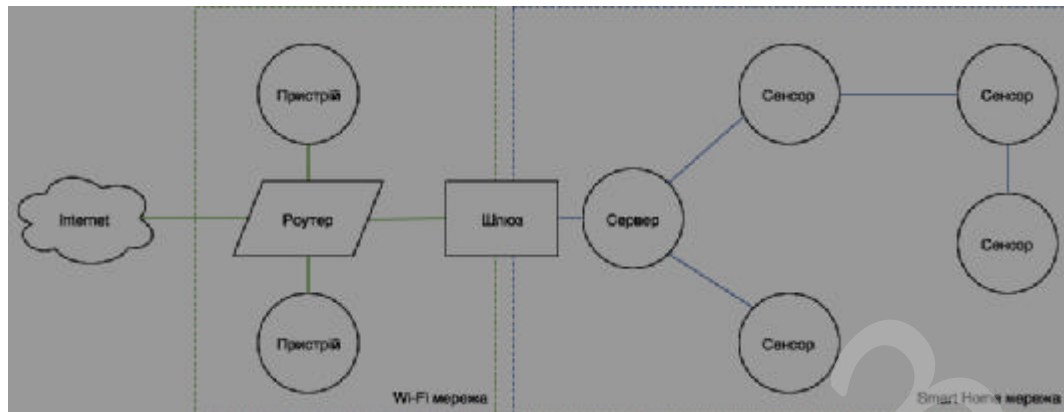


Рисунок 1.1 – Типова структура Smart Home системи

В додаток до вищезгаданих спеціалізованих протоколів іноді використовуються стандартні технології, такі як Wi-Fi або Bluetooth. Це вирішує проблеми функціональності та комплексного управління системою, підтримує архітектуру системи і ставить нові виклики щодо безпеки, енергоспоживання та надійності. Топології мереж, які використовують ці протоколи, також важливі, оскільки в умовах щільної забудови або наявності бетонних конструкцій у будівлі теорія "кріплення" може обмежувати їх діапазон дії.

Надалі буде надано детальний аналіз найпопулярніших комунікаційних протоколів, які використовуються в більшості систем "Розумний дім" сьогодні: ZigBee, WiFi і Bluetooth.

Протокол ZigBee

Створення альянсу ZigBee. Альянс ZigBee (ZigBee Alliance) є глобальною організацією, яка об'єднує розробників та виробників для сприяння розвитку та впровадженню технології ZigBee.

Процес розробки стандарту включає в себе налаштування альянсу, виробництво специфікацій, профілів програмного забезпечення та інших

документів, необхідних для розробки та виробництва пристроїв, що відповідають стандарту.

Характеристики мереж ZigBee:

Самоорганізація. Процес самоорганізації дозволяє пристроям ZigBee автоматично знаходити один одного та формувати мережу без необхідності в ручному втручанні.

Самовідновлення. Мережі ZigBee можуть самовідновлюватися після виникнення збоїв. Кожен пристрій може встановлювати нові маршрути для передачі повідомлень у випадку втрати з'єднання з іншими вузлами.

Загалом, технологія ZigBee має широкий спектр застосувань, зокрема в розумних домах, де важливо мати мережі, які можуть ефективно організовуватися та відновлюватися без значного втручання користувача.

ZigBee - це стандарт для протоколів зв'язку раннього доступу з використанням невеликих малопотужних цифрових приймачів, заснований на стандарті IEEE 802.15.4-2006 для бездротових персональних інтерфейсів, таких як, наприклад, бездротові гарнітури, які підключаються до мобільних телефонів через короткі хвили. півхвили. Технологія визначається специфікацією ZigBee, яка була створена, щоб бути тихішою та дешевшою, ніж інші власні мережі, такі як Bluetooth. ZigBee призначений для мобільних додатків, де необхідний захист акумулятора та безпека передачі даних на кордоні.

ZigBee Alliance - це організація, яка надає та публікує стандарти ZigBee, а також публікує профілі додатків, які дозволяють OEM-виробникам створювати розумні продукти. Поточний список профілів програми, опублікованих або вже в розробці:

- домашня автоматизація;
- раціональне використання енергії (ZigBee Smart Energy 1.0/2.0);
- автоматизація комерційного будівництва;
- телекомунікаційні програми;
- персональне спостереження;

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		11

– дистанційне управління.

Відносини між IEEE 802.15.4 і ZigBee аналогічні відносинам між IEEE 802.11 і Wi-Fi Alliance.

Специфікація ZigBee 1.0 була ратифікована 14 грудня 2004 року і стала доступною для членів ZigBee Alliance. Проте лише 30 жовтня 2007 року була опублікована специфікація ZigBee 2007. Перший профіль додатка ZigBee "Home Automation" був опублікований 2 жовтня 2007 р.

ZigBee працює в діапазонах радіочастот для риболовлі, науки та медицини (діапазон ISM): 868 МГц у Європі, 915 МГц у США та Австралії та 2,4 ГГц у більшості країн світу (під юрисдикцією більшості країн світу). Виробники чіпів ZigBee зазвичай поєднують радіо та мікроконтролер із флеш-пам'яттю від 60 Кб до 128 Кб. Падіомодуль також може бути розроблений окремо з будь-яким процесором і мікроконтролером. Зазвичай, виробники модулів пропонують програмне забезпечення ZigBee, хоча доступні інші незалежні ресурси. Оскільки ZigBee може прокинутися (тобто перейти зі сну в активний режим) за 30 мілісекунд або менше, затримка відлуння може бути дуже мала, особливо в поєднанні з Bluetooth, для якого затримка, яку він створює під час переходу зі сну в активний режим, зазвичай досягає секунд. Оскільки ZigBee проводить більшу частину часу в режимі сну, енергоспоживання може бути дуже низьким, завдяки чому досягається тривалий термін служби батареї. Перший випуск схеми наразі відомий як ZigBee 2004. Другий випуск називається ZigBee 2006 і в новій версії замінює протокол MSG/KVP, написаний у ZigBee 2004, на "clact libraries eriv". Існує три різних типи пристроїв ZigBee:

Координатор ZigBee (ZC) - це головний пристрій, який визначає основну топологію деревовидної мережі і може взаємодіяти з іншими мережами. У кожній мережі повинен бути принаймні один координатор ZigBee, який починає мережу. Він зберігає інформацію про мережу та виконує функції довіреного центру та сховища секретних паролів.

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		12

Маршрутизатор ZigBee (ZR) - цей маршрутизатор може виступати як проміжний маршрутизатор, передаючи дані від інших пристроїв. Він також може запускати додатки.

Кінцевий пристрій ZigBee (ZED) - його функціональні можливості дозволяють йому обмінюватися інформацією з вузлом вищого рівня (координатором або з маршрутизатором), але він не може передавати дані від інших пристроїв. Такі функціональності дозволяють вузлу перебувати в сплячому стані протягом значної частини часу, що дозволяє економити енергопотребу батареї. ZED потребує мінімальної кількості пам'яті, і тому він може бути більш доступним у виробництві, ніж ZR чи ZC.

Приклад класичної Zig -Bee мережі з інтеграцією з дротовою мережею Ethernet показаний на рисунку. 1.2.

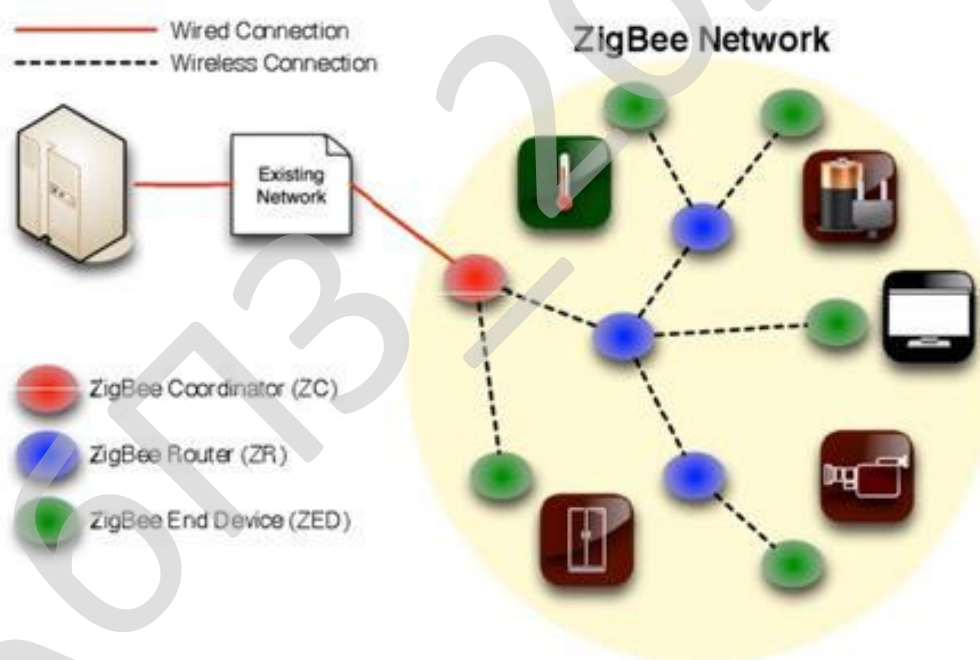


Рисунок 1.2 – Мережа ZigBee інтегрована з мережею Ethernet

На завершення сформулюємо основні переваги та недоліки технології ZigBee.

Основні переваги технології ZigBee

Переваги

- низьке енергоспоживання. Вузли ZigBee можуть працювати довгий час в режимі сну, ефективно використовуючи енергію батарей;
- одна велика подія. Збільшена дальність передачі даних може бути досягнута за рахунок створення однієї великої мережі;
- швидкий вихід із режиму зчеплення. Прискорений процес встановлення з'єднання під час ініціалізації;
- підтримка різних топологій. Здатність до роботи в граничних топологіях, включаючи самовідновлення мережі при втраті зв'язку між компонентами.

Недоліки

- відсутність адаптерів ZigBee в працюючих комп'ютерах, смартфонах тощо: Обмежена підтримка пристроїв іншими технологіями;
- необхідність встановлення адаптерів або створення шлюзу з Wi-Fi: Потребує додаткового обладнання для взаємодії з іншими мережами;
- низький рівень стандартизації та унікальна апаратно-програмна платформа для тестування додатків. Специфікації можуть відрізнятися між виробниками, що може створювати проблеми з сумісністю;
- низька швидкість передачі даних. Загальна швидкість передачі обмежена, особливо порівняно з Wi-Fi;

Wi-Fi: Wi-Fi використовується для передачі цифрових даних через радіоканали. У сфері розумного дому він використовується разом з ZigBee, проте має свої переваги:

Переваги Wi-Fi:

- висока швидкість передачі даних. Підтримує швидкості понад 100 Мбіт/с, що забезпечує ефективний доступ до Інтернету та стрімінг великих об'ємів даних;

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		14

- можливість роботи в топології прикордонного мосту. Дозволяє користувачам взаємодіяти з різними точками доступу для покриття великих просторів.

Обидві технології мають свої застосування, і їх використання залежить від конкретних потреб і вимог конкретного розумного будинку чи офісу.

Структура звичайної Wi-Fi мережі з доступом до Інтернету показана на рисунку 1.3.

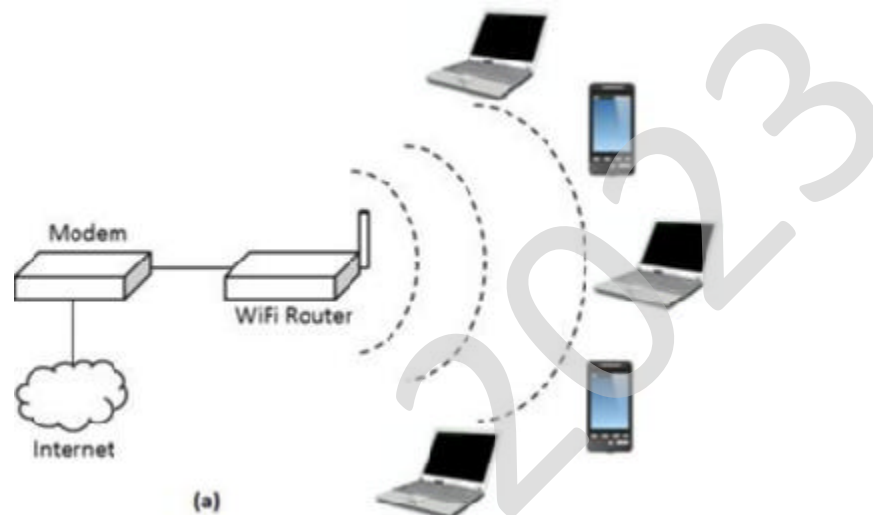


Рисунок 1.3 – Структурна схема звичайної Wi-Fi мережі

Технологія Wi-Fi Direct відкриває нові можливості для прямого безпосереднього з'єднання між пристроями, обійшовши необхідність у точці доступу чи маршрутизаторі. Ось кілька ключових особливостей та переваг Wi-Fi Direct:

Особливості Wi-Fi Direct

- пряме з'єднання. Wi-Fi Direct дозволяє пристроям підключатися один до одного безпосередньо, обходячи існуючі точки доступу чи маршрутизатори;
- масштабованість. Мережу Wi-Fi Direct можна легко розширити, додаючи нові пристрої до існуючого безпосереднього з'єднання;

- просте підключення. Встановлення з'єднання подібне до процесу Bluetooth, що робить його зручним і досить інтуїтивним;
- широкий спектр застосувань. Застосовується для різноманітних сценаріїв, включаючи обмін файлами, відео- або аудіо-потоками, спільні ігри і т.ін.;
- сумісність з Wi-Fi: Якщо пристрій підтримує Wi-Fi Direct, він може з'єднуватися з іншими пристроями, які також підтримують цю технологію.

Переваги Wi-Fi Direct

- зручність та простота використання: Простий процес налаштування і з'єднання, схожий на той, який використовується в Bluetooth;
- широкий охоплення. Забезпечує до 100 метрів передачі даних, що дозволяє використовувати його на великих відстанях;
- безпосередність підключення. Дозволяє швидко обмінюватися даними між пристроями навіть без наявності Інтернет-підключення.

Така технологія відкриває нові перспективи для безпосереднього з'єднання різних пристроїв та може бути використана в різних областях, включаючи розумний дім.

Типи підключень Wi-Fi Direct показано на рисунку 1.4.



Рисунок 1.4 – Типи з'єднань Wi-Fi Direct

Точка доступу передає свій мережевий ідентифікатор (SSID) за допомогою спеціальних пакетів сигналу зі швидкістю 0,1 Мбіт/с кожні 100 м. Таким чином, 0,1 Мбіт/с є найнижчою швидкістю передачі даних для Wi-Fi. Знаючи SSID мережі, клієнт може визначити, чи можливе підключення до цієї точки доступу. Коли входите в зону покриття двох точок доступу з ідентичними SSID, приймач може перемикатися між ними на екрані даних для рівня сигналу. Стандарт Wi-Fi надає клієнту повну свободу у виборі провайдера для підключення.

Швидкість передачі даних (SSID Broadcast)

- швидкість передачі. Виправлення швидкості передачі даних для SSID не обов'язково обмежена 0,1 Мбіт/с. Справжня швидкість передачі даних залежить від стандарту Wi-Fi (наприклад, 802.11b, 802.11g, 802.11n, 802.11ac), який підтримується пристроєм;

- SSID Broadcast: SSID (Service Set Identifier) не транслюється окремими пакетами із стандартною швидкістю передачі даних. Замість цього, інформація про SSID включається в управлінські кадри та асоційовані кадри, які використовуються для встановлення та управління з'єднанням.

Перемикання між точками доступу (Roaming)

- клієнт може визначити рівень сигналу та перемикається між точками доступу для оптимізації з'єднання. Це важливо для забезпечення стійкості та найкращої продуктивності у бездротових мережах;

- вибір провайдера. Зазначений момент про повну свободу вибору провайдера для підключення відображає гнучкість і сумісність Wi-Fi. Користувачі можуть вибирати між різними мережами залежно від їх уподобань та доступності;

- зони Wi-Fi та підтримка декількох комп'ютерів. Wi-Fi дійсно дозволяє підключати кілька пристроїв до точки доступу, що сприяє об'єднанню декількох пристроїв у мережі та обміну інформацією між ними;

- відстань передачі та перешкоди. Відстань передачі даних залежить від потужності передавача, але також може бути вплинута архітектурою приміщення та наявністю перешкод, таких як стіни чи металеві конструкції, які можуть призвести до зниження сигналу.

Загалом Wi-Fi - це потужна технологія для передачі даних в бездротових мережах, яка знаходить широке застосування в різних сферах, включаючи розумний будинок.

Розумний будинок і технологія Wi-Fi

- можливість підключення декількох пристроїв. Зони Wi-Fi дозволяють користувачам підключати свої пристрої до точок доступу, створюючи комфортне середовище для з'єднання декількох комп'ютерів та інших розумних пристроїв в одній мережі;

- вплив на відстань передачі. Відстань передачі інформації у мережі Wi-Fi залежить від кількох факторів, таких як потужність передавача, наявність перешкод та характеристики антени. Це визначає межі покриття конкретного пристрою або точки доступу;

- регулювання програмної потужності. Деякі моделі обладнання дозволяють програмне регулювання потужності передавача, що може бути корисним для оптимізації покриття та витрат енергії в мережі.

Це важливі аспекти для розумного будинку, де з'єднання між пристроями та якість сигналу грають критичну роль в забезпеченні ефективності та надійності роботи різних розумних систем та пристроїв.

Переваги та недоліки Wi-Fi в розумному будинку

Переваги:

- широкий розповсюджений доступ. Wi-Fi роутери легко доступні та розповсюджені, що забезпечує простоту налаштування та співпрацю з різними пристроями;

- можливість роботи в різних режимах. Робота в режимах Ad-hoc і Wi-Fi Direct дозволяє безпосереднє з'єднання пристроїв без необхідності

маршрутизатора або модему, що полегшує підключення пристроїв в розумному будинку;

- високий рівень стандартизації. Стандартизована архітектура Wi-Fi забезпечує сумісність та співпрацю між різними виробниками, що спрощує інтеграцію різноманітних пристроїв.

Недоліки:

- високе енергоспоживання. Wi-Fi споживає більше енергії порівняно з іншими бездротовими технологіями, такими як Bluetooth або ZigBee, що може впливати на тривалість роботи батарей живлення пристроїв;

- складність налаштування топології мережі. Налаштування топології мережі може вимагати додаткових заходів, і в деяких випадках потребує використання спеціальних пристроїв для підтримки оптимальної структури мережі;

- велика кількість енергії. Використання Wi-Fi може вимагати значної кількості енергії, що може впливати на роботу батарейного живлення розумних пристроїв;

- надмірна швидкість передачі даних. Швидкість передачі даних Wi-Fi може бути занадто великою для деяких застосувань у розумному будинку, де великий обсяг даних не завжди є необхідним;

- залежність від потужності пристроїв. Радіус дії та швидкість передачі даних Wi-Fi можуть бути обмеженими потужністю пристроїв, що може впливати на охоплення та стабільність сигналу в розумному будинку.

Bluetooth

Bluetooth є важливою технологією для розумного будинку, пропонуючи ряд переваг у порівнянні з іншими бездротовими протоколами. Давайте розглянемо деякі ключові аспекти та характеристики Bluetooth в контексті Smart Home:

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		19

- низька потужність. Bluetooth відомий своєю низькою споживаною енергією, особливо в режимі очікування. Це робить його ідеальним для використання в розумних пристроях, які працюють на батарейках;

- зручна топологія. Технологія Bluetooth дозволяє легко встановлювати зв'язок між пристроями, що спрощує створення мережі в розумному будинку без складних конфігурацій;

- висока швидкість передачі даних. Bluetooth забезпечує високу швидкість передачі даних, що робить його ефективним для передачі великих обсягів інформації, таких як відео чи аудіо дані;

- шифрування та аутентифікація. Bluetooth використовує шифрування та механізми аутентифікації для забезпечення безпеки передачі даних між пристроями;

- частотний перехід. Принцип роботи за технологією Frequency-Hopping Spread Spectrum (FHSS) дозволяє уникати перешкод та забезпечує надійну комунікацію в забруднених радіочастотних середовищах;

- Bluetooth Mesh. Стандарт Bluetooth Mesh дозволяє створювати великі мережі пристроїв у розумному будинку, розширюючи зону покриття та підтримуючи багато пристроїв в одній мережі;

- широке застосування. Bluetooth застосовується в різних типах пристроїв для розумного будинку, включаючи освітлення, датчики, аудіосистеми, замки та інші;

- Bluetooth SIG. Special Interest Group (SIG) забезпечує подальший розвиток технології Bluetooth та створення стандартів, що підтримують інновації у сфері розумних пристроїв.

Узагальнюючи, Bluetooth є важливою складовою для побудови надійних та зручних систем в розумному будинку, забезпечуючи зв'язок між різними пристроями з низьким споживанням енергії та високою швидкістю передачі даних.

Загальна електрична схема показана на рисунку 1.5.

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		20

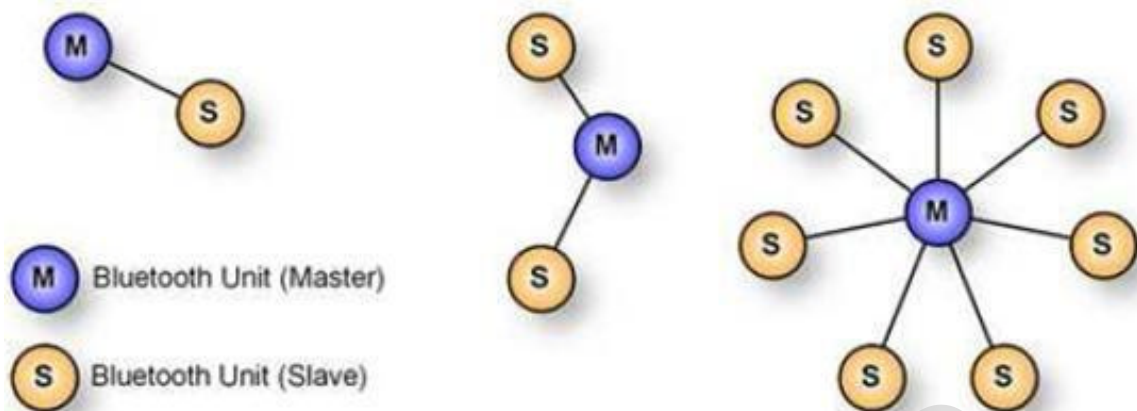


Рисунок 1.5 – Мережа Bluetooth Piconet

У кожній мережі один принтер працює як активний (master), а інші як пасивний (slave). Активний принтер визначає схему, за якою будуть працювати всі пасивні принтери його краю, і синхронізує свою роботу. Стандарт Bluetooth передбачає з'єднання незалежних і навіть несинхронізованих пристроїв (до 10) в так звану «скатернет». Для цього кожна пара пристроїв повинна мати хоча б одну повну копію, яка в одному буде активною, а в іншому пасивною. Отже, до інтерфейсу Bluetooth в межах зони розсіяної мережі можна одночасно підключити максимум 1 пристрій, однак ніхто не обмежує використання шлюзових пристроїв, які копіюють той самий Інтернет на більші відстані підключення.

Загальна гранична схема «мережі розсіювання» показана на рисунку. 1.6.

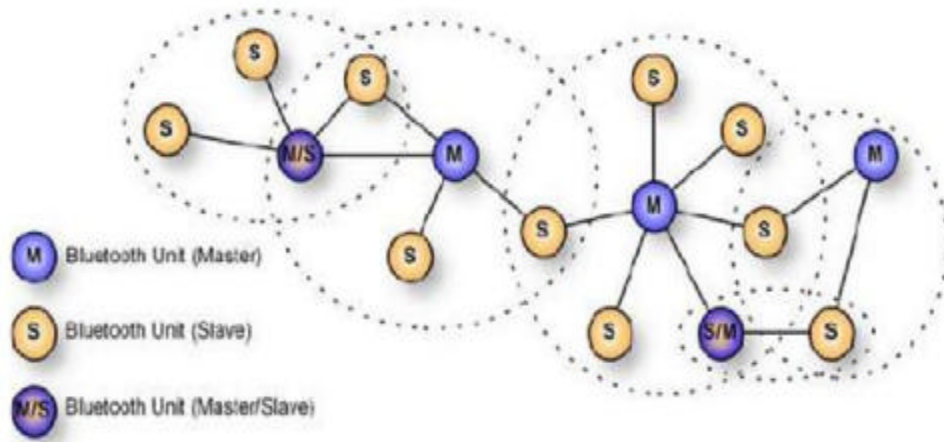


Рисунок 1.6 – Мережа Bluetooth Scatternet

Bluetooth Low Energy (BLE), також відомий як Bluetooth Smart, є технологією передачі даних з низьким енергоспоживанням, спроектованою для забезпечення безперервного зв'язку між різними пристроями при низькому споживанні енергії. Ця технологія входить в стандарт Bluetooth версії 4.0 і вище. Основною метою BLE є забезпечення ефективного використання енергії, що робить його ідеальним для пристроїв IoT, а також в розумних домах. Характеристики Bluetooth Low Energy:

- енергоефективність: Однією з основних переваг BLE є його низьке споживання енергії. Це дозволяє пристроям працювати на батареях протягом довгого часу, що робить його ідеальним для пристроїв, які працюють в умовах обмеженого джерела живлення;
- безперебійне підключення: BLE дозволяє пристроям підтримувати постійний зв'язок з іншими пристроями, не витрачаючи при цьому значної кількості енергії. Це особливо важливо для пристроїв, які повинні надсилати або отримувати дані на регулярній основі;
- підтримка коротких передач даних: BLE призначений для відправлення коротких пакетів даних, таких як вимірювання датчиків або інші компактні

інформаційні дані. Це дозволяє оптимізувати енергоспоживання і забезпечити ефективну передачу даних;

- використання в розумних домах та IoT: BLE широко використовується в розумних домах та інтернеті речей (IoT) через свою здатність працювати з енергоефективністю і підтримувати безперебійний зв'язок з різними пристроями.

У контексті розумного дому BLE може використовуватися для з'єднання різноманітних пристроїв, таких як сенсори, освітлення, замки, термостати та інші, з мобільними пристроями або між собою для обміну даними і керування. Процес з'єднання в Bluetooth LE зображено на рисунку. 1.7.

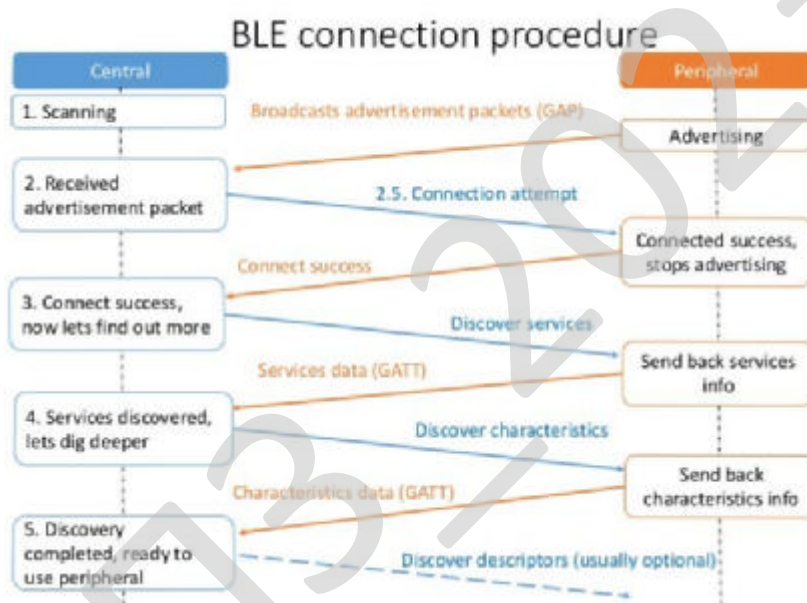


Рисунок 1.7 – Процес встановлення з'єднання Bluetooth

Схема стандартів Bluetooth представлена на рисунку 1.8

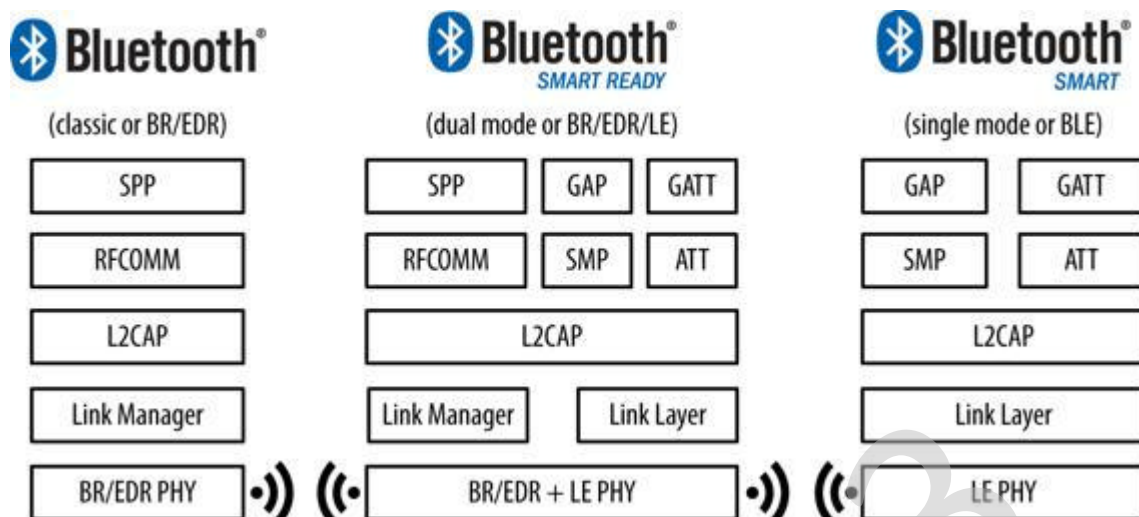


Рисунок 1.8 - Схема стандартів Bluetooth

Основні блоки Bluetooth-пристроїв та їх компоненти:

- Додаток. Реалізує логіку роботи для кінцевого користувача. У контексті Bluetooth він відповідає за взаємодію з користувачем та виконання конкретних функцій або сервісів.
- Головний пристрій (Хост). Надає верхні рівні стеку протоколів Bluetooth. Містить різні протоколи, серед яких основні:
 - GAP (Generic Access Profile): Профіль загального доступу, визначає правила з'єднання із пристроями, такі як режими видимості та методи взаємодії.
 - GATT (Generic Attribute Profile): Профіль загальних атрибутів, визначає способи представлення та обміну даними між Bluetooth-пристроями.
 - L2CAP (Logical Link Control and Adaptation Protocol): Протокол логічного з'єднання та адаптації, забезпечує послуги нижчих рівнів стеку протоколів.
 - ATT (Attribute Protocol): Протокол атрибутів, використовується для обміну даними про атрибути (характеристики) між Bluetooth-пристроями.
 - SM (Security Manager): Менеджер безпеки, відповідає за реалізацію процедур забезпечення безпеки в Bluetooth-пристроях.

- HCI (Host Controller Interface): Інтерфейс хост-контролера, частина на стороні хоста, яка дозволяє контролеру та хосту обмінюватися командами та даними.

с) Контролер: Займається нижніми рівнями Bluetooth. Включає в себе протоколи:

- HCI (Host Controller Interface): Інтерфейс хост-контролера, частина на стороні контролера, яка дозволяє взаємодіяти з хостом.

- LL (Link Layer): Рівень з'єднання, відповідає за управління каналами зв'язку між Bluetooth-пристроями.

- PHY (Physical Layer): Фізичний рівень, визначає характеристики апаратної частини Bluetooth-зв'язку, такі як модуляція та частотний діапазон.

Комерційні продукти зазвичай використовують один із таких апаратних сценаріїв:

- SoC - це універсальна система, яка поєднує додаток, пристрій і плату керування. Використовується в компактних недорогих пристроях, таких як датчики;

- рішення на двох мікроконтролерах, коли додаток і пристрій підключаються до контролера за допомогою UART, USB, SDIO. Таку конфігурацію можна використовувати, наприклад, у мобільних додатках;

- двоетапне рішення, в якому програма підключається до комунікаційного пристрою (host і controller) відповідно до чотириетапного протоколу.

Bluetooth 5.0

- збільшення дальності і швидкості передачі даних. Дальність передачі даних у Bluetooth 5.0 була збільшена в 4 рази порівняно з попереднім поколінням. Швидкість передачі даних також подвоїлася, що робить Bluetooth 5.0 більш швидким і потужним для передачі великої кількості інформації;

- стабільність і безпека. Оновлений стандарт володіє вищим рівнем стабільності та захисту даних, що робить його більш відповідальним для застосувань, де важлива надійність та конфіденційність;

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		25

- Mesh-топология. Додана підтримка mesh-топології розширює можливості стандарту, дозволяючи створювати мережі, в яких кожен пристрій може взаємодіяти з іншими. Це важливо для розумних будинків і промислових застосувань;

- обробка даних для "маяків". Bluetooth 5.0 підтримує обробку даних для "маяків", що дає можливість використовувати цю технологію для створення місцевих сценаріїв та розумних рішень в просторах, де простеження місця є важливим (наприклад, магазини);

- сумісність і доступність. - продукти з підтримкою Bluetooth 5 стали доступними на ринку, зокрема смартфони, такі як Galaxy S8, які використовують цей стандарт для забезпечення швидкої і ефективної передачі даних;

Ці поліпшення роблять Bluetooth 5.0 дуже привабливим для розумних пристроїв, мереж Інтернету речей (IoT) і різних інших застосувань, які вимагають високої продуктивності та забезпечення безпеки та стабільності зв'язку. На закінчення сформулюємо

Переваги технології Bluetooth LE

Високий рівень стандартизації та сумісність. Bluetooth LE має високий рівень стандартизації, що дозволяє йому бути сумісним із різноманітними пристроями та платформами.

Низька вартість Bluetooth LE є економічним в плані вартості реалізації, що робить його привабливим для виробників пристроїв.

Ультранизьке енергоспоживання є однією з основних переваг є низьке енергоспоживання, що робить його ідеальним для пристроїв з обмеженим джерелом живлення, таких як датчики та носимі пристрої.

Швидкість передачі даних Bluetooth LE може досягати швидкості передачі даних більше 1 Мбіт/с, що дозволяє ефективно обмінюватися інформацією. Підгонка продуктивності модуля. Модуль може налаштовуватися в залежності від потреб, що дозволяє збільшити швидкість або зменшити радіус дії в залежності від конкретного використання.

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		26

Процес з'єднання простий, що полегшує використання та підключення пристроїв.

Bluetooth LE забезпечує захист зв'язку, що робить його безпечним для використання в різних сценаріях.

Наявність уніфікованих API для роботи з різними платформами полегшує розробку та інтеграцію пристроїв.

Недоліки технології Bluetooth LE

Підтримка mesh-топології з'явилася тільки в останній версії Bluetooth 5.0, і ще не всі пристрої підтримують цю функцію.

Навіть з усіма покращеннями, швидкість Bluetooth LE все ще може бути обмеженою для деяких додатків, порівняно з іншими технологіями.

Загалом, Bluetooth LE є важливим стандартом для бездротового зв'язку в різних сценаріях, зокрема в розумних домах, здоров'ї та фітнесі, а також в інтернеті речей (IoT).

Концепції розумного будинку

У сучасних житлових будинках велика кількість технологічного обладнання, яке відповідає за комфорт, безпеку, комунікацію та зручність для мешканців. Ці системи взаємодіють між собою, легко управляються і інтегруються, надаючи будинку статус "розумного". Розумний будинок може автоматично підтримувати оптимальний мікроклімат, керувати освітленням та меблями, а також забезпечувати безпеку від пожеж і витоків. Він реагує на відсутність мешканців, забезпечуючи ефективне використання енергії та оптимальний режим освітлення. Розумний будинок також може повідомити власника про непередбачені події та допомогти віддалено контролювати та моніторити стан будинку, забезпечуючи власнику зручний та безпечний простір для життя.

Система інтелектуальної автоматизації

Розумний будинок представляє собою інтелектуальну систему автоматизації, яка управляє інженерними системами будівлі для забезпечення

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		27

комфорту та безпеки. Ця система орієнтується на поліпшення естетики зовнішнього вигляду будівель та має основні цільові завдання - створення комфортних умов для мешканців та забезпечення їхньої безпеки.

Система «Розумний дім» включає в себе різні засоби автоматизації, такі як управління освітленням, керування електроприводами, контроль часу, управління системою вентиляції, централізоване управління, домашній кінотеатр, мультиплекс, системи пожежної сигналізації, системи безпеки та протипожежного захисту, системи контролю доступу та інші. Ці засоби надають можливість централізованого контролю та інтелектуального управління в різних приміщеннях - будь то житлові, офісні чи громадські простори.

Користувачі мають можливість налаштувати індивідуальні параметри свого оточення, такі як освітлення, температура повітря та інші, для забезпечення оптимального середовища проживання. Розумний будинок дозволяє насолоджуватися комфортом та високим рівнем безпеки в будь-якому просторі, будь-який час.

Порядок роботи системи

Система управління будинком включає в себе ряд ключових компонентів, що забезпечують контроль за необхідними системами, такими як освітлення, кондиціонування, опалення та інші. Отримання інформації про стан усіх систем може здійснюватися як при прямому перебуванні в будинку, так і дистанційно. Загальна схема системи управління включає такі елементи:

Центральний блок управління / Головний блок управління це центральний модуль, який координує роботу всіх підключених пристроїв та систем. Він відповідає за збір інформації та виконання вказівок користувача.

Датчики - система обладнана різними датчиками, такими як температурні, світлові, димові, пилові тощо. Ці датчики забезпечують систему інформацією про оточуюче середовище.

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		28

Нагрівальні прилади. Сюди входять диммери, лампи, ІЧ-випромінювачі та інші прилади, які можуть бути регульовані для забезпечення потрібних умов у приміщенні.

Інтерфейси управління - це ключові засоби взаємодії з системою, такі як ключові перемикачі, ІЧ-пульти та радіопульти, пульти управління, web/var інтерфейс.

Власна мережа управління - це об'єднуючий елемент, який спільно керує всіма компонентами системи.

Стационарні пристрої включає лампи, кондиціонери, компоненти домашнього кінотеатру та інші пристрої, які можуть бути автоматизовані.

Допоміжні мережі. Система взаємодіє з різними мережами, такими як Ethernet, телефонна мережа, аудіо та відео вихід. Ця інтегрована система надає користувачеві можливість контролювати та моніторити різні аспекти життєдіяльності будинку як зсередини, так і здалеку.

Програмне забезпечення проекту

Центральний процесор в системі "Розумний будинок" виконує ключову функцію управління підтримуваними пристроями за допомогою різних інтерфейсів, таких як Ethernet, RS-232, RS-485, ІЧ-порт, аналогові та цифрові входи/виходи. Він також має багатозадачну операційну систему, інструменти комп'ютерного програмування і, в окремих випадках, веб-сервер.

Датчики розташовані у різних частинах квартири, які з'єднані однією мережею безпосередньо або через проміжний екран. Інтерфейси управління створюють загальне керування системою "Розумний будинок".

Загальний алгоритм роботи системи "Розумний дім" такий. Інформація від датчиків або інтерфейсів передається до центрального блоку управління вздовж межі контролю. Програмне забезпечення центрального процесора обробляє отриману інформацію і генерує команди для програм обслуговування. Команди надходять як зі своїх, так і з допоміжних рубежів. Методи формування команд, а

також формат і зміст інформації, яку отримує система, встановлюються на етапі розробки програмного забезпечення з урахуванням вимог проекту.

2.2 Обґрунтування вибору методів розробки

Для дослідження та програмної реалізації системи управління розумним будинком з підсистемою безпечної передачі даних були обрані наступні заходи та мови програмування:

Система обслуговування «Розумний дім» складатиметься з елементів:

1) електроприлади - це безперервно всі електронні пристрої, управління якими має бути автоматизованим;

2) сенсори - контроль та інформаційний контроль розумного будинку. Тільки вони створюють поле єдності в подібних системах;

3) мікроконтролери - це автономні системи, які об'єднують датчики в групи, до складу яких також входить центральний блок управління - мікроконтролер, який посилає інформацію від джерела до кінцевих вузлів;

4) інтернет – комп'ютер, який створює інтерфейс між користувачем і мережею розумного будинку. Він відповідає за надійність і функціональність;

5) канали передачі даних – логічні та фізичні канали, якими передаються дані з урахуванням вимог (безпека, швидкість тощо);

6) зовнішня служба, яка реалізує поле бази даних для статистики та іншої службової інформації;

7) мобільні пристрої – це пристрої, за допомогою яких користувач забезпечує безпеку розумного будинку через Інтернет;

8) мова програмування - дозволяє створювати програми з набагато більшим комфортом, надаючи користувачеві всі переваги мови високого рівня. Ще раз відзначимо, що архітектура і система команд AVR була створена за допомогою постійного вивчення компілятора мови Cі, і в ньому приховані

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		30

особливості цієї мови. Компіляція вихідних даних, написаних на Сі, є швидкою та створює компактний ефективний код.

9) система безпеки передачі даних (система шифрування) – Base64/AES/RSA.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на магістерську роботу, реалізації підлягає програмне забезпечення, яке призначено для системи управління розумним будинком з підсистемою безпеки передачі даних та захистом від кібератак.

В процесі розробки магістерської роботи необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методикку побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		31

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

КБПЗ - 2023

					БКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		32

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Враховуючи тему роботи, необхідно розробити програмне забезпечення для системи управління в проекті "Розумний Дім" та створити максимально автоматизовану систему управління, встановивши необхідні датчики та активуючи систему захисту від кібератак.

Розумний будинок є інтелектуальною системою автоматизації управління інженерними системами сучасної будівлі. Забезпечення комфорту та безпеки для людини в її домі, спальні чи офісі є основним завданням системи "Розумний дім". Інтелектуальна автоматика контролює всі інженерні системи в будинку, дозволяючи централізовано регулювати комфортні параметри, такі як температура, вологість, освітлення в різних зонах та забезпечує безпеку.

Система "Розумний будинок" повинна надавати механізм централізованого контролю та інтелектуального управління в житлових, офісних або громадських приміщеннях. Встановивши таку систему вдома або на роботі, користувач отримує можливість:

- контролювати необхідні системи (освітлення, кондиціонування, опалення тощо);
- отримати доступ до інформації про стан всіх систем життєзабезпечення будинку (в приміщенні або дистанційно).

Загальна схема системи управління включає такі елементи:

- центральний блок управління / головний блок управління;
- датчики (температури, світла, диму, пилу тощо);
- нагрівальні прилади (диммери, лампи, ГЧ-випромінювачі тощо);
- інтерфейси управління (ключові перемикачі, ГЧ-пульти та радіопульти, пульти управління, web/var інтерфейс);
- власна мережа управління, що об'єднує зазначені елементи;

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		33

- стаціонарна побутова техніка (лампи, кондиționери, компоненти домашнього кінотеатру тощо);
- допоміжні мережі (Ethernet, телефонна мережа, аудіо та відео вихід);
- програмне забезпечення проекту.

Основною функцією центрального процесора є керування пристроями через інтерфейси, такі як Ethernet, RS-232, RS-485, ІЧ-порт, аналогові та цифрові входи/виходи. Крім того, центральний процесор включає багатозадачну операційну систему, засоби програмування і можливий веб-сервер. Датчики розташовані в різних частинах квартири, пов'язані однією мережею безпосередньо або через проміжний екран. Інтерфейси управління створюють загальне управління системою Розумний будинок.

Основна концепція

Система управління представляє собою комплекс програмно-апаратних заходів, спрямованих переважно на ефективне використання ресурсів, зокрема на зменшення витрат можливих джерел, таких як електроенергія і тепло в акумуляторах. Паралельно, вона може включати додаткові функції, такі як контроль присутності. Давайте розглянемо детальніше всі особливості цієї системи.

Враховуючи ці особливості, система управління розроблена з орієнтацією на ефективність та зручність користувача, а також на мінімізацію негативного впливу на навколишнє середовище. Детально розглянемо нижче.

Безпека та захист

Енергозберігаюча система керування освітленням у багатоповерхових будинках, охоплюючи під'їзди, стоянки, відкриті тераси, підвали, горища, спрямована на значний зниження споживаної електроенергії в 10-15 разів. У цій системі використовується система управління освітленням з окремими енергетичними компонентами, яка може копіювати існуючі лінії електропередач.

Важливим етапом в енергозбереженні освітлення є поліпшення технічного обслуговування освітлювальних приладів. Ефективна стратегія енергозбереження

включає централізоване управління освітленням за допомогою спеціально підготовлених графіків включення та виключення світла.

Енергозберігаюче освітлення може бути досягнуте через оптимальне планування будівлі та вбудованих приміщень, а також застосування енергозберігаючих ламп. Автоматичні вимикачі світла з використанням інфрачервоних та електронних датчиків забезпечують найкращу енергетичну безпеку.

Електронні датчики вимірюють рівень освітленості і можуть видавати команди на включення або вимикання освітлення. Датчики присутності визначають, чи перебуває людина в приміщенні, що дозволяє ефективно керувати включенням світла.

Система також використовує різні типи електронних вимикачів, які можуть працювати автономно або в складі автоматизованої системи управління, що називається «розумний будинок». Датчики пилу інтегровані в енергозберігаючі вимикачі, що раціонально керують освітленням враховуючи планування приміщення.

Система управління також включає новий захист електропостачання, який використовує електромагнітне обладнання від провідних європейських компаній. Монтаж системи здійснюється за допомогою телефонного кабелю, що спрощує інтеграцію та експлуатацію.

Зазначена енергозберігаюча система керування освітленням в багатоповерхових будинках впроваджує інноваційні підходи до оптимізації використання електроенергії. Деякі ключові особливості цієї системи включають:

Система управління освітленням. Застосування централізованої системи управління, яка дозволяє ефективно керувати освітленням у всіх приміщеннях будинка. Графіки включення та виключення світла розроблені з урахуванням режимів користувачів та оптимального використання природного світла.

Енергозберігаючі технології. Використання енергозберігаючих ламп, які дозволяють економити електроенергію порівняно з традиційними лампами. Це

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лам		35

може бути досягнуто за рахунок використання LED-ламп, які є не лише ефективними, але й довговічними.

Автоматизовані вимикач. Використання автоматичних вимикачів з інфрачервоними та електронними датчиками. Ці датчики можуть визначати рівень освітленості в приміщенні і вмикають або вимикають світло в залежності від умов.

Датчики присутності. Застосування датчиків присутності, які реагують на рух або інфрачервоне випромінювання людини. Це дозволяє системі управління точно визначати, чи перебуває хто-то в приміщенні, і регулювати освітлення відповідно.

Електромагнітне обладнання. Використання якісного та надійного електромагнітного обладнання від провідних європейських виробників, що забезпечує стабільну роботу системи та її захист від перевантажень та коротких замикань.

Дистанційне управління. Можливість дистанційного управління системою через телефонний кабель, що дозволяє зручно керувати освітленням з віддаленого місця.

Всі ці заходи спрямовані на забезпечення ефективного та енергозберігаючого освітлення в будинку, що сприяє як економії електроенергії, так і забезпеченню комфортних умов для мешканців.

Освітлення

В інтелектуальній системі "Розумний дім" активування освітлення стає простим завдяки одній кнопці або пульта дистанційного керування. З пульта ви легко налаштуєте освітлення, включаючи лампи, підсвічування та світильники, відповідно до ваших вподобань. Якщо вам потрібно пригостити гостей та створити атмосферу, система "Розумний дім" дозволяє змінювати освітлення в кімнаті одним рухом.

Датчики пилу автоматично активують світло при наближенні. Для забезпечення комфорту і затишку в усіх приміщеннях, кожна кімната, передпокій

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		36

та коридор повинні мати ефективне освітлення. Без інтелектуальної системи "Розумний дім" це вимагало б встановлення великої кількості освітлювальних приладів з численними вимикачами.

Ця система вбереже вас від потреби у великій кількості вимикачів, замінюючи їх компактними вимикачами. Тепер з одного стандартного вимикача можна керувати дванадцятьма групами освітлення, плавно регулюючи їх яскравість та вмикаючи/вимикаючи за бажанням. Стінні вимикачі чи панелі легко дозволять вам створити різноманітні сценарії освітлення, які забезпечать вашому будинку атмосферу затишку та комфорту. Наприклад, сценарій "Вечір" включить одну групу світлових приладів з певною інтенсивністю, вимкне іншу групу, закриє жалюзі та перейде систему клімат-контролю в комфортний режим. Вночі світло в під'їздах та коридорах автоматично вмикається на часткову потужність при виявленні пилу, що забезпечує вам зручність у темряві.

Системою освітлення можна також керувати дистанційно з пульта, ноутбука чи мобільного телефону. Повертаючись в будинок уночі, вас зустрічає підсвічування фасаду, ландшафтне освітлення та пішохідні доріжки.

Контроль освітлення є ключовою функцією для вашого будинку. Завдяки інтелектуальному програмуванню можна ефективно витратити електроенергію та регулювати яскравість ламп. Немає потреби шукати вимикач світла у темряві - вимикайте його просто виходячи з кімнати. Інтелектуальна система вимкне світло тільки тоді, коли ви зачините двері, та ввімкне м'яке освітлення, якщо ви виходите вночі, щоб не травмувати очі різким світлом. Зранку система визначить, яке освітлення необхідне, враховуючи погоду на вулиці.

Автоматизована система керування освітленням може бути налаштована для розпізнавання присутності.

Система клімат-контроль

Ця система клімат-контролю базується на вбудованих алгоритмах, які дозволяють підтримувати параметри повітряного середовища та кліматичні зони в приміщеннях із мінімальними витратами енергії. Система надає можливість

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		37

здійснювати операції з затримкою, такі як обігрів або охолодження. Наприклад, вночі може відключатися кондиціонування та опалення, за винятком теплої підлоги. Це сприяє зниженню температури в нежилых приміщеннях і спальнях, створюючи комфортні умови для сну та економляючи електроенергію.

Система також дозволяє мінімізувати роботу техніки та обладнання під час відсутності людей, використовуючи режими "щоденна відсутність" та "відпустка". Наприклад, можна налаштувати, щоб у вихідні години подача тепла в приміщення включалася або відключалася. Це особливо актуально для замкових котеджів з автономним опаленням.

Система дозволяє індивідуально налаштувати температуру, вологість та кількість свіжого повітря в кожній кімнаті, створюючи індивідуальну кліматичну систему для кожного члена сім'ї. Вона також забезпечує фінансову економію та вирішує проблеми безпеки, дозволяючи дистанційно вмикати котел опалення або переводити його в економічний режим. Система дозволяє створити комфортний клімат в будинку, активуючи її через телефон або Інтернет.

З використанням датчиків, вимикачів і панелей, система контролює параметри мікроклімату в будинку, забезпечуючи стабільні умови відповідно до режимів роботи. Така система дозволяє ефективно управляти температурою, вологістю та іншими параметрами в будинку, забезпечуючи комфортне життя.

Контроль проникнення

Постановка і зняття будинку з охорони виконується за допомогою кодової панелі, розміщеної у тамбурі. При відкритті входних дверей у людини є 30 секунд на введення правильного коду на панелі. Якщо ж код не буде введений розумний будинок включить сирени і відправить SMS-повідомлення на декілька телефонних номерів.

Датчики руху, розташовані в кухні, в спальні, в вітальні дозволять виявити проникнення через вікна.

Схема застосування датчиків представлена на рисунку 3.1



Рисунок 3.1 - Схема застосування датчиків

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

Контроль протікання води

Виявлення протікання труб водопостачання може стати неприємною ситуацією, яка не тільки зачіпає власні майнові інтереси, але й може впливати на власність сусідів. Розумний будинок може ефективно виявляти та уникати витоків води. Контрольовані зони включають в себе санвузли та кухню, тобто ті приміщення, де прокладені труби водопостачання.

Датчики спеціально реагують на протікання труб або перелив води через краї раковин. У випадку виявлення протікання, розумний будинок автоматично перекриває доступ води в квартиру та надсилає повідомлення за вказаними номерами телефонів.

Схема використання датчиків протікання води представлена на рисунку 3.2



Рисунок 3.2 – Схема використання датчиків протікання води

Отже, в ході роботи над даним підрозділом було обрано систему управління над розумним домом, яка являє собою сукупність підсистем управління окремими параметрами та групами датчиків/контролерів.

3.2 Розробка структурної схеми

Структурна схема системи - це сукупність об'єктів та їхніх взаємозв'язків. Основною метою структурної схеми є візуальне відображення компонентів розглядуваної системи, її основних блоків, вузлів і зв'язків між ними.

Структурна схема розробленої системи представлена на рисунку 3.3.



Рисунок 3.3 – Структурна схема системи «Розумний дім»

Пристрій, побудований за наведеною конструктивною схемою, живиться від електричної мережі (220 В, 50 Гц). Змінна напруга на вході пристрою перетворюється в стабілізовану постійну напругу 3,5 В за допомогою підсилювача-перетворювача. Цей підсилювач живить мікроконтролер DSPIC33FJ16 і модуль Bluetooth Lov Energi RN4020 від Microchip. RN4020 відповідає за бездротовий зв'язок між пристроєм і комп'ютером, а також передачу даних в обох напрямках.

Мікроконтролер DSPIC33FJ16 встановлює зв'язок з RN4020 через канал UART, генерує допоміжні сигнали, вимірює аналогові значення необхідних параметрів електричного пристрою, таких як струм, напруга і температура, і перетворює їх в цифровий формат. Крім того, мікроконтролер створює сигнали для виконавчого елемента - оптодіода MOC3023, який комутує електричний пристрій. В електричному пристрої вбудовані датчики напруги і струму, які у формі аналогового сигналу (0-3,5 В) подаються на входи АЦП мікроконтролера.

3.3 Розробка функціональної схеми

На рисунку 3.4 представлена функціональна схема складної системи. З рисунка видно, що ця система складається з наступних блоків:

- сервер розумного будинку;
- веб-сервер;
- контролер із супутнім обладнанням;
- Bluetooth модуль RN4020;
- модуль безпеки та шифрування;
- Веб-інтерфейс;
- мобільний додаток;
- виконавчі механізми;
- розумна розетка.

Ці блоки взаємодіють між собою, утворюючи комплексну систему управління розумним будинком.

Bluetooth Lov Energi Server

Здійснює підключення та отримання даних від мережевих пристроїв SmartSwitch за допомогою протоколу Bluetooth Low Energy. У режимі реального часу сервер передає отримані дані на веб-сервер і очікує від нього відповідь, яка представляє собою результати роботи алгоритму оптимізації. На основі цих результатів сервер генерує керуючий сигнал, який подальше передається на відповідний SmartSwitch для виконання відповідних дій.

Веб-сервер

Веб-сервер зберігає дані, отримані від сервера Bluetooth LE. На їх основі, використовуючи алгоритм оптимізації, він приймає рішення про підключення/відключення побутової техніки до мережі для обмеження навантаження на електромережу або уникнення аварійних ситуацій. Крім того, веб-сервер надає API для ручного керування системою через графічний інтерфейс користувача.

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		42

Модулі BLE серії RN4020

Модулі BLE серії RN4020 виготовлені компанією Microchip Technology Inc. Вони включають вбудований стек BLEnergy, набір ASCII-команд і можливість створення скриптів. Ці функції полегшують розробку готових пристроїв BLE, не вимагаючи низькорівневого програмування або встановлення спеціальних середовищ розробки, бібліотек і компіляторів. Розробник може використовувати звичайні інструменти, до яких він звик. Усі налаштування модуля зберігаються в незалежній пам'яті, включаючи стандартні профілі малого споживання SIG і приватний профіль MLDP для потокових даних.

Розумна розетка

Розумна розетка, також відома як розумна вилка, є електричною розеткою, яку можна вмикати та вимикати автоматично або за допомогою команди зі смартфона. Вона дозволяє автоматизувати будь-який пристрій, підключений до неї.

Функціональна схема системи «Розумний дім» представлена на рисунку 3.4.

					БКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		43

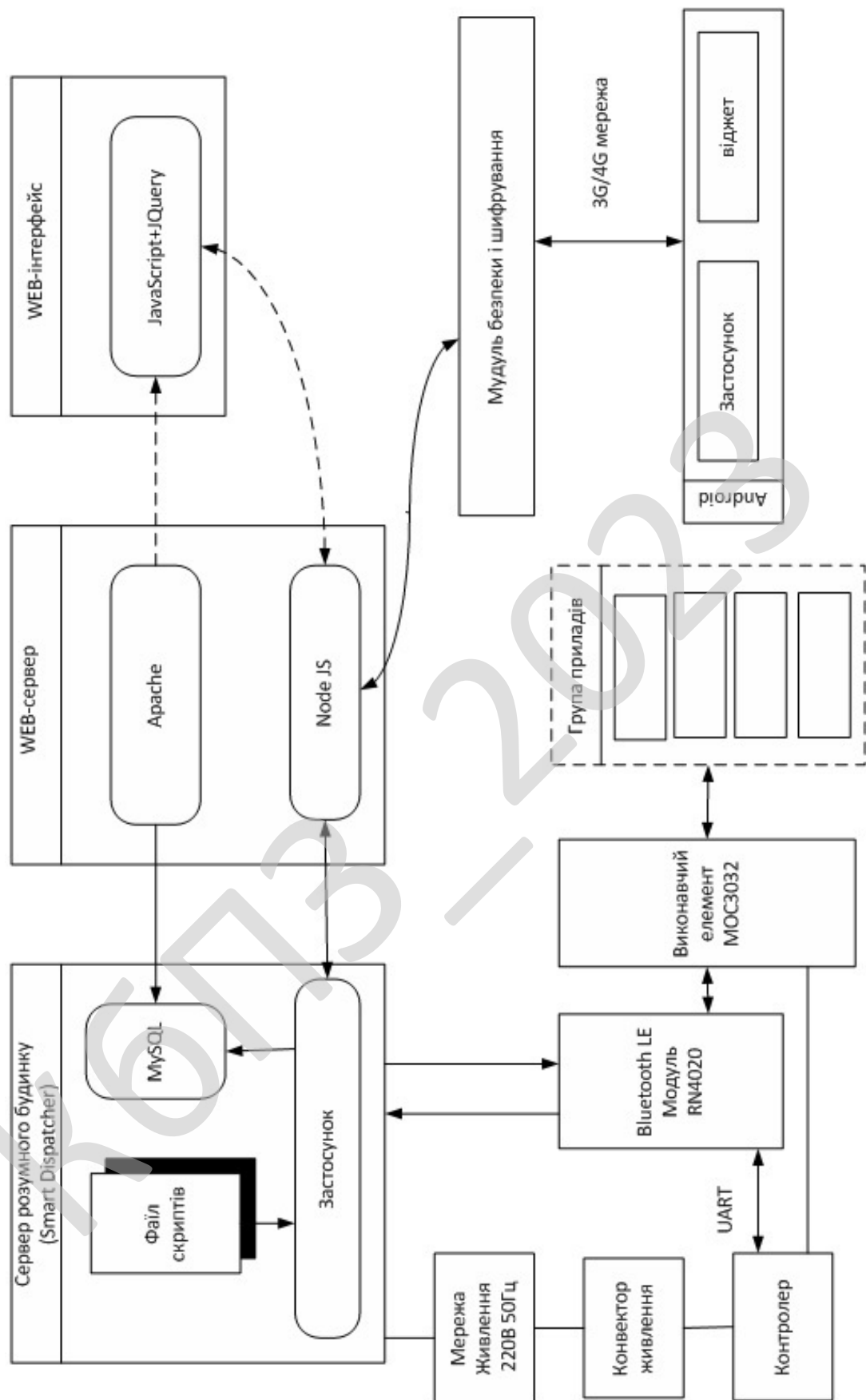


Рисунок 3.4 – Функціональна схема системи «Розумний дім»

Вим.	Арк.	№ докум.	Підпис	Лат

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи, яка подана на рисунку 3.5. відображає взаємодію компонентів у системі. Використана модель проектування, яка представляє "потоки" даних в інформаційній системі. Для візуалізації обробки даних та структури проекту використана діаграма взаємодії процесу.

У цьому контексті зазвичай розробники спочатку створюють діаграму взаємодії процесу на рівні контексту, що показує, як система взаємодіє в цілому. Подальша деталізація включає в себе додавання деталей до процесів та потоків даних, щоб продемонструвати розроблену систему.

Застосовуючи існуючі методи, була розроблена система управління розумним будинком з підсистемою забезпечення безпеки передачі сигналів управління від смартфона до мікроконтролера. Це забезпечує безпеку передачі даних під час дистанційного керування системою за допомогою Інтернет-технологій.

Після огляду системи, структурно-функціональної діаграми та діаграми взаємодії процесів ми переходимо до огляду блок-схеми хмарної програми та підпрограм, які були створені для реалізації системи.

Діаграма процесів системи «Розумний дім» представлена на рисунку 3.5.

					БКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		45

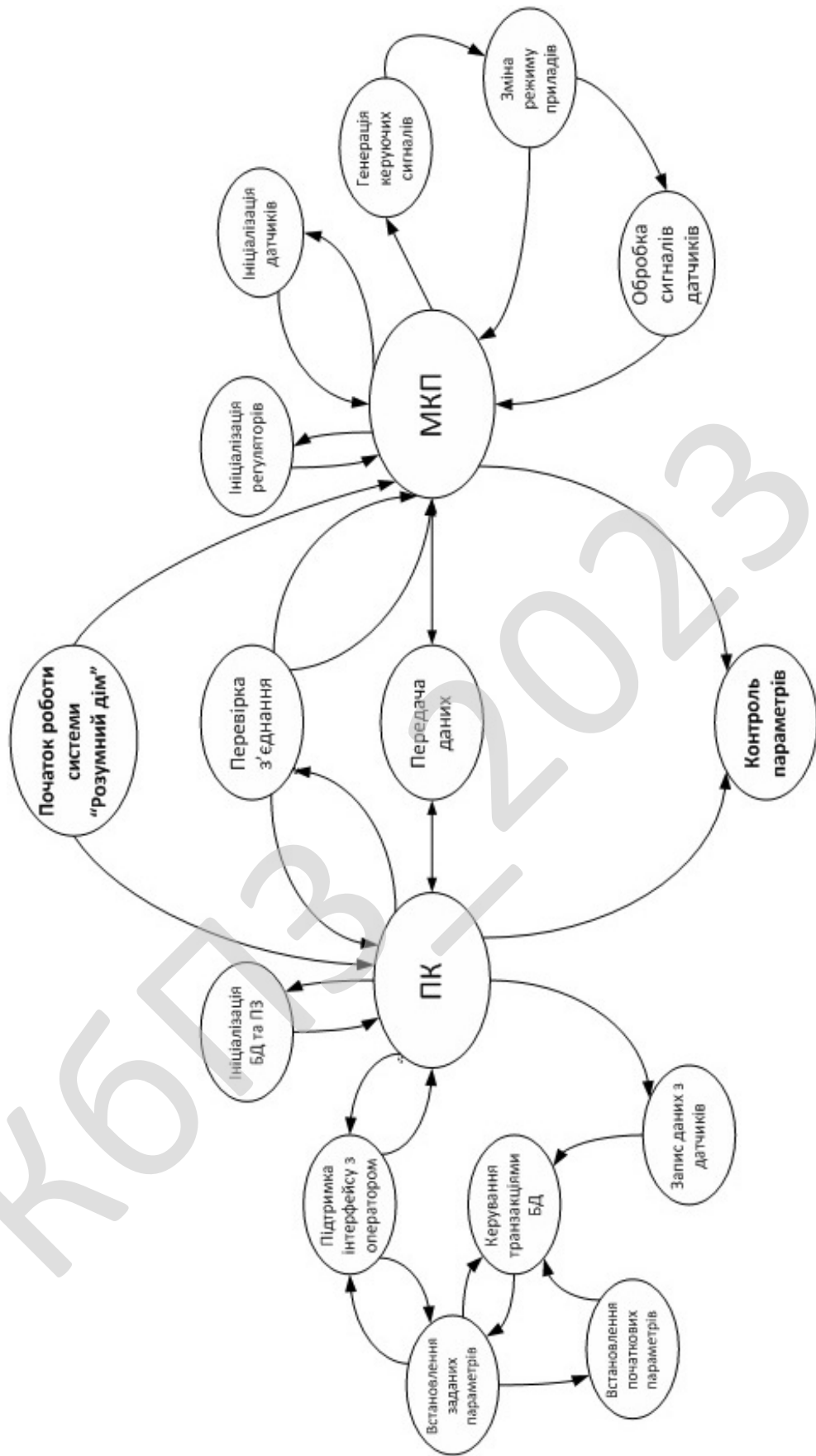


Рисунок 3.5 – Діаграма процесів системи «Розумний дім»

Вим.	Арк.	№ докум.	Підпис	Лат
------	------	----------	--------	-----

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ І ПРОГРАМНИХ РІШЕНЬ

Ця програма призначена для моніторингу та контролю потоку електроенергії, що споживається (або виробляється) електричними пристроями.

Вимоги до розробки:

- прилад живиться від мережі (220В, 50Гц);
- прилад повинен мати високопродуктивний модуль Bluetooth Low Energy для забезпечення підключення до мережі системи;
- пристрій повинен мати мікроконтролер, що забезпечує роботу модуля Bluetooth LE, вимірювання аналогових значень технологічних параметрів (тиск, напруга і температура) і їх перетворення в 10-ти інвертований двійковий код d, формування сигналів для повороту вмикати або вимикати електроприлади;
- прилад повинен мати зовнішнє оптичне з'єднання для ввімкнення або вимкнення живлення електроприладів.

Прилад, побудований за наведеною конструктивною схемою, як уже зазначалося, живиться від електроенергії (220В, 50Гц). Змінна вхідна напруга приладу перетворюється за допомогою коду будівлі в стабілізовану постійну напругу 3,5 В, яка живить мікроконтролер DSPIC33FJ16 і модуль Bluetooth Low Energy RN4020 (Microchip). RN4020 забезпечує бездротовий зв'язок між принтером і комп'ютером і передачу даних в обох напрямках.

Мікроконтролер DSPIC33FJ16 зв'язується з RN4020 за допомогою каналу UART, генерує допоміжні сигнали та імітує аналогові значення необхідних параметрів електронного пристрою, таких як напруга, напруга та температура, і їх перетворення в цифровий формат. Крім того, мікроконтролер формує сигнали для виконавчого елемента - оптрона МС3023, який комутує електронний пристрій. В електричній пристрій вбудовані датчики напруги і струму, які у

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		47

вигляді аналогового сигналу (0-3,5 В) подаються на входи мікроконтролера АЦП.

Bluetooth LE модуль Microchip RN4020

Компанія Microchip є світовим лідером на ринку мікрокомп'ютерів. Microchip пропонує кілька продуктів Bluetooth LE - від окремих мікросхем до модулів.

Однією з ключових переваг модулів є те, що вони зазвичай уже сертифіковані виробником і мають сертифікати FCC або CE/ETSI, а також пройшли тестування Bluetooth SIG.

Ще однією перевагою модулів є те, що вони мають гарантовано працюючу радіочастотну складову. Модулі, як правило, мають інтегрований шлях і налаштовуються декількома командами, що значно скорочує час спроб друку.

Одним з найпопулярніших модулів є RN4020, принципова схема якого показана на рисунку 4.1.

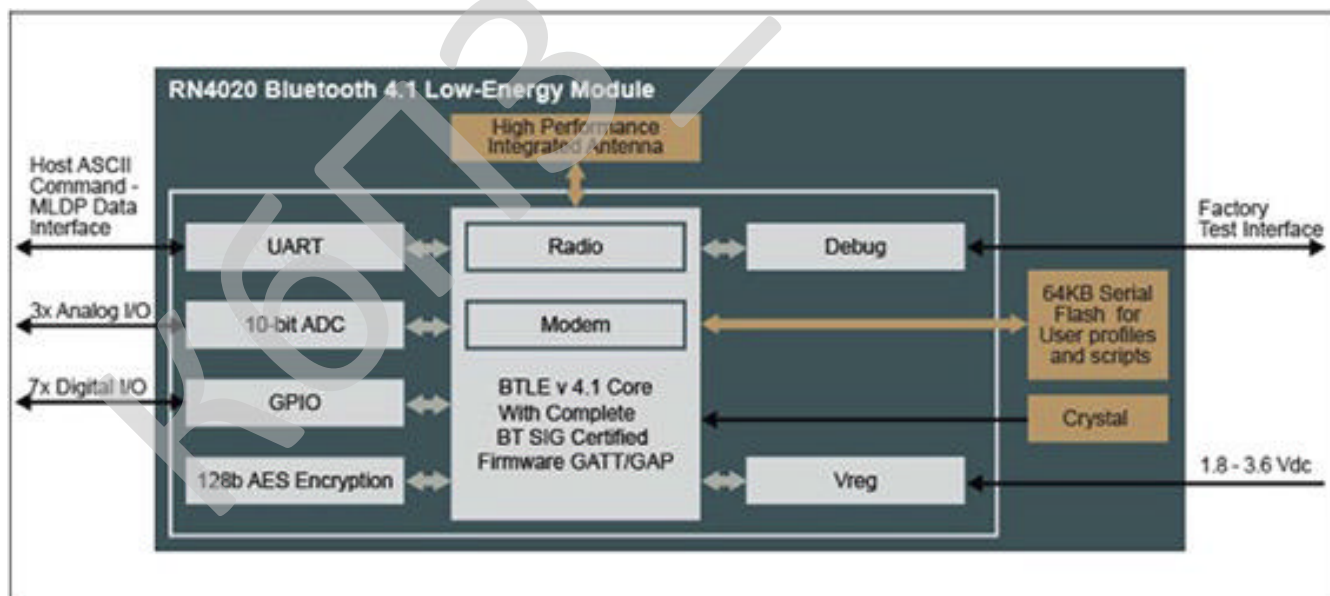


Рисунок 4.1 - Структурна схема модуля RN4020



Рисунок 4.2 – Модуль Bluetooth LE RN4020

Мікроконтролер Microchip dsPIC33FJ16

В якості контролера використовується спеціалізований мікроконтролер компанії Microchip DSPIC33FJ16MC102, що дозволяє реалізувати весь описаний функціонал в однопроцесорній схемі із максимальним використанням переваг сучасних сигнальних процесорів. Структурна схема DSPIC33FJ16MC102 наведена на рисунку 4.3.

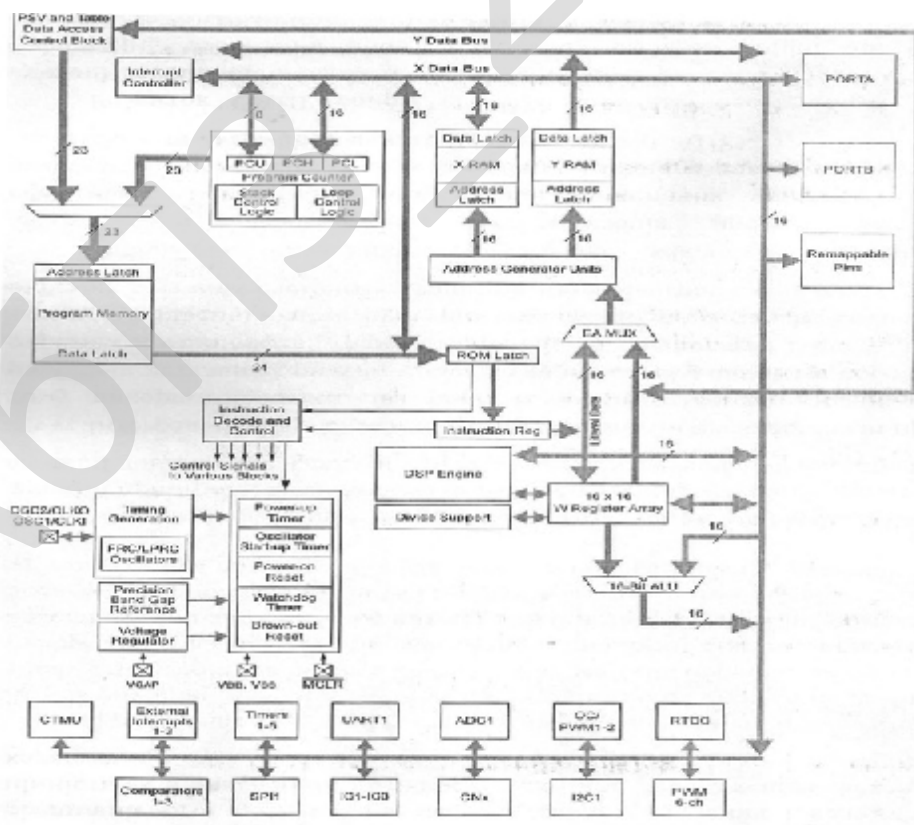


Рисунок 4.3 – Структурна схема мікроконтролера DSPIC33FJ16MC102

Вим.	Арк.	№ докум.	Підпис	Лам
------	------	----------	--------	-----

Схему входів/виходів мікроконтролера з позначеннями входів/виходів показано на рисунку 4.4

28-Pin SPDIP/SOIC/SSOP

■ = Pins are up to 5V tolerant

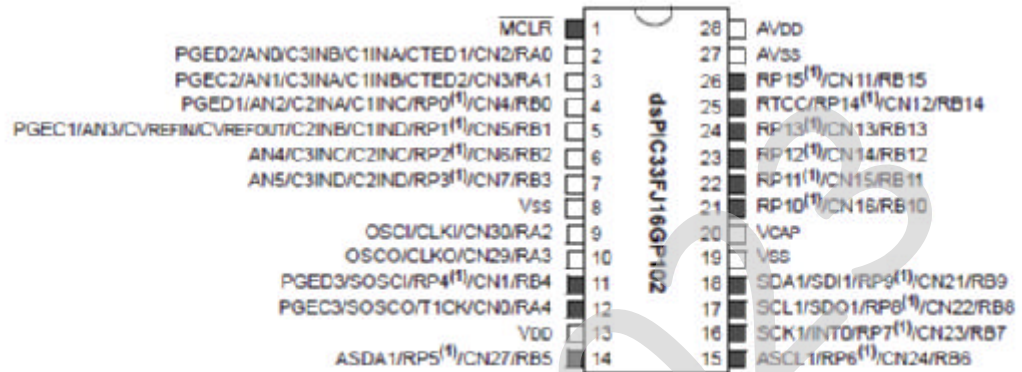


Рисунок 4.4 – Загальна структура мікроконтролера DSPIC33FJ16MC102

Проектування схеми

На рисунку 4.5 наведено електричну схему з'єднання RN4020 та мікроконтролера DSPIC33FJ16MC102, а також підключення до нього конектора.

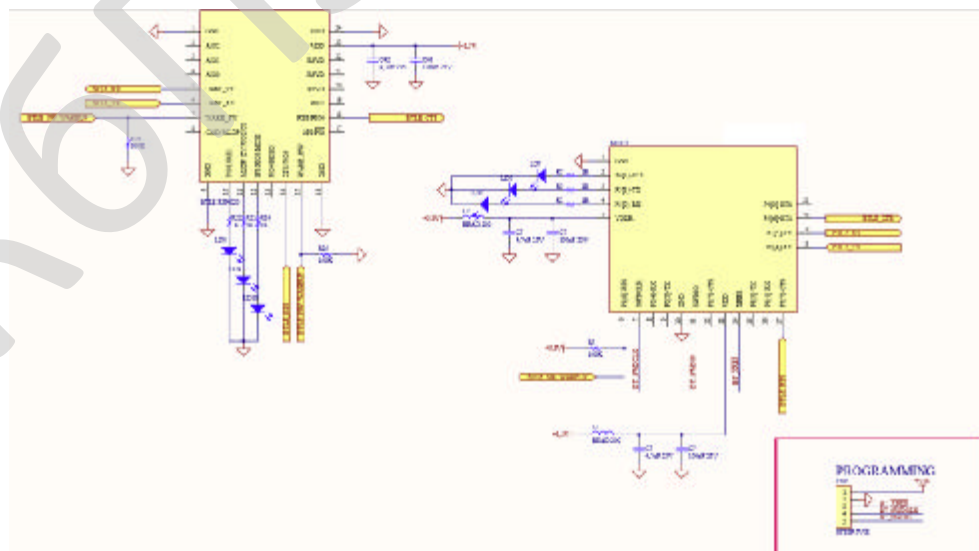


Рисунок 4.5 – Схема з'єднання RN4020 та мікроконтролера

Програмування мікроконтролера

Програмне забезпечення для мікроконтролера виконано у середовищі MPLAB X IDE із апаратними засобами відладки та програмування PICkit3.

У якості апаратного засобу відладки та програмування використано PICkit рисунку 4.6.



Рисунок 4.6 – відладчик-програматор PICkit

PICkit 3 - це перший у своєму роді простий у використанні налагоджувач програм, який підтримується за допомогою ПК із встановленим програмним забезпеченням MPLAB X IDE. PICkit 3 є невід'ємною частиною набору апаратних і програмних інженерних засобів для мікроконтролерів Microchip PIC (MCU) і контролерів цифрових сигналів (DSP) dsPIC, які базуються на внутрішній системі ICSP через двонаправлений послідовний інтерфейс.

MPLAB X IDE — це програма, яка працює в пізніших операційних системах на комп'ютерах (Windows, Mac OS, Linux) для тестування додатків для мікрокомп'ютерів Microchip і цифрових сигнальних процесорів. Його називають інтегрованим середовищем розробки (IDE), оскільки воно забезпечує зручний інтерфейс для розробки коду для мікроконтекстних програм у IDE Oracle з відкритим кодом NetBeans.

Весь програмний проект розподіляється на декілька файлів та папки за їх функціональним призначенням, що показано на рисунку 4.7.

					БКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		51

Файл "bluetooth.c" тут оголошено функції, що відповідають за передачу та прийом даних. Далі наведено функцію що ініціалізує модуль набором команд, що запускають у ньому необхідний сервіс та активує його характеристики:

```
char BT_SetupModule()  
{  
    BT_SendCommand("sf,1\r"); //reset module settings  
    BT_SendCommand("ss,30000000\r");  
    BT_SendCommand("sr,32000000\r");  
    BT_SendCommand("sn, Ener1 \r"); //set name  
    BT_SendCommand("r,1\r"); //reload module to enable changes  
    return TRUE  
}
```

Далі наведено функцію, що передає повідомлення через UART інтерфейс на RN4020:

```
char BT_SendPacket(struct BT_Message *MessageOut)  
{  
    WriteTxBuffer(BT_SOF_1);  
    WriteTxBuffer(BT_SOF_2);  
    WriteTxBuffer((int)MessageOut->Command);  
    WriteTxBuffer((int)MessageOut->Data);  
    WriteTxBuffer('\r'); //Load carriage return  
    WriteTxBuffer('\n'); //Load line feed  
    UART_TxStart(); //Start the transmission  
    return TRUE;  
}
```

Далі наведено функцію що приймає повідомлення від RN4020:

```
char BT_ReceivePacket(struct BT_Message *Message)  
{  
    char messageChar;  
    if (IsNewRxData())  
    {  
        messageChar = ReadRxBuffer();  
        if(messageChar == '\n')  
            return TRUE;  
    }  
    return FALSE;  
}
```

Файл “main.c”

Виклик основної функції main(). Наведено фрагмент нескінченного циклу в якому відбувається підключення до пристрою з подальшою пересилкою даних та перевіркою наявності вхідних повідомлень для комутації навантаження:

```
if(blueetoothState == Connected)
{
sendADCvalues();
gotPacket = BT_ReceivePacket(&IncomingMessage);
if(gotPacket == TRUE)
{
if(IncomingMessage.Command == SWITCH)
{
TRIAC_1 = (TRIAC_1 + 1) % 2;
}
}
}
```

Таким чином, ця програма забезпечує наступну взаємодію зі смартфоном: коли він підключається через Bluetooth до смартфона, починається передача даних (струм, напруга, температура). У свою чергу комутатор може відправляти на плату команду SWITCH, після чого мікрокомп'ютер перемикає все електрообладнання через оптрон.

Програмна реалізація системи

Для реалізації системи, яка об'єднує описані нижні точки в єдину межу, розроблена сучасна архітектура. Він включає наступні кроки:

- Чіп Bluetooth Lov Energi, який встановлює з'єднання, отримує дані від SmartSwitch типів меж за протоколом Bluetooth Lov Energi. У штатному режимі він передає отримані дані на веб-сервер і очікує від нього відповідь у вигляді результату алгоритму оптимізації, на підставі якого передає сигнал обмеження на відповідний SmartSwitch;

- Web-плагін. Зберігає дані, отримані від пристроїв Bluetooth LE, на своїй сторінці за допомогою алгоритму оптимізації отримує інструкції щодо підключення/відключення побутової техніки до мережі з метою обмеження

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		54

навантаження на електричне коло або уникнення аварійних ситуацій. Надає API для легкого керування файловою системою з графічного інтерфейсу браузера.

На рисунку 4.8 показано початковий контекст діаграми DFD системи.

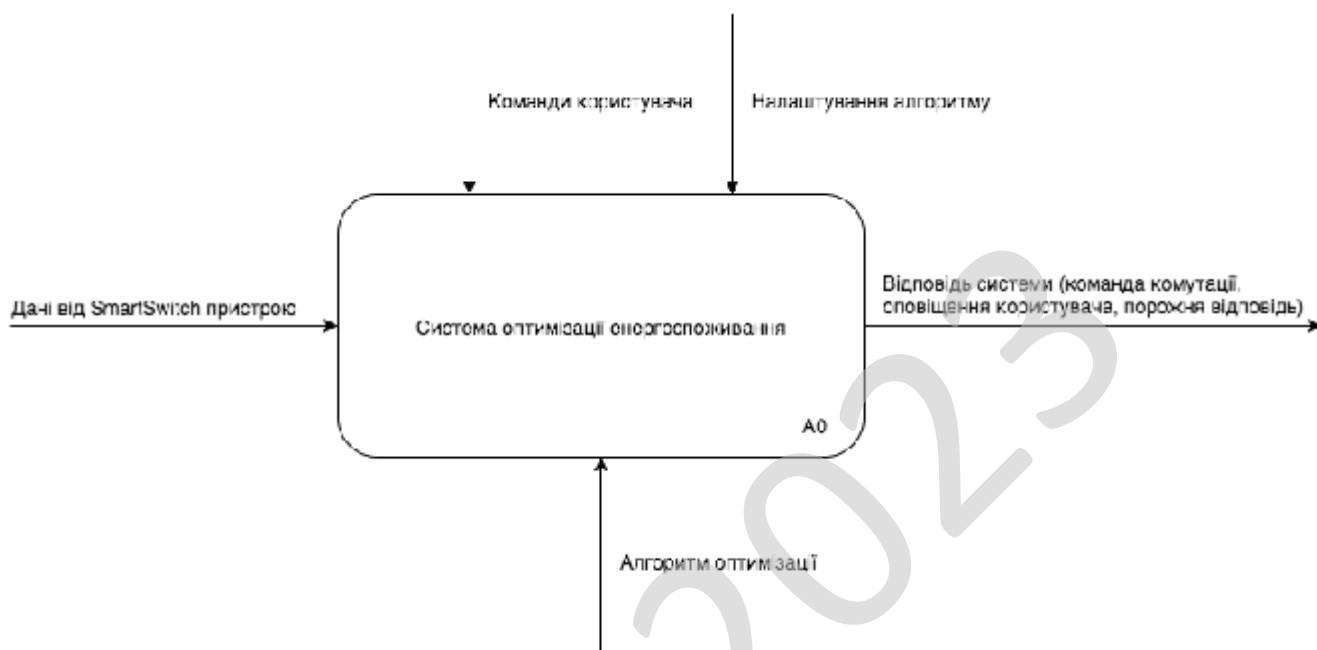


Рисунок 4.8 – Початкова контекстна DFD діаграма

На рисунку 4.9 блок A0 детально описано відповідно до архітектури системи, яка складається з двох сокетів, які взаємодіють за допомогою протоколу WebSocket, і інтерфейсу копіювання, визначеного для моніторингу та керування

СИСТЕМОЮ.

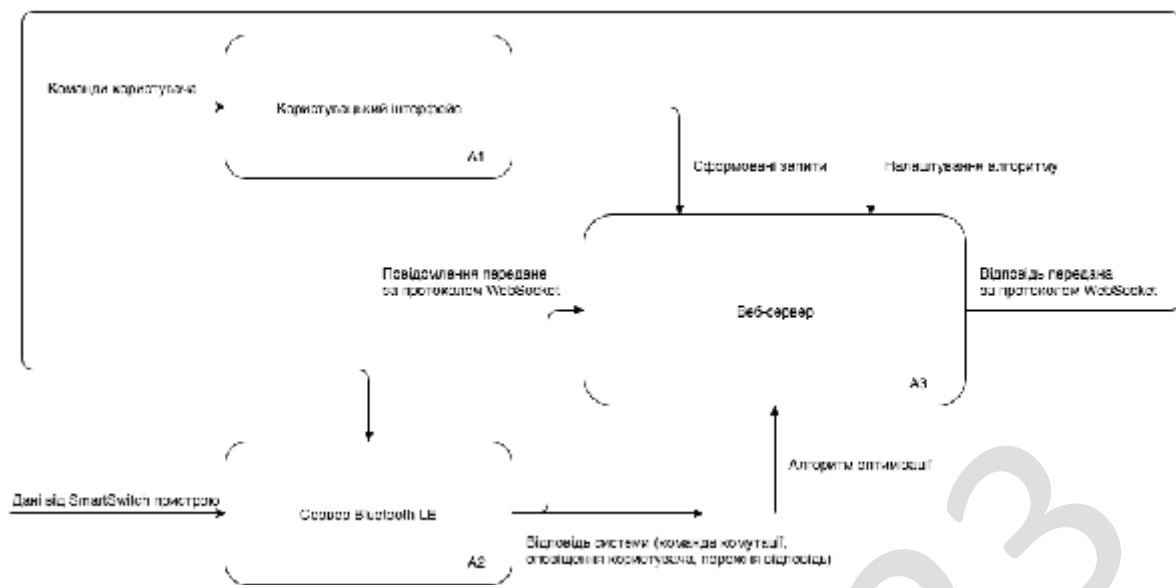


Рисунок 4.9 – Деталізована контекстна DFD діаграма

Для програмної реалізації системи обрано мову програмування Javascript та програмну платформу NodeJS.

В якості бази даних було обрано PostgreSQL, а Redis, нереляційну базу даних ключів і значень, використовували для кешування даних і налаштувань системи.

Протокол WebSocket вибрано для обміну даними в режимі реального часу між веб-підтримкою та Bluetooth LE. Програмний код системи наведено у вкладенні.

У ньому використано такі бібліотеки:

- Noble – багатоплатформна бібліотека, написана на Javascript для роботи з Bluetooth Low Energy в Інтернеті (MacOS, Linux, Windows);
- ExpressJS - програмне забезпечення для створення веб-додатків та API для Node.js Мінімалістичний веб-інтерфейс з можливістю підключення великої кількості плагінів;
- SocketIO – реалізація протоколу WebSocket на мові програмування Javascript для обміну даними між процесами в реальному часі. Як WebSocket

може обробляти запити JSONP або AJAX залежно від можливостей клієнта та мережі;

- pg-promise – Javascript API для роботи з СУБД PostgreSQL;
- redis – Javascript API для роботи з базою Redis, яка не потребує запиту;
- VueJS – мінімальний фреймворк для створення копіювання інтерфейсу.

Сервер Bluetooth LE. Бібліотека Noble

Бібліотека Noble береться як вихідний код, який забезпечує зручний API коду для всіх необхідних типів взаємодії за допомогою протоколу Bluetooth Low Energy у керованій подіями парадигмі, специфічній для мови Javascript:

1) Сканування активних пристроїв мережі:

```
noble.on('stateChange', function(state) {  
  if (state === 'poweredOn') {  
    noble.startScanning();  
  } else {  
    noble.stopScanning();  
  }  
});
```

2) Колбек (callback) для обробки події виявлення нового пристрою внаслідок сканування. В середині тіла даного колбеку відбувається уся робота з самим пристроєм (підключення, передача даних):

```
noble.on('discover', function(peripheral) {  
  console.log('peripheral: ', peripheral.advertisement.localName);  
});
```

3) Підключення до виявленого пристрою. Ідентифікаторами за якими розрізняють різні пристрої є: uuid – унікальний ідентифікатор та MAC-адреса Bluetooth адаптера:

```
peripheral.connect(function(error) {  
  console.log('connected to peripheral: ', peripheral.uuid);  
});
```

4) Відключення:

```
peripheral.disconnect(function(error) {  
  console.log('disconnected from peripheral: ', peripheral.uuid);  
});
```

5) Отримання усіх сервісів пристрою. Сервіси є своєрідними додатками Bluetooth LE. Кожен сервіс має своє власне призначення. Кількість сервісів одного пристрою є необмеженою:

```
peripheral.discoverServices(null, function(error, services) {  
  console.log('discovered the following services:', services);  
});
```

6) Отримання усіх характеристик певного сервісу. Характеристики є інтерфейсами взаємодії для інших пристроїв з даним сервісом.

Поділяються за призначенням на read, write та indicate характеристики:

```
service.discoverCharacteristics(null, function(error, characteristics) {  
  console.log('discovered characteristics:', characteristics);  
});
```

7) Зчитування даних характеристики пристрою (для характеристик типу read):

```
characteristic.read(function(error, data) {  
  console.log(data.toString('utf8'));  
});
```

8) Підписка на нові дані від характеристики (для характеристик типу indicate):

```
characteristic.on('data', function(data, isNotification) {  
  console.log(data.toString('utf8'));  
});
```

9) Надсилання даних до характеристики пристрою (для характеристик типу

write):

```
characteristic.write(new Buffer([0x01]), true, function(error)  
{  
  console.log('sent');  
});
```

Слід зазначити, що ця бібліотека накладає певні обмеження на кількість одночасних підключень до мережі в залежності від швидкості адаптера Bluetooth Low Energy, який може бути вбудованим або зовнішнім. Зазвичай це близько 10 підключених програм одночасно. Однак бібліотека надає чудові можливості для

стан кожного побутового приладу і всієї системи в цілому. Зв'язки сутностей діаграми бази даних показано на малюнку 4.10.

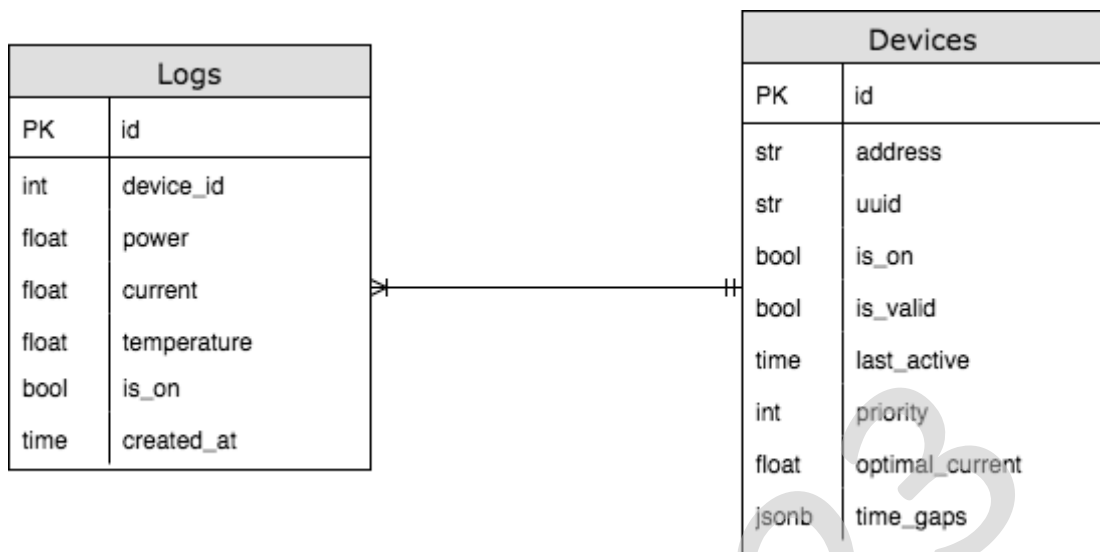


Рисунок 4.10 – ER діаграма бази даних

Деталі реалізації алгоритму оптимізації системи наведені нижче.

Системні параметри (Settings) представляють собою кожну пару «ключ-значення», яка характеризує певні важливі для системи показники. Конкуренція в Redis за збільшення швидкості доступу. Список використовуваних системних налаштувань:

- AUTO_MODE = <bool> Встановлює режим роботи системи. Якщо встановлено значення false, система працює лише в режимі навчання з інтерфейсу запису, тобто наступні дії не виконуються. При значенні true система працює в автоматичному режимі, тобто алгоритм приймає необхідність перемикання виходів;

- ROVER = <float> Параметри алгоритму. Встановлює максимально допустиме загальне навантаження системи у ватах. При перевищенні цього значення пристрій відключається від електричних перешкод. Далі з певною періодичністю робляться спроби підключити пристрій;

- TEMPERATURE = <float> Параметри алгоритму. Встановлює максимально допустиме значення температури в одиницях для цілі. Коли він збільшується, відповідний пристрій відокремлюється від електричного бар'єру.

Крім того, відповідний екземпляр позначається як `is_valid = false`, що забороняє пов'язувати його з дозволом запису файлера, чи може він продовжувати працювати.

На рисунку 4.11 показано інтерфейс редагування для додавання, редагування та видалення параметрів.

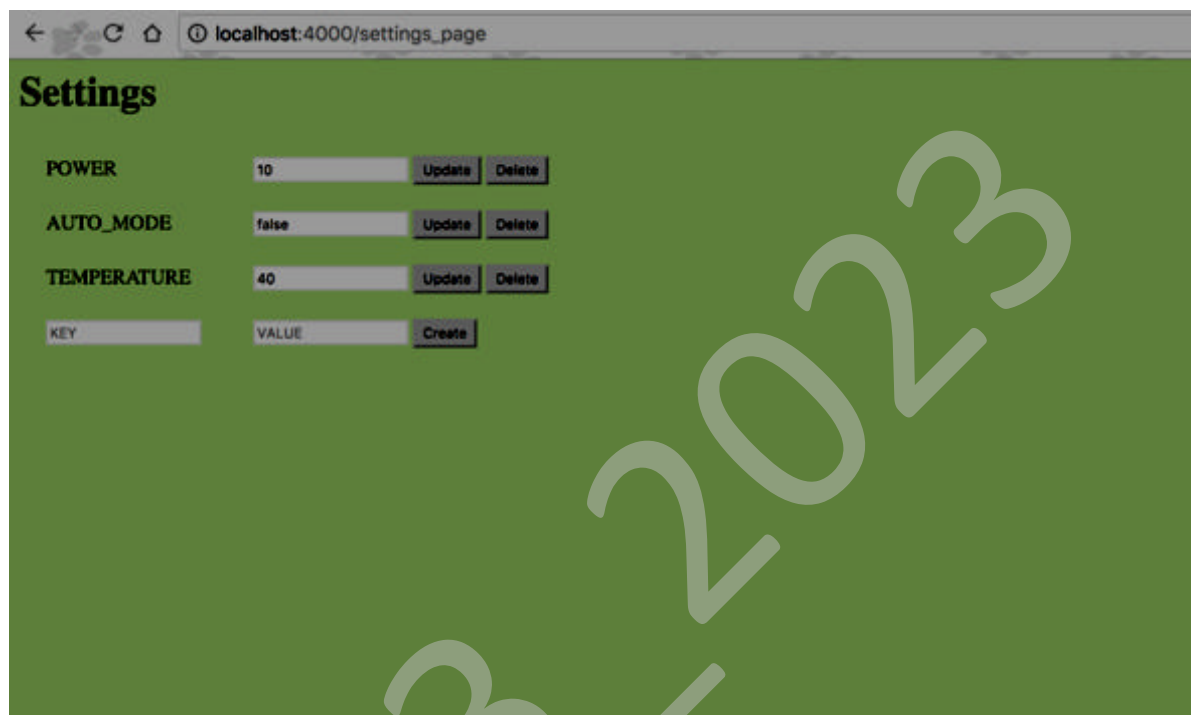


Рисунок 4.11 – Користувацький інтерфейс для управління налаштуваннями

Для реалізації веб-інтерфейсу веб-сервера було обрано фреймворк VueJS, який забезпечує можливість зручного шаблонування HTML-анотацій, схожих на Angular, двостороннього обміну даними та їх оновлення.

Однак обмін даними з веб-сервером здійснюється за допомогою запитів AJAX у форматі JSON.

Для цього було протестовано сумісний AJAX API бібліотеки jQuery. Завдяки цьому не потрібно оновлювати сайт, щоб отримати поточні дані з веб-

сервера - вони автоматично оновлюються з інтервалом в 1 секунду, що дозволяє контролювати систему в реальному часі.

На рисунку 4.12 показано вигляд домашньої сторінки веб-сервера, яка містить інформацію про систему.

ID	Address	UUID	Current Power	Last Active	Priority	Time gaps	Optimal current	Is valid	Is on	Switch
1	00:1ac:0:1b:09:be	97ce5aa29a046bb3b6e3be52137991	0.0000	2017-05-22T09:22:56.832Z	1	[["08:30:00", "22:00:00"]]	0.00	true	false	Switch
2	00:1ac:0:45:ef:a0	9ee98577583e45e781362b08b11623e1	0.0000	2017-05-22T09:22:58.582Z	2	[["08:30:00", "20:15:00"]]	0.00	true	false	Switch

Рисунок 4.12 – Домашня сторінка додатка

Підсистема безпеки

Під загрозою інформаційній безпеці розуміється подія або дія, яка може спричинити зміну у функціонуванні системи, пов'язану з порушенням безпеки інформації, що в ній обробляється.

Вразливість інформації – можливість виникнення такого стану, при якому створюються умови для реалізації загроз інформаційній безпеці.

Атака на інформаційну систему – це дії зловмисника, які полягають у виявленні та використанні тієї чи іншої вразливості. Іншими словами, атака на КС – це реалізація в ньому загрози інформаційній безпеці.

Проблеми, які виникають з безпекою передачі інформації при роботі в комп'ютерних мережах, можна розділити на три основні види [17]:

- перехоплення інформації - зберігається цілісність інформації, але порушується її конфіденційність;

- модифікація інформації - вихідне повідомлення змінюється або повністю замінюється іншим і надсилається адресату;

- зміна авторства інформації. Ця проблема може мати серйозні наслідки. Наприклад, хтось може надіслати лист від чужого імені (такий вид обману називається спуфінгом), або веб-сервер може видати себе за інтернет-магазин, приймати замовлення, номери кредитних карток, але не відправляти будь-які товари.

Специфіка комп'ютерних мереж, з точки зору їх уразливості, здебільшого полягає в наявності інтенсивної інформаційної взаємодії між територіально рознесеними та різнорідними (різного типу) елементами.

Уразливими є буквально всі основні структурні та функціональні елементи КС: робочі станції, сервери (хост-машини), міжмережні мости (шлюзи, комутаційні центри), канали зв'язку тощо.

Відома велика кількість різноманітних загроз інформаційній безпеці різного походження. У літературі існує багато різних класифікацій, де в якості критеріїв класифікації використовуються види небезпеки, ступінь злого умислу, джерела загроз тощо. Одну з найпростіших класифікацій показано на рисунку 4.13.

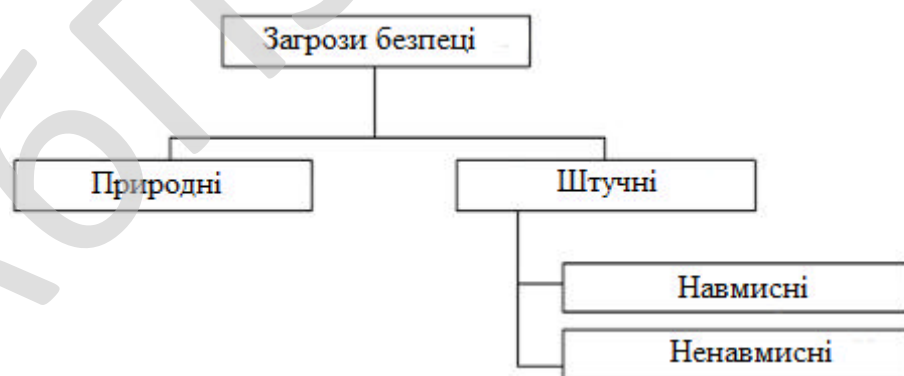


Рисунок 4.13 - Загальна класифікація загроз безпеки.

Природні загрози — це загрози, спричинені впливом на інформаційну систему та її елементи об'єктивних фізичних процесів або незалежних від людини стихійних природних явищ.

Штучні загрози – це загрози інформаційній системі, спричинені діяльністю людини. Серед них, виходячи з мотивації дій, можна виділити:

- ненавмисні (ненавмисні, випадкові) загрози, викликані помилками в проектуванні інформаційної системи та її елементів, помилками в програмному забезпеченні, помилками в поведінці персоналу тощо;

- навмисні (навмисні) погрози, пов'язані з корисливими устремліннями людей (злочинців).

На рисунку 4.14 зображено види загроз безпеки інформації в КМ.

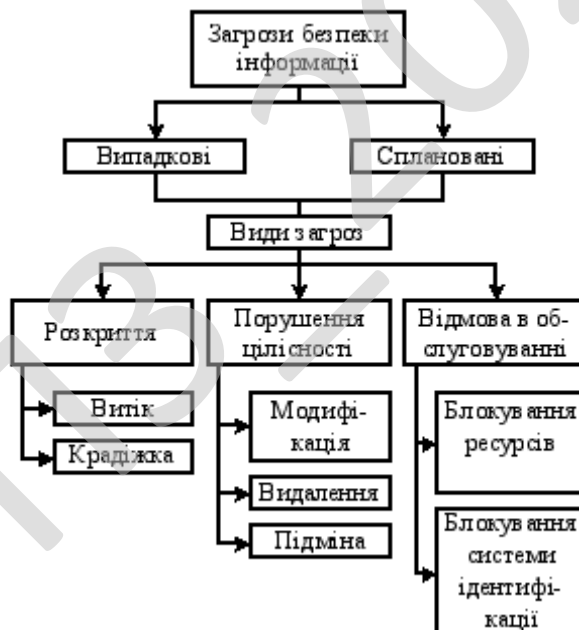


Рисунок 4.14 – види загроз безпеки інформації в КМ

Найбільш поширені загрози

Найпоширенішими та найнебезпечнішими (з точки зору шкоди) є ненавмисні помилки звичайних користувачів, операторів, системних адміністраторів та інших осіб, які обслуговують комп'ютерну мережу. Іноді такі

помилки насправді є загрозами (неправильно введені дані або програмна помилка, яка спричинила збій системи), іноді вони створюють уразливості, якими можуть скористатися зловмисники (як правило, адміністративні помилки). За деякими даними, до 65% втрат є результатом ненавмисних помилок.

Не стільки бід приносять пожежі та повені, скільки безграмотність і недбалість у роботі.

Очевидно, що найбільш радикальним способом боротьби з ненавмисними помилками є максимальна автоматизація та жорсткий контроль.

Інші загрози доступності можна класифікувати відповідно до компонентів інформаційної системи, які є метою загрози:

- відмова від користувача;
- збій внутрішньої мережі;
- вихід з ладу підтримуючої інфраструктури.

Для користувачів зазвичай розглядаються такі загрози:

- небажання працювати з інформаційною системою (найчастіше проявляється при необхідності освоєння нових можливостей і при невідповідності вимог користувача реальним можливостям і технічним характеристикам);
- невміння працювати з системою через відсутність відповідної підготовки (відсутність загальної комп'ютерної грамотності, невміння інтерпретувати діагностичні повідомлення, невміння працювати з документацією тощо);
- неможливість роботи з системою через відсутність технічного забезпечення (неповна документація, відсутність довідкової інформації тощо).

Програмні атаки

Велике навантаження (зазвичай пропускну здатності мережі, обчислювальних можливостей процесора або оперативної пам'яті) можна використовувати як засіб для виведення мережі з нормального режиму роботи.

За місцем розташування джерела загрози таке споживання поділяється на локальне та віддалене. При прорахунках у конфігурації системи локальна програма здатна практично монополізувати процесор і/або фізичну пам'ять, зводячи швидкість виконання інших програм до нуля.

Найпростішим прикладом віддаленого споживання ресурсів є атака під назвою «SIN-flood». Це спроба переповнити таблицю сервера "напіввідкритих" TCP-з'єднань (встановлення з'єднань починається, але не завершується). Така атака принаймні ускладнює легальним користувачам встановлення нових з'єднань, тобто сервер виглядає недоступним.

Останнім часом дистанційне споживання ресурсів проявилось в особливо небезпечній формі - у вигляді скоординованих розподілених атак, коли цілком легальні запити на підключення та/або обслуговування надсилаються на сервер з багатьох різних адрес на максимальній швидкості. Початком «моди» на такі атаки можна вважати лютий 2000 року, коли жертвами стали кілька найбільших систем електронної комерції (точніше, власників і користувачів систем). Якщо є недолік архітектури у вигляді дисбалансу між пропускною здатністю мережі та продуктивністю сервера, надзвичайно важко захиститися від атак розподіленої доступності.

Завдання забезпечення безпеки

Захист інформації в каналах зв'язку та базах даних за допомогою криптографічних методів;

- аутентифікація об'єктів даних і користувачів (автентифікація сторін, що спілкуються);

- виявлення порушень цілісності об'єктів даних;

- забезпечення захисту технічних засобів і приміщень, в яких здійснюється обробка конфіденційної інформації, від витоку побічними каналами та від можливих вбудованих у них електронних пристроїв захоплення інформації;

- забезпечення захисту програмних продуктів і комп'ютерної техніки від впровадження в них програмних вірусів і закладок;
- захист від несанкціонованих дій на каналі зв'язку з боку осіб, які не мають права використовувати засоби шифрування, але метою яких є компрометація секретної інформації та дезорганізація роботи абонентських пунктів;
- організаційно-технічні заходи, спрямовані на забезпечення безпеки конфіденційних даних.

Схема передачі даних розумного дому та виявлення потенційної небезпеки

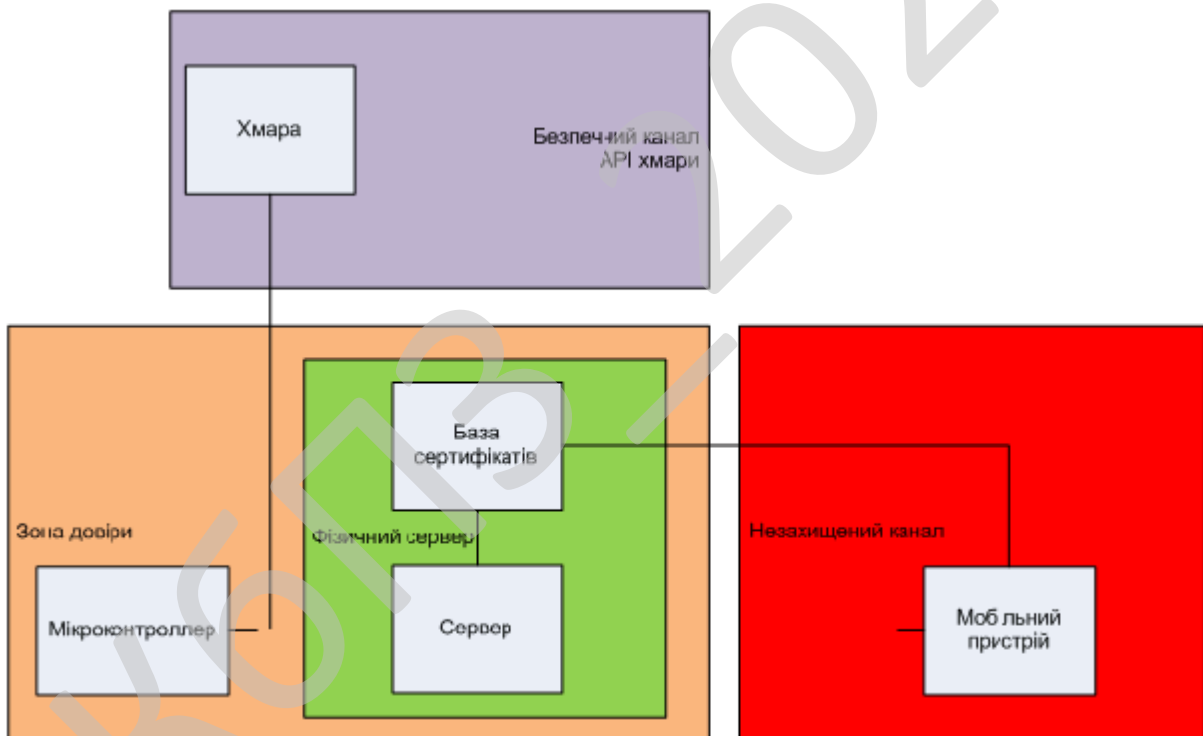


Рисунок 4.15 – Схема передачі даних в системі «Розумний дім»

База сертифікатів – це частина фізичного сервера, де зберігаються всі цифрові підписи, до яких логічний сервер має повний доступ, а користувач має частковий доступ зі свого мобільного пристрою.

Мікроконтролер - це пристрій, який безпосередньо відповідає за управління розумним будинком.

У локальній мережі (мережі сервера) дані вважаються умовно захищеними.

Небезпека полягає в незахищеному каналі користувача. Одним із класичних сценаріїв є «людина посередині», тобто здатність інших осіб підключатися до каналу між сервером і користувачем і видавати себе за одну з цих ключових осіб, взявши на себе роль невидимого посередника.

Для запобігання витоку інформації слід використовувати схеми шифрування даних.

Вибір системи шифрування

Отже, давайте розберемо системи та алгоритми шифрування. Серед систем система з використанням РКІ є найбільш оптимальною, оскільки забезпечує надійний канал для передачі сеансового ключа між сервером і клієнтом. В якості сертифіката ми візьмемо якийсь ключ, відомий тільки користувачеві та базі даних сертифікатів (у нашому випадку це серверний компонент), і база даних сертифікатів зберігає ключ разом із відкритим ключем користувача. Тоді система передачі даних матиме наступний вигляд (для зручності розіб'ємо її на 2 фази: авторизація та прямий зв'язок між 2 вузлами):

Авторизація

- клієнт підписує код «SH0» (що означає запит авторизації) своїм секретним ключем;
- сервер приймає повідомлення і відправляє його в базу сертифікатів;
- якщо база даних сертифікатів знаходить вказаний підпис, виконується авторизація, тобто: сервер отримує відкритий ключ клієнта з бази даних і надсилає сеансовий ключ, зашифрований цим ключем AES, включаючи фразу «SH1»; інакше створюється виняток "SH4".

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		68

Сесія «спілкування»

Після отримання ключа сеансу клієнт і сервер тепер можуть передавати дані один одному за допомогою парольної фрази "SH2".

Після завершення клієнт надсилає «SH3» із закодованою фразою «вихід», що призводить до припинення сеансу та знищення ключа сеансу відповідно.

Раз на рік система сертифікатів повинна регенерувати ключі, щоб звести ймовірність злому до мінімуму.

Не варто забувати, що існують і безпосередньо мікроконтролери, що відповідають за зчитування і управління пристроями. З самого початку ми визначили локальну мережу сервера як «зону довіри», але існує ймовірність того, що злочинець знаходиться поблизу сервера, але відносно недовго, тобто настільки, що неможливо знайти ключ для такого тривіального алгоритму шифрування, як Base64. Цей алгоритм цілком підходить для таких цілей, тому що він не перевантажує відносно малопотужні мікроконтролери, забезпечуючи тим самим певний захист інформації, і в той же час не вносить зайвих затримок.

Тест підсистеми безпеки

Для перевірки правильності роботи описаного вище модуля пропонуються наступні тести:

1) Перевірка диференціації команд:

- спробуйте запустити, відправивши повідомлення (SH2), очікуваним результатом буде відключення SH4;

- відправити запит на авторизацію (SH0) при використанні дійсного сертифіката. Очікуваний результат – SH1;

- відправити запит авторизації (SH0) при використанні неіснуючого сертифіката. Очікуваний результат – SH4;

- завершити сеанс і спробувати надіслати повідомлення (SH2), очікується відповідь SH4;

2) перевірити правильність шифрування невеликих обсягів даних:

- локальне шифрування повідомлень «Hello, world!»;

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		69

- зашифрувати попереднє повідомлення та передати його через Інтернет;
- шифрування та пересилання зображень по мережі;

3) Перевірка правильності шифрування великих обсягів даних:

- шифрування файлів однотипними символами. Перевірити створений файл на «простоту»;
- шифрування та передача повідомлень вагою до 1 Мб;
- перевірка роботи системи сертифікатів;
- використовувати сертифікат функціонування, пов'язаний з цим мобільним пристроєм, очікується позитивна реакція;
- використовувати сертифікат роботи іншого пристрою, очікується відключення;
- використовувати неіснуючий сертифікат. Очікується виняток;
- спроба нападу "людина посередині";
- створити міні-програму на стороні сервера, яка буде виконувати роль посередника між логічним сервером і клієнтом;
- змінити коди. Очікуваним результатом є завершення роботи SH4 для всіх випадків;

Так, під час роботи над даним блоком було досліджено роботу схем шифрування:

Симетричні системи

- системи відкритих ключів;
- інфраструктура відкритих ключів;

Алгоритми шифрування:

симетричний:

- Base64;
- AES;
- DES;

Асиметричний:

- PDA;

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		70

- еліптичні криві.

Визначено змішану схему шифрування з використанням Base64, AES, RSA, тести для її перевірки та програмну реалізацію, яка їм задовольняє.

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

На рисунку 4.16 показана блок-схема алгоритму роботи основної програми управління.

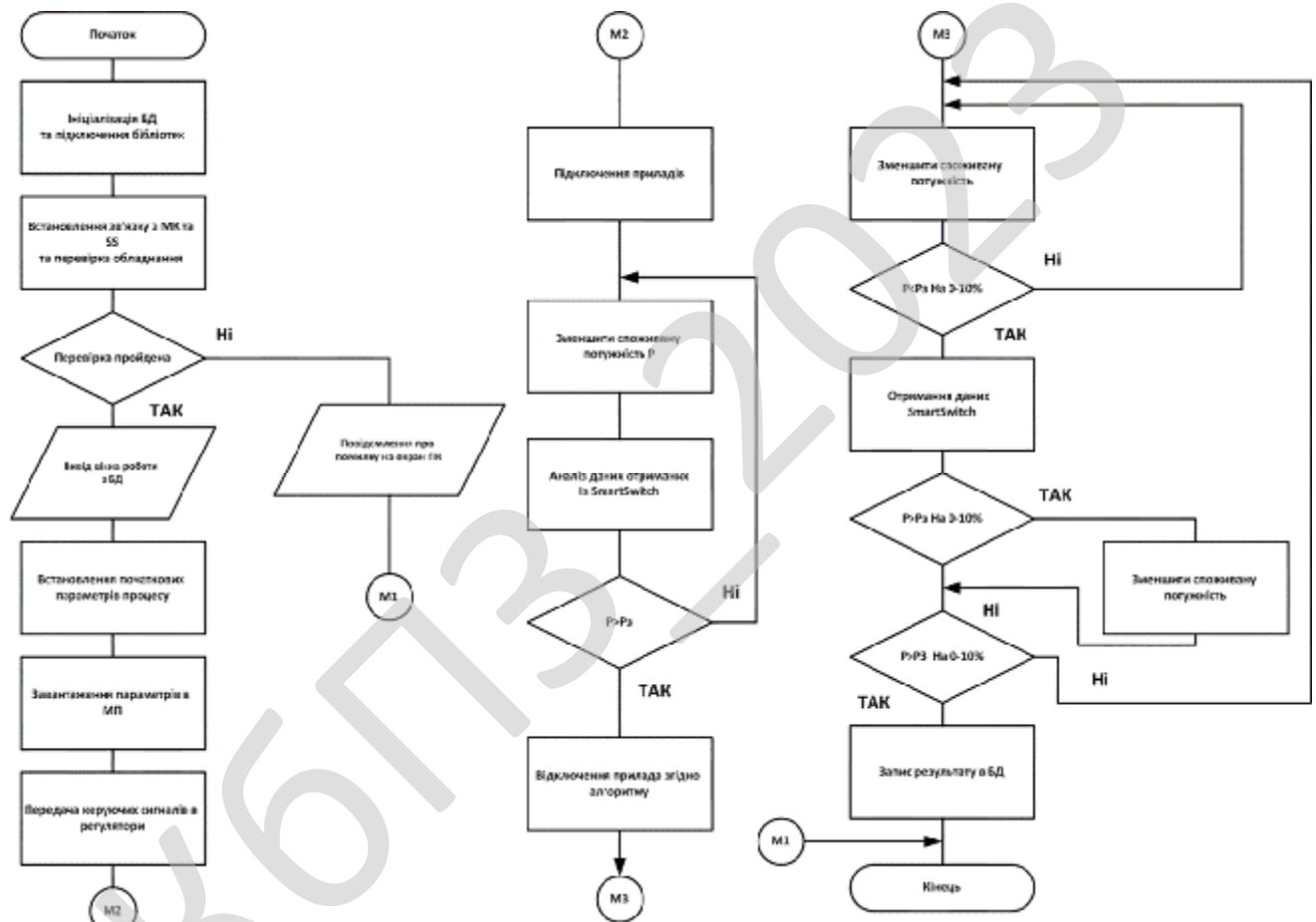


Рисунок 4.16 - блок-схема роботи основної програми управління

Вим.	Арк.	№ докум.	Підпис	Лат
------	------	----------	--------	-----

Розглянемо алгоритм роботи з онлайн-програмами. Його структурна схема показана на рисунку 4.17.

З рисунка видно, що після запуску програми спочатку відображається головне вікно програми. Потім база даних ініціалізується та пов'язуються відповідні бібліотеки.

На наступному етапі автоматично встановлюється зв'язок між мікроконтролером і «розумними розетками» системи. Якщо ця перевірка пройдена та отримано відповідний сигнал, система починає сканування всіх пристроїв системи з виведенням даних на монітор.

Наступним кроком є установка параметрів системи за замовчуванням і передача керуючих сигналів на мікропроцесор. Після встановлення параметрів система аналізує отримані параметри та, у разі перевищення заданого енергоспоживання, відключає резервні пристрої за алгоритмом, записаним у базі даних.

Так відбувається один із кроків управління системою «Розумний дім».

Всі інші системи працюють за аналогічним алгоритмом.

					БКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		72

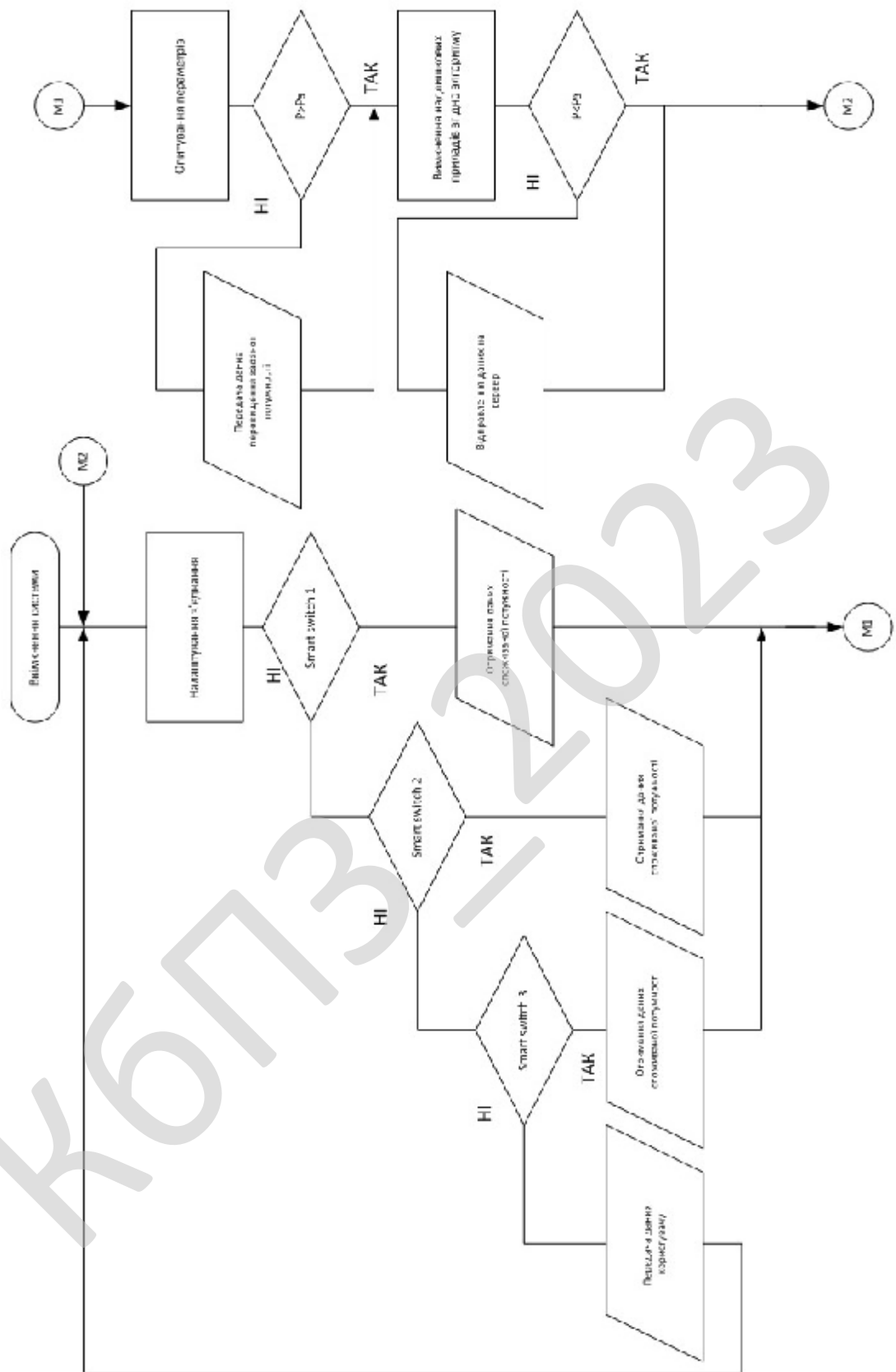


Рисунок 4.17 - блок-схема алгоритму з системи

Вим.	Арк.	№ докум.	Підпис	Лам
------	------	----------	--------	-----

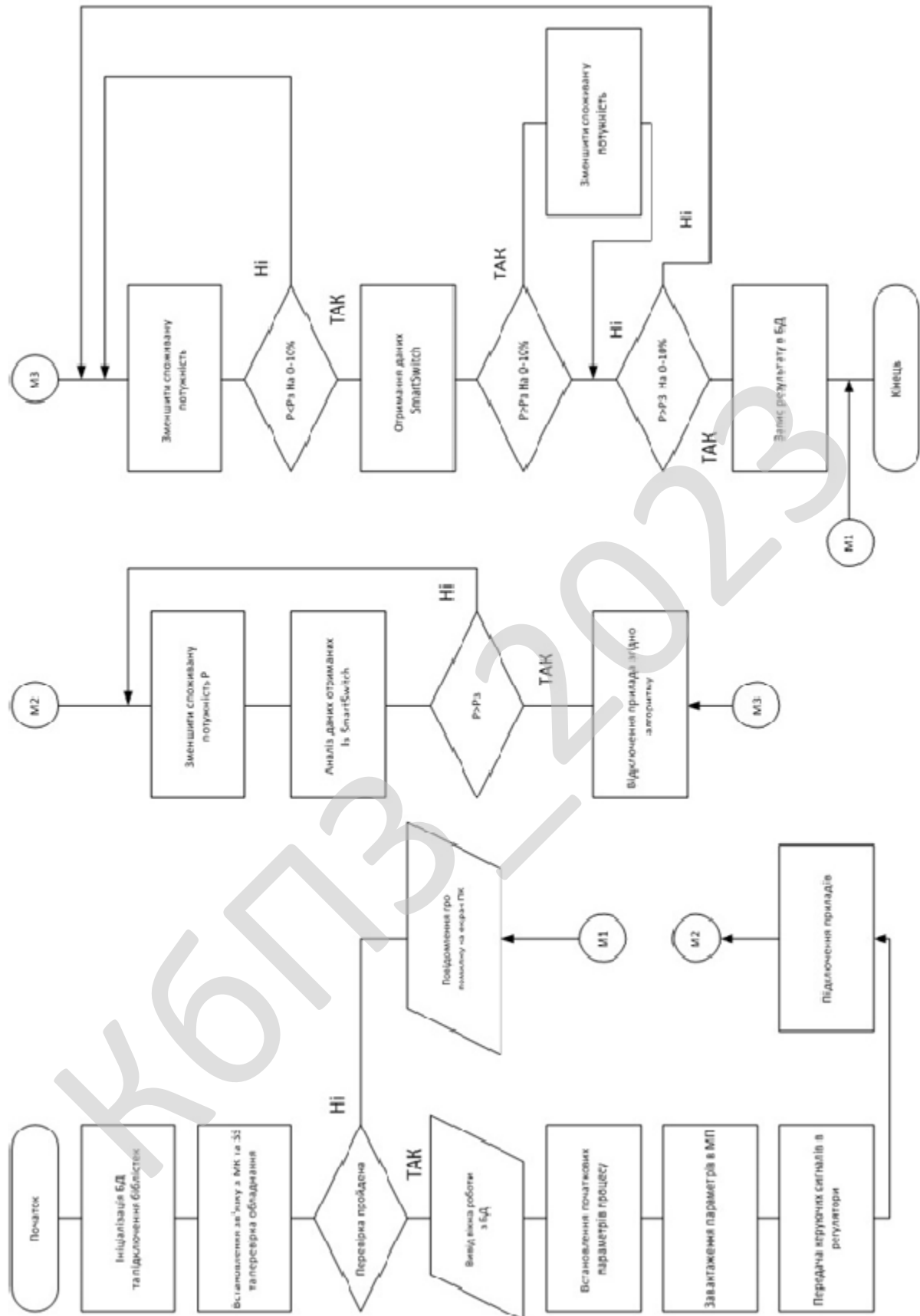


Рисунок 4.18 - Блок-схема основного алгоритму роботи системи

Вим.	Арк.	№ докум.	Підпис	Лат
------	------	----------	--------	-----

4.2 Захист розробленого програмного забезпечення

Стандартні функції мови PHP `password_hash()` і `password_verify()` були використані для захисту даних розробленого програмного забезпечення веб-системи командного управління проектами.

`password_hash()` використовується для хешування пароля користувача, а `password_verify()` використовується для перевірки вказаного пароля на хеш.

Перший параметр функції `password_hash()` - це пароль, який потрібно захистити, а другий параметр визначає алгоритм, за яким буде обчислюватися хеш. Зараз підтримуються два алгоритми хешування:

- `PASSWORD_DEFAULT` - використовує алгоритм `BCrypt`, який забезпечує хеш із 60 і більше символів;
- `PASSWORD_BCRYPT` - використовує алгоритм `CRYPT_BLOWFISH` для створення хешу. У цьому випадку результатом завжди буде рядок із 60 символів.

Основна перевага цієї функції полягає в тому, що розробнику не потрібно турбуватися про значення «соль» і вартість обчислення хешу, тому що ця функція робить все за розробника. Якщо розробник хоче використовувати власне значення солі, він повинен передати його в третьому параметрі функції.

Приклад використання цієї функції: `<?php echo password_hash("rasmuslerdorf", PASSWORD_DEFAULT); ?>`

Результатом виконання функції буде наступне значення:

```
$2y$10$.vGA109wmRjrwAVXD98HNOgsNpDczlqm3Jq7KnEd1rVAGv3Fykk1a
```

Приклад з передачею свого значення «солі» параметром функції:

```
<?php
$options = [ 'cost' => 12 ];
echo password_hash("rasmuslerdorf", PASSWORD_BCRYPT,
$options);
?>
```

Результатом виконання функції буде наступне значення:

```
$2y$12$QjSH496pcT5CEbzjD/vtVeH03tfHKFy36d4J0Ltp3lRtee9HDxY3K
```

Як бачимо два значення відрізняються один від іншого.

Також сайт було захищено від таких атак як: XSS-атаки та SQL-ін'єкції.

Від XSS-скриптів сайт було захищено наступним чином: достатньо замінювати спеціальні символи "<" і ">" на "<" і ">", що виконує php-функція `htmlspecialchars` та `strip_tags`. Також можна написати у файлі `.htaccess` такий код:

```
Options +FollowSymLinks
RewriteEngine On
RewriteCond %{QUERY_STRING} (\<|%3C).*script.*(\>|%3E) [NC,OR]
RewriteCond %{QUERY_STRING} GLOBALS(=|\[[|\%[0-9A-Z]{0,2}) [OR]
RewriteCond %{QUERY_STRING} _REQUEST(=|\[[|\%[0-9A-Z]{0,2})
RewriteRule ^(.*)$ index.php [F,L]
```

Для захисту сайту від SQL-ін'єкцій було використано php-функцію `mysql_real_escape_string`. Ця функція екранує спеціальні символи в рядку, що використовуються в SQL-запиті, беручи до уваги кодування з'єднання.

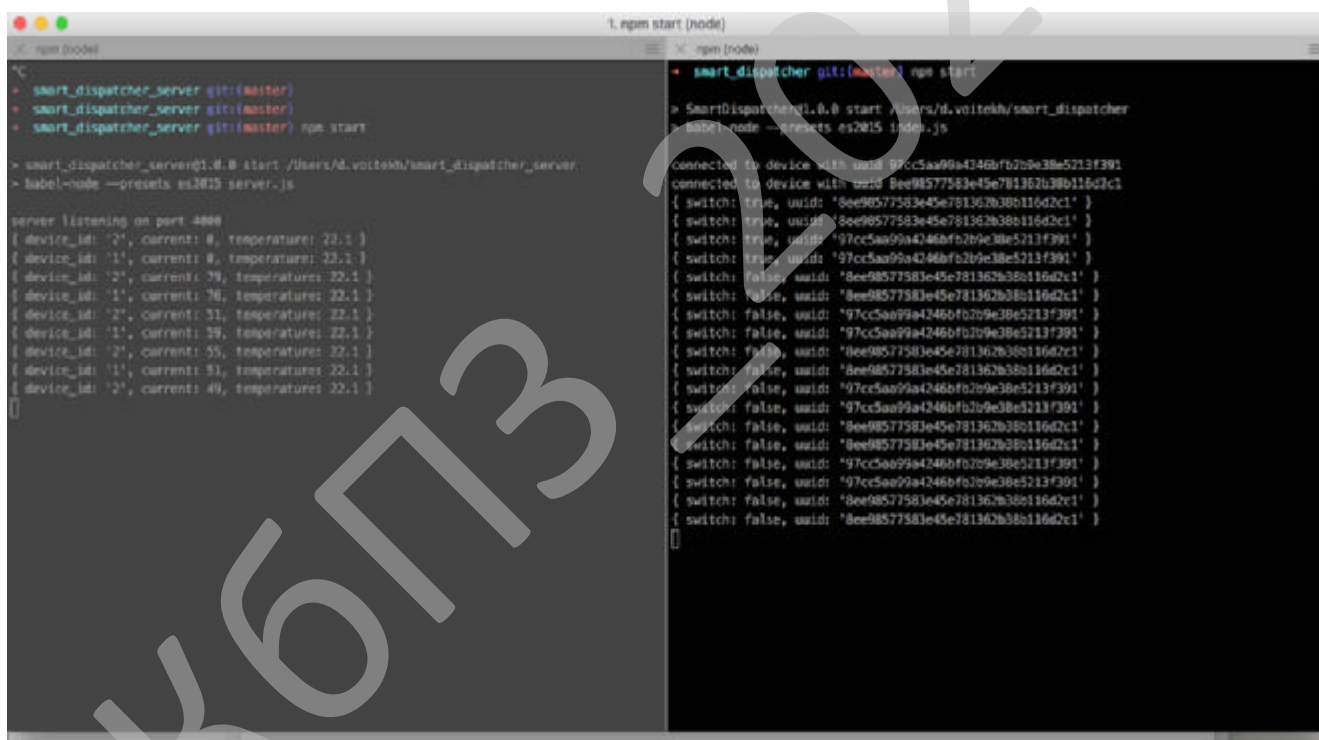
					БКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		76

5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Щоб проілюструвати, як ці системи працюють, давайте розглянемо фактичні сценарії, які вони описують деякі особливості запропонованого алгоритму оптимізації, деталі загального контролю та управління налаштуваннями:

- 1) Підключіть блок живлення до Smart Switch пристрою.
- 2) За допомогою команди `npm start` запускаємо обидва сервера.

Журнали пристрою, який підключається до мережі Bluetooth LE та обмінюється повідомленнями з веб-мережею, показані на рисунку 5.1.



```
smart_dispatcher_server git:(master) npm start
> smart_dispatcher_server@1.0.0 start /Users/d.voitek/h/smarter_dispatcher_server
> label-node --presets es2015 server.js

server listening on port 4000
[ device_id: '2', current: 8, temperature: 22.1 ]
[ device_id: '1', current: 79, temperature: 22.1 ]
[ device_id: '2', current: 39, temperature: 22.1 ]
[ device_id: '1', current: 76, temperature: 22.1 ]
[ device_id: '2', current: 31, temperature: 22.1 ]
[ device_id: '1', current: 39, temperature: 22.1 ]
[ device_id: '2', current: 55, temperature: 22.1 ]
[ device_id: '1', current: 31, temperature: 22.1 ]
[ device_id: '2', current: 49, temperature: 22.1 ]

smart_dispatcher git:(master) npm start
> SmartDispatchers21.0.0 start /Users/d.voitek/h/smarter_dispatcher
> label-node --presets es2015 index.js

connected to device with uuid: 97cc5aa99a4246bf2b9e38e5213f391
connected to device with uuid: Bee98577583e45e781362b38e116d2c1
{ switch: true, uuid: '8ee98577583e45e781362b38e116d2c1' }
{ switch: true, uuid: '8ee98577583e45e781362b38e116d2c1' }
{ switch: true, uuid: '97cc5aa99a4246bf2b9e38e5213f391' }
{ switch: true, uuid: '97cc5aa99a4246bf2b9e38e5213f391' }
{ switch: false, uuid: '8ee98577583e45e781362b38e116d2c1' }
{ switch: false, uuid: '8ee98577583e45e781362b38e116d2c1' }
{ switch: false, uuid: '97cc5aa99a4246bf2b9e38e5213f391' }
{ switch: false, uuid: '97cc5aa99a4246bf2b9e38e5213f391' }
{ switch: false, uuid: '8ee98577583e45e781362b38e116d2c1' }
{ switch: false, uuid: '8ee98577583e45e781362b38e116d2c1' }
{ switch: false, uuid: '97cc5aa99a4246bf2b9e38e5213f391' }
{ switch: false, uuid: '97cc5aa99a4246bf2b9e38e5213f391' }
{ switch: false, uuid: '8ee98577583e45e781362b38e116d2c1' }
{ switch: false, uuid: '8ee98577583e45e781362b38e116d2c1' }
{ switch: false, uuid: '97cc5aa99a4246bf2b9e38e5213f391' }
{ switch: false, uuid: '97cc5aa99a4246bf2b9e38e5213f391' }
{ switch: false, uuid: '8ee98577583e45e781362b38e116d2c1' }
{ switch: false, uuid: '8ee98577583e45e781362b38e116d2c1' }
```

Рисунок 5.1 – Процес запуску системи

3) Далі ми перейдемо в браузер за посиланням <http://localhost:3000/device/1>, щоб побачити поточний стан пристрою. На малюнку 3.8 показано це розташування. Ми бачимо, що такі дані, як поточний стан і енергоспоживання пристрою, реєструються. На даний момент двигун споживає 2,1120 Вт

електроенергії і його стан вважається задовільним. Ми бачимо зміну агрегованих даних з таблиці журналів на графік добового споживання пристрою.



Рисунок 5.2 – Дані про споживання приладу

4) Тепер перемикаємо навантаження пристрою, натискаючи кнопку «Переключити». В результаті двигун заглушився. Оновлено інформацію на екрані - текст `is_on = false`, значення `current_power` змінено на 0.

5) Переходимо в автоматичний режим роботи, на сторінці http://localhost:3000/settings_page виставляємо значення параметра `AUTO_MODE = true`.

6) Ми бачимо, що двигун, який був вимкнений вище, увімкнувся, тому що алгоритм знайшов збіг у часовому інтервалі `['08:20:00', '22:00:00']`. Отже, якщо змінити цей інтервал так, щоб поточний час не потрапляв у нього, двигун вимкнеться.

7) Опишемо частину алгоритму, пов'язану з пошуком дефектів зразка. У цьому прикладі `optimal_current` вказано як 0,08 А. Я зупиню двигун, щоб

зменшити споживання електроенергії. Коли споживаний струм перевищує позначку 0,16А, двигун відключається, с. відповідно `is_valid = false`.

Тому для подальшої роботи з принтером необхідно змінити це значення.

8) Напишемо частину алгоритму балансування навантаження. На сторінці налаштувань ми вказуємо максимальну потужність системи як `POWER = 5`, що відповідає 5 Вт. Увімкніть обидва двигуни пристрою. Загальна споживана потужність становить 4,3 Вт. Зупиняємо двигун першого принтера кнопкою стоп, щоб збільшити струм споживання до 0,12А. Відбулося збільшення загальної потужності до позначки 5 Вт. Цей двигун вимкнено. У наступному циклі отримання даних від цього принтера двигун знову включиться.

Таким чином, при проектуванні цієї системи задоволені всі перераховані нижче вимоги.

КБГІЗ-2023

					БКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		79

6 НАУКОВА НОВИЗНА

У магістерській роботі розроблено програмне забезпечення, яке призначено для системи управління розумним будинком з впровадженням системи захисту від кібератак.

Метою розробки є дослідження та програмна реалізація системи управління розумним будинком з підсистемою безпеки передачі даних та захисту від кібератак.

Об'єктом дослідження є процес управління розумним будинком.

Предметом дослідження є методи реалізації систем управління розумним будинком.

Методи дослідження базуються на методах теорії кодування, методах математичної статистики, методах захисту інформації, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- удосконалено систему управління розумним будинком з впровадженням підсистеми безпеки передачі даних та захистом від кібератак;
- проведено огляд технологій зв'язку в системах «Розумний дім»;
- розроблено вітчизняний продукт управління розумним будинком з підсистемою безпеки передачі даних, який має більш широкі можливості, на відміну від існуючих аналогів.

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми дипломного проекту

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 48 днів (два місяці).

В магістерській роботі проведено дослідження та виконана програмна реалізація системи управління розумним будинком з впровадженням системи захисту від кібератак.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;

Таблиця 7.1 - Початкові данні

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт	N	1
2. Кількість екземплярів програм, шт	Ne	58
3. Запланований термін розробки, днів	Fpq	48 (2 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2
7. Кількість макетів вхідної інформації	–	3

Продовження таблиці 7.1

1	2	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	1
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	60000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	55
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B \quad (7.1)$$

де А - коефіцієнт Боєма, А=2,45;

Size - загальний об'єм відлагодженого програмного коду, тис. рядків;

В - показник ступеня, що визначається співвідношенням

$$B = 1,01 + 0,001 \sum W_i \quad (7.2)$$

де W_i - сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B=1,01+0,001(2,43+3,64+3,38+3,95+2,73) = 1,026$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де $\prod V_j$ - добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де C - визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S - коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПО згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{РП} = 0,3 \cdot 3,23 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 82 = 168 \text{ люд/день}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

Таблиця 7.2 - Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	168	Ф 7.1-7.4
Впровадження	13	Д13
Всього	209	—

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою

$$Ч = \frac{T_{нз} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де F_{pq} - плановий фонд робочого часу одного спеціаліста, днів,

$T_{нз}$ – трудомісткість розробки програмного забезпечення люд-дні,

$$Ч = \frac{209 \cdot 1}{48 - 5} = 4,8 \text{ ставки}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3

Таблиця 7.3 - Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	12	1080	18
Монітор	60	12	720	12
Клавіатура	30	12	360	6
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	3	360	6
Принтер струминний	60	2	120	2
Сканер	20	1	20	0,33
Концентратор–маршрутизатор	30	6	180	3
Кабельні господарства ЛВС на 1 м. п.	2,5	400	1000	16,67
Копіювальний апарат	140	2	280	4,67
Усього за рік:			3 _ч	74,67

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{\text{др}}^c = \frac{Z_{\text{ч}} \cdot n_{\text{міс}}}{1,2} \quad (7.6)$$

$$\Phi_{\text{др}}^c = \frac{75 \cdot 3}{1,2} = 187,5 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}} \quad (7.7)$$

$$Ч_{ел} = 187,5 / (60 \cdot 8) = 0,4 \text{ ставки}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів–електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 - Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2022, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	0,8	0,2
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,2	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,2	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	0,4	
Всього		1,6	

Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	0,5	0,2
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,3	
	Розміщення графіки і контенту на Інтернет сторінках	0,3	
Всього		1,6	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,2
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		1,6	

Складемо штатний розклад виконавців у таблицю 7.5.

Таблиця 7.5 - Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	12098,5	24197
Продакт-менеджер	0,5	12000	12000
Інженер-програміст	4,8	14000	134400
Інженер-електронщик	0,4	9000	7200
Інженер-системотехнік	0,2	9000	3600
Адміністратор мережі	0,2	11000	4400
Системний програміст	0,2	9000	3600
Дизайнер WEB	0,2	9000	3600
Інженер-верстальник	0,2	9000	3600
Бухгалтер-економіст	0,2	10000	4000
Всього за період розробки	$R_{cn}=7,9$	-	$\Phi_{роб}=200597$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{200597}{7,9 \cdot 48} = 529 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		89

$$B_{y\partial} = R_{cn}^1 S_y \Pi_{nl}, \quad (7.9)$$

де R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць.

S_y – питома площа на одне робоче місце, m^2 ,

Π_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 400...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн на одне робоче місце. Тобто

$$I_{нв} = R_{cn}^1 \cdot \Pi_m, \quad (7.10)$$

де Π_m – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 8 \cdot 3500 = 28000 \text{ грн}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7. Дані по оптовій ціні на обладнання та комплектуючі вибирались за комерційною пропозицією фірми Brain за 10.11.23 – джерело <http://brain.com.ua/>

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		90

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11771
Системний блок		7771
Процесор	INTEL Pentium G6405 (BX80701G6405) 1200, 2 ядра, 4 потоки, 4.1 GHz, TDP - 58 Вт, 14nm, BOX	-
Системна плата	ASRock H510M-HDV R2.0 сокет - 1200, DDR4, LAN - 1 Гбіт/с, D-Sub (VGA), DVI, HDMI, - 4 x SATA 6.0 Gb/s, Micro-ATX	-
Жорсткий диск	SSD M.2 2280 240GB Apacer 240 GB, TLC, M.2, SATA III (6Gb/s)	-
Оперативна пам'ять	DDR4 8GB 2400 MHz Patriot	-
Відеокарта	Вбудована, Intel UHD Graphics 610	-
DVD-привод	DVD±RW ASUS DRW-24B5ST Black Bulk	-
Корпус	ProLogix E105 Minitower, Micro - ATX	-
Кардрідер внутрішній	Transcend TS-RDF8K USB 3.0	-
інше	Клавіатура, мишка	-
Монітор	Монітор BenQ GL2450HM Black	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37	2800
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	Powercom BNT-600AP USB	1400

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 - Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	11771	9416,8	103584,8
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	125216,3

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	125216	-	-
Всього по групі	125216	50	62608

Продовження таблиці 7.8

1	2	3	4
Група 5, 6			
4. Вимірювальні пристрої	5190	-	-
5. Транспортні засоби	143000	-	
6. Господарський інвентар	28000	-	-
Всього по групі	176190	20	35238
7. Нематеріальні активи	60000	10	6000
Разом	$K_p = 1769406$		$A_p = 174246$

Примітка: вартість автомобіля взята по даним з автосалону «Кіровоград-Авто», джерело [http://kirovograd-avto.ukravto.ua /catalog/tm-9/model-80/description](http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description), складає 143000 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де N_e – Кількість екземплярів програм, шт.

$$Z_o = 529 \cdot 209 / 58 = 1906 \text{ грн}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де H_q – норматив додаткової зарплати, %

$$Z_d = 1906 \cdot 10 \cdot 0,01 = 191 \text{ грн}$$

Відрахування на соціальні потреби за нормативом $H_c=22\%$ від суми основної та додаткової зарплати

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де H_c – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22(1906+191) = 461 \text{ грн}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z=15\%$ від основної зарплати

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де H_z – загальногосподарські витрати, %

$$G_{ocn} = 1906 \cdot 15 \cdot 0,01 = 286 \text{ грн}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де Z_{M1} – вартість паперу, грн., Z_{M2} – вартість запам'ятовуючих пристроїв, грн., Z_{M3} – вартість фарби, картриджів, тонеру, грн., N_e – кількість екземплярів програм, шт.

Згідно виданих норм приймаємо 1/3 пачку паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n=240$ грн., визначаємо вартість паперу за період розробки $N_m=2$ міс:

$$Z_{M1} = C_n \cdot N. \quad (7.16)$$

$$Z_{M1} = 240 \cdot 1/3 = 80 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості, що дорівнює кількості 10 CD + 1 DVD:

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 40 грн/шт., DVD-R LG 4,7Gb, 16x speed Cake box - 74 грн/шт.

$$Z_{M2} = 40 \cdot 10 + 74 = 474 \text{ грн.}$$

Згідно виданих норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		94

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1	2	3
1. Основна зарплата виконавців	Z_o	1906
2. Додаткова зарплата виконавців	Z_o	191
3. Відрахування на соціальні потреби	C_{oc}	461
4. Загальногосподарські витрати	G_{ocn}	286
5. Витрати на матеріали	Z_M	39
6. Освоєння нових операційних систем, мов програмування	O_n	286
7. Амортизація основних фондів	A_M	501
8. Повна собівартість програмного забезпечення	C_n	3670
9. Плановий прибуток	P_p	2018,5
10. Ціна підприємства $C_n = C_n + P_p$	C_n	5688,5
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	1137,7
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	6826,2

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 - Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн	
	Базовий	Новий
Вартість програмної продукції	–	6826
Всього капітальних витрат	–	6826

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 - Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування	<i>З_р</i>	29524	21472
2. Витрати на електроенергію	<i>З_{ел}</i>	0	0
3. Витрати на амортизацію	<i>З_{ам}</i>	0	3413
Всього витрат за рік	<i>I</i>	29524	24885

Витрати на технічне обслуговування:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де T_p – кількість годин обслуговування системи за рік, год.,

Z_z – заробітна плата обслуговуючого персоналу, грн/год

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 220 годин на рік до 160 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 220 \cdot 100 \cdot 1,1 \cdot 1,22 = 29524 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 160 \cdot 100 \cdot 1,1 \cdot 1,22 = 21472 \text{ грн.}$$

Витрати на електроенергію визначаються з урахуванням спожитої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$).

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

Витрати на електроенергію при впровадженні нової системи не змінюються.

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 - Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн., за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	6826	–	3413
Всього відрахувань	-	–	6826	–	3413

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою

$$E_e = (C_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.; E_p – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (5688 - 3670) \cdot 58 - (0,05 \cdot 1408000 + 0,5 \cdot 125216 + 0,2 \cdot 176190 + 0,1 \cdot 60000) \cdot 2/12 = 88003 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p^* – балансова вартість основних фондів розробника.

$$T_e = \frac{1769406}{(5688 - 3670) \cdot 58 \cdot 12 / 2} = 2,52 \text{ років}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n (K_n - K_{\bar{o}}), \quad (7.27)$$

де $I_{\bar{o}}, I_n$ – величина експлуатаційних витрат за базовим и новим варіантом відповідно, $K_{\bar{o}}, K_n$ – об'єм капітальних вкладень за варіантами, що порівнюються

$$E_{cn} = (29524 - 24885) - 0,5 \cdot 6826 = 1226 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n} \quad (7.28)$$

$$T_{cn} = \frac{6826}{29524 - 24885} = 1,47 \text{ року}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 - Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	58
2. Повна собівартість розробленої програми	Грн.	3670
3. Ціна розробленої програми	Грн.	5688
4. Плановий прибуток від реалізації розробленої програми	Грн.	2018
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1769406
7. Загальний прибуток від реалізації програмної продукції	Грн.	117044
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	88003
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	2,52
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	6826
11. Величина економічного ефекту у користувача програмної продукції	Грн.	1226
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	1,47

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

КБГПЗ - 2023

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		101

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Аналіз умов праці програміста

Інтернет відіграє важливу роль у житті сучасної людини. Кожного дня мільйони людей використовують Інтернет для пошуку необхідної інформації, спілкуванні у соціальних мережах, перегляду новин. Багато людей користуються Інтернетом у професійних цілях, оскільки завдяки Інтернету з'явилося багато нових професій. Тому для веб-розробника так важливо розробити зручний інтерфейс для зручного сприйняття інформації, та необхідний функціонал, який буде відповідати необхідним вимогам та навантаженням. Все це вимагає багато багато часу та великого навантаження з боку розробників.

Тому так важливо слідкувати за умовами праці, в яких відбувається робочий процес. Оскільки захворювання можуть бути спричинені надмірним фізичним або розумовим навантаженням, через велику нервово-емоційну напругу, або через виробниче середовище. В даному розділі магістерської роботи проведемо аналіз основних чинників при роботі програміста.

При роботі за компютером, розробник має велике зорове навантаження, тому йому необхідне належне освітлення приміщення. Якщо в приміщенні недостатньо природного освітлення, потрібно використовувати спеціальні світильники. Також оскільки розробник значний час працює з електричними приборами є можливість, бути ураженим електричним струмом, тому потрібно дотримуватись всіх необхідних норм. Серед основних чинників, які впливають на розробників під час трудової діяльності можна виділити[46]:

1. Рівень освітлення в приміщенні.
2. Температура, вологість в приміщенні.
3. Рівень шуму на робочому місці.

4 .Напруга в електричному ланцюзі, електричні показники.

Розберемо кожний з чиників окремо, проаналізуємо які повинні бути стандарти кожного з чиників, відповідно до правил з охорони праці.

Рівень штучного освітлення

Головним документом для встановлення норм необхідних показників освітлення є ДБН В.2.5-28:2018 «Природне та штучне освітлення» [47].

Сьогодні найбільш розповсюдженими є світлодіодні ламп. В середньому світловіддача від таких ламп знаходиться на рівні 80-120 Лм/Вт [48]. Джерелом живлення прийнято вважати електричну мережу у 220В. А освітленість робочого приміщення повина бути $E = 300-500$ Лк, оскільки робота програміста відноситься до робіт середньої точності з присвоєнням розряду зорових робіт IV[49].

Рівень сітла повинен бути достатнім, щоб працівник міг працювати без навантаження на зір. Це залежить від системи освітлення, кількості світильників, їх типу та розміщення у приміщенні. Допустиме значення освітленості робочої поверхні приймається $E = 400$ лк [50].

Для покращення освітлення комп'ютерній лабораторії будуть використовуватися світлодіодні лампи, а саме FL-LED T8-900 світловий потік яких $F=1500$ лм.

Мікроклімат робочої зони: температура, відносна вологості, швидкість руху повітря

Головним документом для встановлення норм мікроклімату робочої зони є ДСН 3.3. 6.042 -99 «Державні санітарні норми мікроклімату виробничих приміщень» [51].

Праця програміста за важкістю відноситься до легкої фізичної роботи категорії Ia [49]. Де вказано, що в приміщенні, я кому знаходиться компютерне обладнання, повинні бути встановлені певні норми, оскільки офісна техніка є джерелом тепловиділень, що може спричинити підвищення температури. Серед

потимальних параметрів для роботи встановлена температура 23 – 25⁰С і вологість на рівні 40 – 60% в залежності ві періоду року.

Для підтримки комфортної температури можна використовувати як організаційні методи, наприклад розпорядок дня, так і технічне обладнання, наприклад кондиționери, вентиляцію. Як правило в холодний період часу використовуються додаткове опалення для підтримання комфортної температури, а в літку встановлюються кондеціонери.

Рівень шуму на робочому місці

Як правило при використанні великої кількості компютерів в одному приміщенні, через гудіння, рівень шуму має значення більше норми. Допустима норма становить менше 50 дБ [52].

Гучний шум негативно впливає на умови праці та організм людини. Якщо шум триває тривалий час цу може спричинити головні болі, біль у вухах, підвищення стомлюваності, зниження концентрації та уваги. Такі симптоми можуть викликати стресові ситуації у людини. Все це шкодить продуктивності працівника та його стану здоровья.

Щоб встановити необхідний рівень шуму, використовують додаткову звукоізоляцію. Для цього найчастіше використовуються мати та плити із скляного та мінерального волокна, м'які плити з деревних стружок, картон, гуму, утеплений лінолеум, а також заміна вікон на звукоізолюючі.

8.2 Заходи профілактики при роботі з комп'ютерною технікою

Санітарно-гігієнічні норми є важливим критерієм при роботі в приміщенні. Від них залежить здорове працівників, їх рівень працездатності, втомлюваність. Щоб всього цього уникнути потрібно стежити за нормами на робочому місці.

Якщо говорити про електробезпеку в приміщенні, то в приміщенні необхідно устаткування розподільних щитів спеціальними розетками з

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		104

заземлюючими контактами, повинні бути заземлені всі прилади і пристрої, час від часу повина проводитися перевірка всіх приладів, щорічна здача іспитів з охорони праці.

Для оптимальних показників мікроклімату та освітленості потрібні використовувати дефлектор, для організації вентиляції, та повітрообміну. Та перевірку освітленості в приміщенні згідно відділом охорони праці, щоб відповідати нормам для зорової роботи .

Ще однією проблемою з якою часто зустрічаються програмісти є мала рухливість та повний сидячий робочий день. Тому рекомендується час від часу робити невеликі перерви, під час обіднього перериву вживати їжу не на робочому місці. У окремих випадках, коли при дотриманні всіх санітарних норм, працівник все одно себе погано почуває, дозволяється індивідуальний підхід для обмеження роботи з обчислювальними пристроями. Тривалість роботи за комп'ютером не повинна безперервно тривати більше 4 годин.

Для зменшення зорового та нервово-емоційного навантаження, та поліпшення мозкової діяльності рекомендується робити перерви для психологічного та фізичного розвантаження.

В приміщеннях також повинні бути протипожежне обладнання, та інструкція у разі надзвичайних ситуаціях. Повинна бути особа, яка відповідає за пожежну безпеку, перевіряє обладнання, та системи протипожежного захисту а також щорічне проведення інструктажів серед працівників.

Автоматична пожежна сигналізація повинна відповідати вимогам ДБН ДБН В.2.5-56:2014, яке вимагає використання вогнестійких кабелів та автоматичну роботу системи оповіщення та евакуації людей у випадку надзвичайної ситуації [53].

При перевірці, приміщення повинно відповідати всім нормам пожежної безпеки. Це виконується за допомогою перевірки пожежної охорони та техніки, проведенню інструктажу і своєчасне інформування пожежної охорони про несправність пожежної техніки, впровадження систем протипожежного

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		105

захисту. Організаційні та технічні заходи, спрямовані на попередження виникнення пожежі, обмеження поширення вогню та успішної евакуації людей.

8.3 Розрахнок занулення глухозаземленої нейтралі

Занулення, як основний засіб захисту, застосовується в електроустановках до 1 кВ з глухозаземленою нейтраллю трансформатора або генератора [54]. Початкові дані для розрахунку занулення глухозаземленої нейтралі трансформатора виробничого приміщення:

Загальна потужність: $P = 5$ кВт.

Кількість електродвигунів: $m = 10$

Потужність освітлювальних приладів: $P_o = 3$ кВт.

Довжина магістрального кабеля: $LM = 70$ м.

Довжина розгалудження: $l = 22$ м.

Лінійна напруга $U = 380$ В.

Фазна напруга $U_\phi = 220$ В.

Визначаємо силу номінального струму електроустановки:

$$I_{ном} = I_{max} = (P * 1000) / (\sqrt{3} * U_{л} * \cos\phi) \quad (8.1)$$

$$I_{ном} = I_{max} = (5 * 1000) / (\sqrt{3} * 380 * 0,85) = 8,9 \text{ А}$$

де: P – номінальна сумарна потужність електроприладів, кВт;

$U_{л}$ – лінійна напруга, В;

$\cos\phi$ – коефіцієнт потужності, приймається в залежності від типу електрообладнання в межах 0,8..0,87.

Визначаємо силу пускового струму електродвигуна:

$$I_{пус} = 5 * I_{ном} \quad (8.2)$$

$$I_{\text{пус}} = 5 * 8,9 = 44,5 \text{ А}$$

Визначаємо номінальну силу струму апарата захисту:

$$I_{\text{н}} = I_{\text{пус}}/b \quad (8.3)$$

$$I_{\text{н}} = 44,5 / 2,5 = 17,8 \text{ А}$$

b – коефіцієнт пуску електродвигуна – для легких умов пуску – 2,5..3.

Вибираємо запобіжник ПН 2-100 з плавкою вставкою $I_{\text{ном}} = 50 \text{ А}$.

Визначаємо найменше допустиме по умовам спрацьовування захисту значення сили струму короткого замикання:

$$I_{\text{ктіп}} = I_{\text{н}} * K \quad (8.4)$$

$$I_{\text{ктіп}} = 50 * 3 = 150 \text{ А}$$

$I_{\text{н}}$ – номінальний струм апарата захисту;

K – коефіцієнт надійності;

Знаходимо переріз провoda або кабеля розгалуження з умови допустимого нагрівання:

$$I_{\text{доп}} = I_{\text{вст}}/a \quad (8.5)$$

$$I_{\text{доп}} = 50 / 3 = 16,6 \text{ А}$$

Вибираємо площу перерізу 10 мм^2 ($S_{\text{ф}}$) при числі проводів $i = 4$ розташований у повітрі. Визначаємо максимальний робочий струм:

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		107

$$I_{роб} = K_0(K_3 \cdot (P \cdot 1000) / (\sqrt{3} \cdot U_L \cdot \cos\phi) \cdot m + K_3 \cdot (P_0 \cdot 1000) / (\sqrt{3} \cdot U_L \cdot \cos\phi)) \quad (8.6)$$

$$I_{роб} = 0,75 \cdot (0,85 \cdot ((5 \cdot 1000) / (1,73 \cdot 380 \cdot 0,85)) \cdot 10 + (3 \cdot 1000) / (1,73 \cdot 380 \cdot 0,85)) = 60,4 \text{ А}$$

K_0 – коефіцієнт одночасності роботи групи електроприймачів;

$$K_0 = 0.7 \dots 0.8; K_3 = 0.8 \dots 0.9;$$

K_3 – коефіцієнт завантажених електродвигунів;

$P_0 = 3 \text{ кВт}$ – потужність освітлювальної мережі;

Визначається струм короткочасного перевантаження магістрального кабеля:

$$I_{пер} = K_0(K_3 \cdot (P \cdot 1000) / (\sqrt{3} \cdot U_L \cdot \cos\phi) \cdot n + K_3 \cdot (P_0 \cdot 1000) / (\sqrt{3} \cdot U_L \cdot \cos\phi)) + I_{пус} \quad (8.7)$$

$$I_{пер} = 0,75 \cdot (0,85 \cdot (5 \cdot 1000) / (1,73 \cdot 380 \cdot 0,85)) \cdot 9 + 0,85 \cdot (30 \cdot 1000) / (1,73 \cdot 380 \cdot 0,85) + 44,5 = 130,0643 \text{ А}$$

Струм спрацювання електромагнітного розчеплювача додатково перевіряємо по максимальному струму перевантаження лінії:

$$I_{спр} \geq 1,25 \cdot I_{пер} \quad (8.7)$$

$$I_{спр} \geq 1,25 \cdot 130,0643 = 162,5803 \text{ А}$$

Приймаємо $I_{спр} = 162,5803 \text{ А}$. Вимикач : А3714Б.

Вибираємо площу перерізу S_{ϕ} магістрального кабеля (провідника) по Доп. $S_{\phi} = 70 \text{ mm}^2$, – кабель АВРГ прокладений в землі, $i=3$ (число проводів). Вибрану площу перерізу перевіряємо для автоматів з електромагнітним розчеплювачем:

$$I_{доп} \geq I_{спр} / 4,5 \quad (8.8)$$

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		108

$$I_{\text{доп}} \geq 162/4,5 = 36 \text{ А}$$

Проводимо узгодження з номінальним струмом автомата:

$$I_{\text{доп}} = I_{\text{спр}}/3 \quad (8.9)$$

$$I_{\text{доп}} = 162/3 = 54 \text{ А}$$

Значення 36 і 54 А. менше ніж $I_{\text{max}} = 60,4 \text{ А.}$, значить площа перерізу кабеля вибрана вірно. Визначаємо потужність трансформатора:

$$N_{\text{тр}} = ((K_{\text{п}} * P_{\text{ном}})/\cos\phi) \quad (8.10)$$

$$N_{\text{тр}} = (0,7 * 53)/0,8 = 46,375 \text{ кВт*А}$$

$P_{\text{ном}}$ – сумарна потужність електроприймачів, кВт;

$\cos\phi$ – середній коефіцієнт потужності електроприймачів (0,8);

$K_{\text{п}}$ – коефіцієнт попиту (0,7);

Одержане значення потужності трансформатора округляємо до ближчого стандартного значення. Визначаємо опір трансформатора Z_{T} . Вибираємо трансформатор на 40 кВА ($Z_{\text{T}} = 0,562 \text{ Ом}$). визначаємо орієнтовно площу перерізу провідника. Для магістрального кабеля:

$$S_{\text{н1}} \geq 0,5 * S_{\phi} \quad (8.11)$$

$$S_{\text{н1}} \geq 0,5 * S_{\phi} = 0,5 * 70 = 35 \text{ мм}^2$$

Визначаємо для розгалуження:

					БКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		109

$$S_{n2} \geq 0,5 \cdot 10 = 5 \text{ мм}^2$$

Округляємо ці значення до найближчих більших 35 мм^2 (S_{n1}), і 6 мм^2 (S_{n2}). Визначаємо активний і індуктивний опір фазного і нульового захисного провідників на ділянках 1 і 2:

$$R_{\phi} = \rho * (L_{\phi}/S_{\phi 1}) + \rho * (L/S_{\phi 2}) \quad (8.12)$$

$$R_{\phi} = 0,028 * (70/70) + 0,028 * (22/10) = 0,0896 \text{ Ом}$$

$$R_n = \rho * (L_n/S_{n1}) + \rho * (L/S_{n2}) \quad (8.13)$$

$$R_n = 0,028 * (70/35) + 0,028 * (22/6) = 0,1586 \text{ Ом}$$

Для окремо проложених нульових провідників його приймають рівним $0,6 \text{ Ом/км}$. При прокладці кабелем, або в сталевих трубах індуктивним опором нехтують.

Знаходимо дійсне значення (модуль) струма однофазного короткого замикання:

$$I_{кр} = U_{\phi} / ((Z_T/3) + \sqrt{(R_{\phi} + R_n)^2 + (X_{\phi} + X_n + X'_n)^2}) \quad (8.14)$$

$$I_{кр} = 220 / ((0,562/3) + \sqrt{(0,0896 + 0,1586)^2}) = 418,18 \text{ А}$$

Визначення максимальної напруги на корпусі обладнання відносно землі при замиканні фази на корпус.

$$U_{ктах} = I_{кр} * Z_n \quad (8.15)$$

					БКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		110

$$U_{kmax} = 418,18 * 0,1586 = 66,32 \text{ В}$$

Z_H – повний опір нульового провідника.

Умова не виконується. Необхідно збільшити перерізи Sn 1 та Sn 2 до Sф1 та Sф2 і зробити перерахунок:

$$R_n = 0,028 * (70/70) + 0,028 * (22/10) = 0,0896 \text{ Ом},$$

$$I_{кр} = 220 / ((0,562/3) + \sqrt{(0,0896 + 0,0896)^2}) = 600,21 \text{ А},$$

$$U_{kmax} = 600,21 * 0,0896 = 53,77 \text{ В}.$$

Умова не виконується, необхідно або замінити запобіжник з плавкою вставкою на автоматичний вимикач із струмовим реле, що дає можливість зменшити час замикання на корпус і підвищити допустиму напругу на корпусі або застосувати повторне заземлення нульового захисного провідника. Повторне заземлення нульового захисного провідника:

$$R_n = (U_{доп} * R_o) / ((I_{кр} * Z_H) - U_{доп}) \quad (8.15)$$

$$R_n = 36 * 4 / ((600,21 * 0,0896) - 36) = 8,09 \text{ Ом}.$$

8.4 Висновки

Існує багато факторів, які можуть вплинути на роботу розробника, через некомфортні або навіть небезпечні умови праці, які знаходяться в приміщенні. Серед основних причин виділяється: недостатній рівень світла, гучний шум, високий рівень навантаження, умови мікроклімату. Під час дослідження теми, були переглянуті можливі шкідливі та небезпечні ситуації, які можуть виникнути на робочому місці та способи їх уникнення та ліквідації.

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		111

В результаті можна зробити висновки, які умови повинні бути для продуктивної роботи працівника, як повинно бути організоване приміщення і його робоче місце. За допомогою проведених обчислень, можна становити необхідні для комфортної роботи умови.

КБПЗ - 2023

					БКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		112

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання магістерської роботи, призначено для системи **управління розумним будинком з підсистемою безпеки передачі даних** та захистом від кібератак.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У магістерській роботі наведені теоретичне узагальнення й рішення наукового завдання дослідження методів управління розумним будинком з захистом від кібератак.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем **управління розумним будинком з підсистемою безпеки передачі даних**.
- Досліджена система управління розумним будинком з впровадженням підсистеми безпеки передачі даних для захисту від кібератак.
- На основі отриманих результатів досліджень створена програмна реалізація системи управління розумним будинком.

Розроблені під час виконання магістерської роботи алгоритми дозволяють успішно вирішувати завдання управління системою «Розумний дім».

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудовано алгоритм і обрано середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня C++ та JS. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи **Windows XP/Vista/7/8/10/11**

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати змішану систему шифрування що використовує Base64, AES, RSA.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання.

Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 1226 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить **1,5** роки.

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		114

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мова програмування JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.javascript.ru/>
2. Introduction to Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.dev/>
3. Серверне програмування [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Learn/Server-side>
4. Node.js v14.0.0 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/api/>
5. М. Кантелон , М. Хартер, Т. Головайчук, Н. Райлих // “Node.js в действии”.
6. Янг А., Мек Б., Кантелон М. // “Node.js в действии. 2-е издание”.
7. John Resig, Bear Bibeault, Josip Maras // “Secrets of theJavaScript Ninja”.
8. Express [Електронний ресурс] – Режим доступу до ресурсу: <http://expressjs.com/>
9. Фреймворк AngularJS [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/AngularJS>
10. AngularJS [Електронний ресурс] – Режим доступу до ресурсу: <https://angularjs.org/>
11. Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/>
12. React.js [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.reactjs.org/>
13. AngularJS MVC [Електронний ресурс] – Режим доступу до ресурсу: https://www.tutorialspoint.com/angularjs/angularjs_mvc_architecture.htm
14. Vue.js [Електронний ресурс] – Режим доступу до ресурсу: <https://vuejs.org/>

15. Angular 2 [Електронний ресурс] – Режим доступу до ресурсу: <https://angular.io/>
16. Односторінковий застосунок [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Односторінковий_застосунок
17. Что такое MVC [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.hexlet.io/blog/posts/что-такое-mvc-rasskazyvaem-prostymi-slovami>
18. Build Node.js Apps [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/docs/nodejs/nodejs-tutorial>
19. REST [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/REST>
20. What is REST [Електронний ресурс] – Режим доступу до ресурсу: <https://restfulapi.net/>
21. Тренды веб-разработки [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/451572/>
22. Веб приложение [Електронний ресурс] – Режим доступу до ресурсу: <https://webcase.com.ua/blog/cho-takoe-web-prilozhenie-vse-vidy/>
23. Мова розмітки гіпертексту [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTML>
24. HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/uk/docs/Web/HTML>
25. Каскадні таблиці стилів [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/CSS>
26. Стек MEAN [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/MEAN_\(веброзробка\)](https://uk.wikipedia.org/wiki/MEAN_(веброзробка))
27. MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/MongoDB>
28. Веб-технології для розробників [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/uk/docs/Web>

29. CORS, XSS [Електронний ресурс] – Режим доступу до ресурсу:
<https://dev.to/maleta/cors-xss-and-csrf-with-examples-in-10-minutes-35k3>
30. AJAX [Електронний ресурс] – Режим доступу до ресурсу:
<https://uk.wikipedia.org/wiki/AJAX>
31. Fetch API [Електронний ресурс] – Режим доступу до ресурсу:
https://developer.mozilla.org/ru/docs/Web/API/Fetch_API
32. AngularJS Tutorial [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.w3schools.com/angular/default.asp>
33. jQuery [Електронний ресурс] – Режим доступу до ресурсу:
<https://uk.wikipedia.org/wiki/JQuery>
34. Основи Web-технологій [Електронний ресурс] – Режим доступу до ресурсу:
https://pidruchniki.com/1243020547796/informatika/web-tehnologiyi_pidpriyemstvah
35. npm [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.npmjs.com/>
36. Install MongoDB [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.mongodb.com/manual/administration/install-on-linux/>
37. Как установить Node.js [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.digitalocean.com/community/tutorials/node-js-ubuntu-18-04-ru>
38. Linux [Електронний ресурс] – Режим доступу до ресурсу:
<https://en.wikipedia.org/wiki/Linux>
39. npm-audit [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.npmjs.com/cli/audit>
40. TypeScript [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.typescriptlang.org/>
41. SQL [Електронний ресурс] – Режим доступу до ресурсу:
<https://uk.wikipedia.org/wiki/SQL>
42. MongoDB Compass [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.mongodb.com/products/compass>

43. Postman [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postman.com/>

44. SQL [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/SQL>

45. SPA (Single-page application) [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Glossary/SPA>

46. Охорона праці [Електронний ресурс]: реферат – Режим доступу до ресурсу: https://revolution.allbest.ru/life/00468031_0.html

47. Природне і штучне освітлення ДБН В.2.5-28:2018: державні будівельні норми України [Електронний ресурс] / Ю. Громадський, С. Облакевич, М.Громадський, Г. Фаренюк, Є. Фаренюк, О. Підгорний, О. Сергейчук, Є.Рейцен, В. Єгорченков, Л. Коваль, Д. Радомцев, В. Злоба, Н. Кучеренко, Г.Кожушко, О. Гончар, О. Козенко, Б. Шабашкевіч, Ю. Добровольський, В.Акіменко, С. Гозак, А. Яригін, В. Назаренко, В. Мартиросова, В. Сорокін, Є.Пугачов - Київ 2018 - Режим доступу до ресурсу: https://ledeffect.com.ua/images/_branding/dbn2018.pdf

48. Розрахунок світлодіодного освітлення кімнати [Електронний ресурс] – Режим доступу до ресурсу: <https://luxled.biz.ua/rozrahynok-svitlodoidnogo-osvitlennja-kimnatu-v-kvarturi-abo-bydunky>

49. Охорона праці, охорона праці та безпека в надзвичайних ситуаціях : метод. вказ. до викон. розділів у дипломних роботах / [укл. В.М. Челябієва, О.Л. Гуменюк] - Чернігів ЧДТУ 2013 - [Електронний ресурс] – Режим доступу до ресурсу: http://ir.stu.cn.ua/bitstream/handle/123456789/12461/Охорона_праці_та_безпека._в_надзв._ситуац;метод.вказ..pdf?sequence=1&isAllowed=y

50. Освітленість робочих місць [Електронний ресурс] – Режим доступу до ресурсу:https://ua-referat.com/Освітленість_робочих_місць_сучасні_підходи_до_вимірів_і_оцінки

51. Температурний режим праці [Електронний ресурс] – Режим доступу до ресурсу: <http://poltava.medprof.org.ua/poltava/zakhist-trudovikh-ta-socialno->

ekonomichnikh-prav-pracivnikiv-galuzi/pravova-dopomoga/temperaturnii-rezhim-praci-jakim-vin-maje-butii/

52. Шум. Методи захисту від його дії : метод. вказ. до лабораторної роботи / [укл. В. І. Шмирко, С. М. Журавель] — Запоріжжя: ЗНТУ, 2014. - 14 с. [Електронний ресурс] – Режим доступу до ресурсу: https://zp.edu.ua/sites/default/files/konf/ooop_shum-2014.pdf

53. Системи протипожежного захисту ДБН В.2.5-56:2014 / Б. Платкевич, В. Носач, В. Федюк, В. Мусійчук, В. Євстіфєєв, Г. Дубінський, В. Сокол, А.Бушиленко, В. Дунюшкін, Р. Уханський, С. Пономарьов, В. Приймаченко, А.Приймаченко, С. Пітайчук, Н. Морозова, І. Колосов, О. Лагода, П. Мізін, В.Савченко, М. Федорович, П. Шаповалов, Л. Фесенко — Київ 2015 — [Електронний ресурс] – Режим доступу до ресурсу: <http://kbu.org.ua/assets/app/documents/dbn2/98.1>. ДБН В.2.5-56~2014. Системи протипожежного захисту.pdf

54. Охорона праці. Ч. 2. Занулення : метод. вказ. до викон. розрахунків з викор. персон. ЕОМ IBM–сумісного типу / [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака, А. Е. Солових, С. Е. Катеринич] ; Центральноукраїн. нац. техн. ун-т. - 2–ге вид., перероб. та доп. - Кропивницький : ЦНТУ, 2019. - 27 с.[Електронний ресурс] – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/8769>

55. The Top JavaScript Frameworks [Електронний ресурс] – Режим доступу до ресурсу: <https://www.freecodecamp.org/news/complete-guide-for-front-end-developers-javascript-frameworks-2019/>

56. JavaScript Frameworks 2020 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.npmjs.com/package/migrate-mongo>

57. Web Technology [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/web-technology/>

					ВКРМ-122.23.0018.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Лат		119

К6П3 - 2023

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-122.23.0018.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Повзун В.А.				Дослідження та програмна реалізація системи управління розумним будинком з впровадженням системи захисту від кібератак	Літ.	Аркуш	Аркушів
Перевірів	Босько В.В.					Б	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-22М			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи управління розумним будинком з впровадженням системи захисту від кібератак.

2 Підстава для розробки

Підставою для розробки служить завдання на магістерську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 32-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою магістерської роботи є дослідження та програмна реалізація системи управління розумним будинком з підсистемою безпеки передачі даних для захисту від кібератак

4 Джерела розробки

Джерелом цієї магістерської роботи є відносна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

					ВКРМ-122.23.0018.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- Розробку додатку;
- систему підключення та тестування баз даних ;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-122.23.0018.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мови програмування

Середовище C++/ JS /PHP 7.1 СУБД MySQL та PostgreSQL.

					БКРМ-122.23.0018.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

8 Вимоги щодо охорони праці

В частині охорони праці та техніки безпеки в магістерській роботі повинен бути розглянутий аналіз умов праці програміста та розрахунок штучного освітлення.

					ВКРМ-122.23.0018.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 119 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі бакалаврської дипломної роботи.
Постановка задачі на виконання бакалаврської дипломної роботи (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень бакалаврської дипломної роботи.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

11.1 Подання магістерської дипломної роботи на попередній захист
10.12.2023 р.

1.2 Подання магістерської роботи на захист 20.12.2023 р.

					ВКРМ-122.23.0018.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник випускної кваліфікаційної роботи
за другим (магістерським) рівнем вищої освіти
_____ Босько В.В.

*Дослідження та програмна реалізація системи управління розумним
будинком з впровадженням системи захисту від кібератак*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 22

Літера: РП

Кропивницький – 2023 року

Лістинг 1. Програмний код створення таблиць бази даних

```

Create database exchange,
Use exchange,
CREATE TABLE `message` (
`id` INT(5) NOT NULL AUTO_INCREMENT,
`to_mes` INT(5) NOT NULL,
`from_mes` INT(5) NOT NULL,
`text` VARCHAR(50) NOT NULL,
`status` INT(11) NOT NULL,
`data` VARCHAR(50) NULL DEFAULT NULL,
`delete_to` INT(10) NOT NULL,
`delete_from` INT(10) NOT NULL,
PRIMARY KEY (`id`)
)ENGINE=InnoDB;
CREATE TABLE `register` (
`id` INT(5) NOT NULL AUTO_INCREMENT,
`username` VARCHAR(40) NOT NULL,
`password` VARCHAR(40) NOT NULL,
`status` INT(5) NOT NULL,
PRIMARY KEY (`id`)
)
ENGINE=InnoDB;

```

add_user.php

```

<?require_once('function.php');
$name=$_POST['name'];
$password=$_POST['password'];
session_start();
if(!$name || !$password){echo "Не введені обов'язкові поля.";exit;}
if(strlen($name)>16){echo "Імя містить забагато символів. Мак 16.";exit;}
if(strlen($password)>16){echo "Пароль містить забагато символів. Мак 16.";exit;}
if(strlen($password)<6){echo "Пароль містить недостатньо символів. Мин 6.";exit;}register($name,$password);
$_SESSION['valid_user']=$name;do_html_header();
echo "Ви успішно зареєстровані у ситемі.Перейдіть на сторінку для зареєстрованих користувачів.";
do_html_url('member.php','Перейти.');
```

Файл Main.exe – програмування мікроконтролера для системи управління

```

R7. EQU 7; адреса регистров R0-R7
R6. EQU 6
R5. EQU 5
R4. EQU 4
R3. EQU 3
R2. EQU 2
R1. EQU 1
R0. EQU 0
ACC. EQU0E0H; адрес аккумулятора
B. EQU0F0H; адрес регистра B
PSW. EQU000H; адрес регистра (слова) состояния
SP. EQU81H; адрес указателя стека
DPL. EQU82H; адрес младшей половины DPTR
DPH. EQU83H; адрес старшей половины DPTR
P0. EQU80H; адрес регистра порта P0

```

P1. EQU90H; адрес регистра порта P1
 P2. EQU0A0H; адрес регистра порта P2
 P3. EQU0B0H; адрес регистра порта P3
 В.0. EQU0F0H; адреса отдельных битов регистра В
 В.1. EQU0F1H
 В.2. EQU0F2H
 В.3. EQU0F3H
 В.4. EQU0F4H
 В.5. EQU0F5H
 В.6. EQU0F6H
 В.7. EQU0F7H
 АСС.0. EQU0E0H; адреса отдельных битов аккумулятора
 АСС.1. EQU0E1H
 АСС.2. EQU0E2H
 АСС.3. EQU0E3H
 АСС.4. EQU0E4H
 АСС.5. EQU0E5H
 АСС.6. EQU0E6H
 АСС.7. EQU0E7H
 PSW.0. EQU0D0H; адреса отдельных битов регистра PSW
 PSW.1. EQU0D1H
 PSW.2. EQU0D2H
 PSW.3. EQU0D3H
 PSW.4. EQU0D4H
 PSW.5. EQU0D5H
 PSW.6. EQU0D6H
 PSW.7. EQU0D7H
 P0.0. EQU080H; адреса отдельных линий порта P0
 P0.1. EQU081H
 P0.2. EQU082H
 P0.3. EQU083H
 P0.4. EQU084H
 P0.5. EQU085H
 P0.6. EQU086H
 P0.7. EQU087H
 P1.0. EQU090H; адреса отдельных линий порта P1
 P1.1 EQU091H
 P1.2 EQU092H
 P1.3 EQU093H
 P1.4 EQU094H
 P1.5 EQU095H
 P1.6. EQU096H
 P1.7 EQU097H
 P2.0. EQU0A0H; адреса отдельных линий порта P2
 P2.1 EQU0A1H
 P2.2 EQU0A2H
 P2.3 EQU0A3H
 P2.4 EQU0A4H
 P2.5 EQU0A5H
 P2.6 EQU0A6H
 P2.7 EQU0A7H
 P3.0. EQU0B0H; адреса отдельных линий порта P3
 P3.1 EQU0B1H
 P3.2 EQU0B2H
 P3.3 EQU0B3H
 P3.4 EQU0B4H
 P3.5 EQU0B5H
 P3.6 EQU0B6H
 P3.7 EQU0B7H
 ;
 CS. EQU P3.7
 DCLOCK. EQU P3.6
 DOUT. EQU P3.5
 ;
 .org 0; нижеследующая команда с адреса 0
 ;
 LJM PSTART; на команду после метки START
 ;
 .org 100H; нижеследующая команда с адреса 100H

```

;
START:
MOV P0, #11111111B; начальная установка
MOV P1, #11111111B
MOV P2, #11111111B
MOV P3, #11111111B
CLRDCLOCK; установка DCLOCK в 0
;
L7816; собственно чтение
;
CLRCS; импульс старта преобразования
;
SETBDCLOCK; 1-й тактовый импульс
CLRDCLOCK
;
SETBDCLOCK; 2-й тактовый импульс
CLRDCLOCK
;
SETBDCLOCK; 3-й тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV B.3. C; DB11 В В.3
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV B.2. C; DB10 В В.2
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV B.1. C; DB9 В В.1
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV B.0. C; DB8 В В.0
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.7. C; DB7 В ACC.7
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.6. C; DB6 В ACC.6
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.5. C; DB5 В ACC.5
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.4. C; DB4 В ACC.4
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.3. C; DB3 В ACC.3
SETBDCLOCK; тактовый импульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.2. C; DB2 В ACC.2
SETBDCLOCK; тактовый импульс

```

```

CLRDCLOCK
;
MOVC, DOUT
MOV ACC.1. C; DB1 В ACC.1
SETBDCLOCK; тактовий імпульс
CLRDCLOCK
;
MOVC, DOUT
MOV ACC.0. C; DB0 В ACC.0
;
SETBCS; завершення считывання
;
MOV R4, A; сохр. мл. и ср. тетрады в R4
;
MOVA, B; читаем из регистра в ст. тетраду
ANL A, #00001111B; зануляем старшие 4 бита
MOV R5, A; в R5R4 - результат
;
SJMP L7816; зацикливание
;
. END

```

Файл member.php що відповідає за роботу профільної частини користувача.

```

<?
require_once('function.php');
session_start();
@$username=$_POST['username'];
@$password=$_POST['password'];
if($username && $password)
{
try {login($username,$password);$_SESSION['valid_user']=$username;
} catch(Exception $e)
{do_html_header();
echo "Вхід неможливий невірний логін та пароль.";
do_html_url('login.php','Вхід');
do_html_footer();
exit;}}//далі має бути система спілкування.
$user=user_id($_SESSION["valid_user"]);
if(reg_user()){
do_html_header();
?><div id="content">
<div id="header"></div>
<div id="vnesh">
<div id="vnutr">
<div id="left">
<div id=user></div>
</div>
<div id="right">
<div id=menu>
<p style="display:none;" id=my_user><?echo $user;?></p>
</div>
</div>
<div id="center">
<div id=alluser>
<span>Написати повідомлення:</span><br>
<?
output_users_to_messages($_SESSION["valid_user"]);
?>
</div>
</div>
<div id=podporka></div>
</div>

```

```

</div>
<div id="footer"></div>
</div>
<?
do_html_footer();
}
?>

```

Файл data_valid.php який містить в собі головні функції роботи веб ужитку.

```

<?
function db_connect()
{
@ $result=mysql_pconnect('localhost','user','password');
//mysql_query ("set character_set_client='cp1251'");
//mysql_query ("set character_set_results='cp1251'");
//mysql_query ("set collation_connection='cp1251_general_ci'");
if(!$result){echo "Не вдалось з'єднатись з бази даних";exit;}
}
function register($name,$password)
{db_connect();
mysql_select_db("user");
$zaput="select * from register where username='$name' " ;
$result=mysql_query($zaput);
if(mysql_num_rows($result)>0){echo "Немає даних";exit; }
$zaput="insert into register (username,password,status)
values ('".$name."','".$password."','".$0')";
$result=mysql_query($zaput);
if(!$result){echo"Невдалий результат.";}
}
function login($username, $password)
{
db_connect();mysql_select_db("user");
$zaput="select * from register where username='$username' and
password='$password'";
$result=mysql_query($zaput);
if(!$result)throw new Exception('Помилка');
if(mysql_num_rows($result)>0)return ;
else throw new Exception(' Помилка');
};function reg_user()
{global $valid_user;
if(isset($_SESSION['valid_user']))
{echo '-'.($_SESSION['valid_user']).'.';
$status=1;
$username=$_SESSION['valid_user'];
status_user($status,$username);
?>
<a href="logout.php">Вийти<a/><br><hr><?return true;
}else
{do_html_heading();echo "Поверніться та введіть обовязкові
поля.";do_html_url('login.php','Вхід');do_html_footer();return
false;}}function status_user($status,$username){
db_connect();mysql_select_db("user");$zaput="update register set
status='$status' where username='$username'";mysql_query($zaput);}
function user_id($username){db_connect();
mysql_select_db("user");
$zaput="select id from register where username='$username'";
$result=mysql_query($zaput);
$row=mysql_fetch_assoc($result);
return $row['id'];
}function user_log($id)
{db_connect();
mysql_select_db("user");

```

```

$zaput="select      username      from      register      where      id='$id'";
$result=mysql_query($zaput);$row=mysql_fetch_assoc($result);return
$row['username'];}
function output_users_to_messages($log_user)
{db_connect();
mysql_select_db("user");
$zaput="select      username,id,status      from      register      where      username
!='$log_user' " ;
$result=mysql_query($zaput);
while ($row = mysql_fetch_assoc($result)) {
$user=$row['id'];
$user2=user_id($log_user);
$zaput2="select id from message where status=0 and to_mes='$user2' and
from_mes='$user' ";
$result2=mysql_query($zaput2);
$i=0;while ($row2 = mysql_fetch_assoc($result2))
{
$i=$i+1;
}
if($row['status']==1){
?><font color=blue>
<?
echo "Online ";
?>
</font>
<?
}else{echo "Offline ";}
echo"<a href='send.php?to=".$row["id"]."'>".$row['username']. "</a>";
if($i==0){echo " 0";echo "<br>";}else{echo"+".$i;echo "<br>";}}
function send_mess($from,$to,$message,$id_rewrite_message)
{db_connect();
mysql_select_db("user");
$data=date('m.d.y');
if($id_rewrite_message==false){
$zaput="insert      into      message      (to_mes,from_mes,text,status,data,
delete_to,delete_from)      values
('".$to."','".$from."','".$message."','".$0."','".$ data."','".$0','0')";
$result=mysql_query($zaput);
$zaput1="update      message      set      status=1      where      to_mes='$from'      and
from_mes='$to'";
$result=mysql_query($zaput1);
}else {$zaput="update      message      set      text='$message',      status=0      where
id='$id_rewrite_message'";
$result=mysql_query($zaput);}}?>

```

Файл add_user.php слугує додаванням нового користувача у базу даних.

```

<?
require_once('function.php');
$name=$_POST['name'];
$password=$_POST['password'];
session_start();
if(!$name || !$password){echo "Не введені обов'язкові поля.";exit;}
if(strlen($name)>16){echo "Імя містить забагато символів. Мак 16.";exit;}
if(strlen($password)>16){echo "Пароль містить забагато символів. Мак 16.";exit;}
if(strlen($password)<6){echo "Пароль містить недостатньо символів. Мін 6.";exit;}
register($name,$password);
$_SESSION['valid_user']=$name;
do_html_header();
echo "Ви успішно зареєстровані у ситемі.Перейдіть на сторінку для зареєстрованих користувачів.";

```

```
do_html_url('member.php','Перейти.');
```

```
do_html_footer();
```

```
?>
```

body1.js

```
$(document).ready(function(){
  var header = $('header').height();
  var footer = $('footer').height();
  var body1 = $('.body1');
  body1.css('min-height', ($(window).height() - (header + footer)) - 20);
  $(window).resize(function(){
    var header = $('header').height();
    var footer = $('footer').height();
    var body1 = $('.body1');
    body1.css('min-height', ($(window).height() - (header + footer)) - 20);
  });
});
```

main.css

```
:root {
  --main-color: rgb(20, 79, 150);
  --buttons-color: rgb(66, 238, 72);
}
body {
  margin: 0;
  padding: 0;
  background: #F1F1F1;
}
header {
  background: var(--main-color);
  height: 70px auto;
}
footer {
  width: 100%;
  background: black;
  height: 50px;
}
footer div {
  display: inline-block;
}
#f1 {
  margin-top: 10px;
}
#f2 {
  float: right;
}
.headerSiteTitle {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
  color: white;
  display: inline;
  margin: 10px;
}
.headerSiteTitle a {
  text-decoration: none;
  color: white;
}
#siteTitle {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
```

```
    color: var(--main-color);
    display: inline;
    font-size: 150px;
}
#subTitle {
    margin: 0;
    padding: 0;
    font-family: sans-serif;
    color: var(--main-color);
    display: inline;
    font-size: 50px;
}
#headerButtonsBlock {
    margin: 20px;
    display: inline;
    position: absolute;
    right: 0;
}
#headerButtonsBlock a{
    margin: 5px;
}
#mainInfo {
    margin: 10px;
    background: white;
}

#servicesTypes {
    padding: 10px;
}
#pages {
    padding-bottom: 10px;
}

#adminB {
    display: block;
    margin: 5px;
}

.body1 {
    background: white;
    /* margin: 10px; */
    margin-top: 0px;
    padding: 10px;
}
.services {
    background: white;
    margin: 10px;
    padding: 10px;
}
.button {
    background: white;
    text-decoration: none;
    display: inline-block;
    font-family: sans-serif;
    color: var(--main-color);
}
.button2 {
    background: red;
    width: 150px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-family: sans-serif;
    color: white
}
.button div{
    margin: 10px;
}
```

```
.button2 div{
  margin: 10px;
}
.page {
  color: white;
  text-decoration: none;
  font-family: sans-serif;
  margin: 10px;
  padding: 15px;
  /* background: green; */
}
.pagef {
  color: white;
  text-decoration: none;
  font-family: sans-serif;
  margin: 10px;
}
.serviceList {
  margin: 10;
  padding: 10;
  background: white;
  display: inline-block;
}
.serviceItem {
  margin: 10px;
  padding: 10px;
  /* width: 250px; */
  height: 100px;
  background: var(--main-color);
  color: white;
  font-family: sans-serif;
  display: inline-block;
  box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.serviceItem span {
  vertical-align: top;
  margin-left: 10px;
}
.loginForm {
  padding: 10px;
}
.loginForm input {
  border-style: none;
  text-align: center;
}
.loginForm form {
  padding: 10px;
  width: 200px;
  background: var(--main-color);
  box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.signupForm form {
  padding: 10px;
  width: 300px;
  background: var(--main-color);
  box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.signupForm input {
  border-style: none;
  text-align: center;
}
.button3 {
  background: var(--main-color);
}
```

```
color: white;
text-decoration: none;
font-family: sans-serif;
padding: 10px;
box-shadow: 0 0 10px rgba(0, 0, 0, 10);
margin: 20px;
}
.blogPrev::before {
display: inline-block;
}
.blogPrev {
background: var(--main-color);
text-align: center;
color: white;
margin-left: 0px;
margin-right: 5px;
margin-bottom: 10px;
padding: 10px;
font-family: sans-serif;
display: inline-block;
box-shadow: 0 0 10px rgba(0, 0, 0, 10);
text-decoration: none;
}
.blogPrev span {
text-decoration: none;
}
.publicationText {
background: var(--main-color);
font-family: sans-serif;
margin: 5px;
padding: 5px;
color: white;
box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.publicationText img {
position: absolute;
margin-top: 0;
}
.addComment {
margin: 10px;
padding: 10px;
background: var(--main-color);
color: white;
width: 700px;
font-family: sans-serif;
box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.addComment h3 {
margin-left: 0;
margin-right: 0;
margin-top: 0;
margin-bottom: 10;
}
.comment {
margin: 10px;
padding: 10px;
background: var(--main-color);
color: white;
width: 700px;
font-family: sans-serif;
box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.comment * {
display: inline;
```

```

}
.comment img {
    width: 50px;
    height: 50px;
}
.delete {
    padding: 5px;
    background: red;
    font-family: sans-serif;
    color: white;
    text-decoration: none;
    float: right;
}
}
.currentUserInfo {
    color: white;
    text-align: left;
    font-family: sans-serif;
    background-color: var(--main-color);
    display: inline-block;
    padding: 10px;
    font-size: 11px;
}
}
.myButton {
    background: var(--buttons-color);
    padding: 10px;
    border-radius: 20px;
    color: white;
    text-decoration: none;
    font-family: sans-serif;
    border-style: none;
}
}

```

pubPage.php

```

<?php
    require_once 'setting.php';
    $id = $_GET['id'];
    $_pub = mysqli_query($db, 'SELECT * FROM `publics` WHERE `id` = ' . $id);
    $pub = mysqli_fetch_assoc($_pub);
?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <title><?php echo $pub['Title']; ?></title>
        <link rel="stylesheet" href="main.css">
        <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
        <script src="body1.js" charset="utf-8"></script>
    </head>
    <body>
        <?php include 'header.php'; ?>
        <div class="body1">
            <div class="publicationText">
                <h1><?php echo $pub['Title']; ?></h1>
                <p>
                    <?php echo $pub['Text']; ?>
                </p>
                <span><?php echo $pub['PubDate']; ?></span>
            </div>
            <?php
                $comments = mysqli_query($db, 'SELECT * FROM `comments` WHERE `pubId`
                = ' . $id);

                while ($comment = mysqli_fetch_assoc($comments)) {

```

```

        $_user = mysqli_query($db, 'SELECT * FROM `users` WHERE `id` = ' .
$comment['UserId']);
        $user = mysqli_fetch_assoc($_user);
        ?>
        <div class="comment">
            <span>
                <img align="left" hspace="5" src=<?php echo $user['ava']; ?>
alt="">
                <?php
                    echo $user['FirstName'] . ' ' . $user['Name'] . ' ' .
$user['LastName'];
                ?>
            </span>
            <br>
            <br>
            <p style="margin-left: 60px; margin-top: 10px; text-align:
center;">
                <?php
                    echo $comment['commentText'];
                ?>
            </p>
            <br>
            <span>
                <?php
                    echo $comment['date'];
                ?>
            </span>
            <?php if (($SESSION['currentUser']['isAdmin'])): ?>
            <a class = "delete" href=
            <?php
                echo 'pubPage.php?removeCommentId=' . $comment['id'] .
'&id=' . $id;
            ?>
            >Видайти</a>
            <?php endif; ?>
        </div>
        <?php
    }
    ?>
    <?php if (isset($SESSION['currentUser'])): ?>
    <form class="addComment" action=<?php echo 'addComment.php?id=' . $id;
?> method="post">
        <h3>Коментувати</h3>
        <textarea name="commentText" rows="8" cols="80"></textarea>
        <input class="myButton" type="submit" name="submit" value="Надіслати">
    </form>
    <?php endif; ?>
</div>

    <?php include 'footer.php'; ?>
</body>
</html>

```

```

namespace RecoveryDisk
{
    partial class MainForm
    {
        /// <summary>
        ///
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        ///
        /// </summary>
        /// <param name="disposing">правда, якщо керуючі ресурси повині бути
розташовані, неправда в іншому випадку.</param>
        protected override void Dispose(bool disposing)
        {

```

```

        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Викликаємо метод для підтримки інтерфейсу
    /// </summary>
    private void InitializeComponent()
    {
        this.components = new System.ComponentModel.Container();
        System.Windows.Forms.TreeNode treeNode1 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode2 = new
System.Windows.Forms.TreeNode("Documents and Settings", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode1});
        System.Windows.Forms.TreeNode treeNode3 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode4 = new
System.Windows.Forms.TreeNode("Downloads", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode3});
        System.Windows.Forms.TreeNode treeNode5 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode6 = new
System.Windows.Forms.TreeNode("Program Files", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode5});
        System.Windows.Forms.TreeNode treeNode7 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode8 = new
System.Windows.Forms.TreeNode("RapidDriver", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode7});
        System.Windows.Forms.TreeNode treeNode9 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode10 = new
System.Windows.Forms.TreeNode("Server", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode9});
        System.Windows.Forms.TreeNode treeNode11 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode12 = new
System.Windows.Forms.TreeNode("WINDOWS", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode11});
        System.Windows.Forms.TreeNode treeNode13 = new
System.Windows.Forms.TreeNode("");
        System.Windows.Forms.TreeNode treeNode14 = new
System.Windows.Forms.TreeNode("WINDOWS.0", 18, 20, new
System.Windows.Forms.TreeNode[] {
            treeNode13});
        System.Windows.Forms.TreeNode treeNode15 = new
System.Windows.Forms.TreeNode("SYSTEM2 (C:)", 23, 24, new
System.Windows.Forms.TreeNode[] {
            treeNode2,
            treeNode4,
            treeNode6,
            treeNode8,
            treeNode10,
            treeNode12,
            treeNode14});
        System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(MainForm));
    }

```

```

        this.menuStrip = new System.Windows.Forms.MenuStrip();
        this.fileToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.instrumentToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.adjustmentToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.encodingfilterToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.quickextractionToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.helpToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.separatorToolStripMenuItem = new
System.Windows.Forms.ToolStripItemSeparator();
        this.aboutToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.coderConfigGroupBox = new System.Windows.Forms.GroupBox();
        this.redundancyGroupBox = new System.Windows.Forms.GroupBox();
        this.redundancyMacTrackBar = new
EConTech.Windows.MACUI.MACTrackBar();
        this.allVolCountGroupBox = new System.Windows.Forms.GroupBox();
        this.allVolCountMacTrackBar = new
EConTech.Windows.MACUI.MACTrackBar();
        this.toolTip = new System.Windows.Forms.ToolTip(this.components);
        this.browser = new FileBrowser.Browser();
        this.repairButton = new System.Windows.Forms.Button();
        this.testButton = new System.Windows.Forms.Button();
        this.recoverButton = new System.Windows.Forms.Button();
        this.protectButton = new System.Windows.Forms.Button();
        this.exitToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.testspeedToolStripMenuItem = new
System.Windows.Forms.ToolStripItem();
        this.menuStrip.SuspendLayout();
        this.coderConfigGroupBox.SuspendLayout();
        this.redundancyGroupBox.SuspendLayout();
        this.allVolCountGroupBox.SuspendLayout();
        this.SuspendLayout();
        //
        // menuStrip
        //
        this.menuStrip.Items.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.fileToolStripMenuItem,
        this.instrumentToolStripMenuItem,
        this.adjustmentToolStripMenuItem,
        this.helpToolStripMenuItem});
        this.menuStrip.LayoutStyle =
System.Windows.Forms.ToolStripLayoutStyle.Flow;
        this.menuStrip.Location = new System.Drawing.Point(0, 0);
        this.menuStrip.Name = "menuStrip";
        this.menuStrip.Size = new System.Drawing.Size(986, 21);
        this.menuStrip.TabIndex = 0;
        this.menuStrip.Text = "menuStrip";
        //
        // fileToolStripMenuItem
        //
        this.fileToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.exitToolStripMenuItem});
        this.fileToolStripMenuItem.Name = "файлToolStripMenuItem";
        this.fileToolStripMenuItem.Size = new System.Drawing.Size(45, 17);
        this.fileToolStripMenuItem.Text = "Файл";
        //
        // instrumentsToolStripMenuItem
        //
        this.instrumentToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {

```

```

        this.testspeedToolStripMenuItem));
        this.instrumentToolStripMenuItem.Name =
"instrumentsToolStripMenuItem";
        this.instrumentToolStripMenuItem.Size = new System.Drawing.Size(82,
17);
        this.instrumentToolStripMenuItem.Text = "Інструменти";
        //
        // adjustmentToolStripMenuItem
        //
        this.adjustmentToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.encodingfilterToolStripMenuItem,
        this.quickextractionToolStripMenuItem});
        this.adjustmentToolStripMenuItem.Name =
"adjustmentToolStripMenuItem";
        this.adjustmentToolStripMenuItem.Size = new System.Drawing.Size(74,
17);
        this.adjustmentToolStripMenuItem.Text = "Параметри";
        //
        // encodingfilterToolStripMenuItem
        //
        this.encodingfilterToolStripMenuItem.ImageScaling =
System.Windows.Forms.ToolStripItemImageScaling.None;
        this.encodingfilterToolStripMenuItem.Name =
"encodingfilterToolStripMenuItem";
        this.encodingfilterToolStripMenuItem.Size = new
System.Drawing.Size(216, 22);
        this.encodingfilterToolStripMenuItem.Text = "Шифруючий фільтр";
        this.encodingfilterToolStripMenuItem.Click += new
System.EventHandler(this.encodingfilterToolStripMenuItem_Click);
        //
        // helpToolStripMenuItem
        //
        this.helpToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.separatorToolStripMenuItem,
        this.aboutToolStripMenuItem});
        this.helpToolStripMenuItem.Name = "helpToolStripMenuItem";
        this.helpToolStripMenuItem.Size = new System.Drawing.Size(60, 17);
        this.helpToolStripMenuItem.Text = "Довідка";
        //
        // separatorToolStripMenuItem
        //
        this.separatorToolStripMenuItem.Name = "separatorToolStripMenuItem";
        this.separatorToolStripMenuItem.Size = new System.Drawing.Size(163,
6);
        //
        // aboutToolStripMenuItem
        //
        this.aboutToolStripMenuItem.Name = "aboutToolStripMenuItem";
        this.aboutToolStripMenuItem.Size = new System.Drawing.Size(166, 22);
        this.aboutToolStripMenuItem.Text = "Про програму...";
        this.aboutToolStripMenuItem.Click += new
System.EventHandler(this.aboutToolStripMenuItem_Click);
        //
        // coderConfigGroupBox
        //
        this.coderConfigGroupBox.Controls.Add(this.redundancyGroupBox);
        this.coderConfigGroupBox.Controls.Add(this.allVolCountGroupBox);
        this.coderConfigGroupBox.FlatStyle =
System.Windows.Forms.FlatStyle.Flat;
        this.coderConfigGroupBox.Location = new System.Drawing.Point(414,
26);
        this.coderConfigGroupBox.Name = "coderConfigGroupBox";
        this.coderConfigGroupBox.Size = new System.Drawing.Size(561, 98);
        this.coderConfigGroupBox.TabIndex = 5;
        this.coderConfigGroupBox.TabStop = false;
        this.coderConfigGroupBox.Text = "Конфігурація системи";
        //

```

```

        // redundancyGroupBox
        //
        this.redundancyGroupBox.Controls.Add(this.redundancyMacTrackBar);
        this.redundancyGroupBox.FlatStyle =
System.Windows.Forms.FlatStyle.Flat;
        this.redundancyGroupBox.Location = new System.Drawing.Point(286,
21);

        this.redundancyGroupBox.Name = "redundancyGroupBox";
        this.redundancyGroupBox.Size = new System.Drawing.Size(264, 65);
        this.redundancyGroupBox.TabIndex = 4;
        this.redundancyGroupBox.TabStop = false;
        this.redundancyGroupBox.Text = "Надлишковість кодування";
        //
        // allVolCountGroupBox
        //
        this.allVolCountGroupBox.Controls.Add(this.allVolCountMacTrackBar);
        this.allVolCountGroupBox.FlatStyle =
System.Windows.Forms.FlatStyle.Flat;
        this.allVolCountGroupBox.Location = new System.Drawing.Point(12,
21);

        this.allVolCountGroupBox.Name = "allVolCountGroupBox";
        this.allVolCountGroupBox.Size = new System.Drawing.Size(264, 65);
        this.allVolCountGroupBox.TabIndex = 3;
        this.allVolCountGroupBox.TabStop = false;
        this.allVolCountGroupBox.Text = "Кількість дисків";
        //
        // toolTip
        //
        this.toolTip.AutomaticDelay = 2000;
        this.toolTip.AutoPopDelay = 20000;
        this.toolTip.InitialDelay = 2000;
        this.toolTip.ReshowDelay = 1000;
        //
        // redundancyMacTrackBar
        //
        this.redundancyMacTrackBar.BackColor =
System.Drawing.Color.Transparent;
        this.redundancyMacTrackBar.BorderColor =
System.Drawing.SystemColors.ActiveBorder;
        this.redundancyMacTrackBar.Font = new System.Drawing.Font("Verdana",
8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte) 0));
        this.redundancyMacTrackBar.ForeColor =
System.Drawing.Color.FromArgb(((int) ((byte) 123))), ((int) ((byte) 125))),
((int) ((byte) 123))));
        this.redundancyMacTrackBar.IndentHeight = 6;
        this.redundancyMacTrackBar.Location = new System.Drawing.Point(6,
24);
        this.redundancyMacTrackBar.Maximum = 199;
        this.redundancyMacTrackBar.Minimum = 0;
        this.redundancyMacTrackBar.Name = "redundancyMacTrackBar";
        this.redundancyMacTrackBar.Size = new System.Drawing.Size(252, 28);
        this.redundancyMacTrackBar.TabIndex = 6;
        this.redundancyMacTrackBar.TextTickStyle =
System.Windows.Forms.TickStyle.None;
        this.redundancyMacTrackBar.TickColor =
System.Drawing.Color.FromArgb(((int) ((byte) 148))), ((int) ((byte) 146))),
((int) ((byte) 148))));
        this.redundancyMacTrackBar.TickHeight = 4;
        this.redundancyMacTrackBar.TickStyle =
System.Windows.Forms.TickStyle.None;
        this.redundancyMacTrackBar.TrackerColor =
System.Drawing.Color.FromArgb(((int) ((byte) 24))), ((int) ((byte) 130))),
((int) ((byte) 198))));
        this.redundancyMacTrackBar.TrackerSize = new System.Drawing.Size(10,
16);
        this.redundancyMacTrackBar.TrackLineColor =
System.Drawing.Color.FromArgb(((int) ((byte) 90))), ((int) ((byte) 93))),
((int) ((byte) 90))));

```

```

        this.redundancyMacTrackBar.TrackLineHeight = 3;
        this.redundancyMacTrackBar.Value = 19;
        this.redundancyMacTrackBar.ValueChanged += new
EConTech.Windows.MACUI.ValueChangedHandler(this.redundancyMacTrackBar_ValueChang
ed);
        this.redundancyMacTrackBar.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.redundancyMacTrackBar_MouseUp);
        //
        // allVolCountMacTrackBar
        //
        this.allVolCountMacTrackBar.BackColor =
System.Drawing.Color.Transparent;
        this.allVolCountMacTrackBar.BorderColor =
System.Drawing.SystemColors.ActiveBorder;
        this.allVolCountMacTrackBar.Font = new
System.Drawing.Font("Verdana", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte) 0));
        this.allVolCountMacTrackBar.ForeColor =
System.Drawing.Color.FromArgb(((int) ((byte) 123))), ((int) ((byte) 125))),
((int) ((byte) 123))););
        this.allVolCountMacTrackBar.IndentHeight = 6;
        this.allVolCountMacTrackBar.Location = new System.Drawing.Point(6,
24);
        this.allVolCountMacTrackBar.Maximum = 15;
        this.allVolCountMacTrackBar.Minimum = 0;
        this.allVolCountMacTrackBar.Name = "allVolCountMacTrackBar";
        this.allVolCountMacTrackBar.Size = new System.Drawing.Size(252, 28);
        this.allVolCountMacTrackBar.TabIndex = 5;
        this.allVolCountMacTrackBar.TextTickStyle =
System.Windows.Forms.TickStyle.None;
        this.allVolCountMacTrackBar.TickColor =
System.Drawing.Color.FromArgb(((int) ((byte) 148))), ((int) ((byte) 146))),
((int) ((byte) 148))););
        this.allVolCountMacTrackBar.TickHeight = 4;
        this.allVolCountMacTrackBar.TickStyle =
System.Windows.Forms.TickStyle.None;
        this.allVolCountMacTrackBar.TrackerColor =
System.Drawing.Color.FromArgb(((int) ((byte) 24))), ((int) ((byte) 130))),
((int) ((byte) 198))););
        this.allVolCountMacTrackBar.TrackerSize = new
System.Drawing.Size(10, 16);
        this.allVolCountMacTrackBar.TrackLineColor =
System.Drawing.Color.FromArgb(((int) ((byte) 90))), ((int) ((byte) 93))),
((int) ((byte) 90))););
        this.allVolCountMacTrackBar.TrackLineHeight = 3;
        this.allVolCountMacTrackBar.Value = 2;
        this.allVolCountMacTrackBar.ValueChanged += new
EConTech.Windows.MACUI.ValueChangedHandler(this.allVolCountMacTrackBar_ValueChan
ged);
        this.allVolCountMacTrackBar.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.allVolCountMacTrackBar_MouseUp);
        //
        // browser
        //
        this.browser.AutoValidate =
System.Windows.Forms.AutoValidate.EnablePreventFocusChange;
        this.browser.ListViewMode = System.Windows.Forms.View.List;
        this.browser.Location = new System.Drawing.Point(12, 131);
        this.browser.Name = "browser";
        treeNode1.Name = "";
        treeNode1.Text = "";
        treeNode2.ImageIndex = 18;
        treeNode2.Name = "backreg";
        treeNode2.SelectedImageIndex = 20;
        treeNode2.Text = "backreg";
        treeNode3.Name = "";
        treeNode3.Text = "";
        treeNode4.ImageIndex = 18;
        treeNode4.Name = "CISCO_CCNA";

```

```

treeNode4.SelectedImageIndex = 20;
treeNode4.Text = "CISCO_CCNA";
treeNode5.Name = "";
treeNode5.Text = "";
treeNode6.ImageIndex = 18;
treeNode6.Name = "Documents and Settings";
treeNode6.SelectedImageIndex = 20;
treeNode6.Text = "Documents and Settings";
treeNode7.Name = "";
treeNode7.Text = "";
treeNode8.ImageIndex = 18;
treeNode8.Name = "Downloads";
treeNode8.SelectedImageIndex = 20;
treeNode8.Text = "Downloads";
treeNode9.Name = "";
treeNode9.Text = "";
treeNode10.ImageIndex = 18;
treeNode10.Name = "Inprise";
treeNode10.SelectedImageIndex = 20;
treeNode10.Text = "Inprise";
treeNode11.Name = "";
treeNode11.Text = "";
treeNode12.ImageIndex = 18;
treeNode12.Name = "Program Files";
treeNode12.SelectedImageIndex = 20;
treeNode12.Text = "Program Files";
treeNode13.ImageIndex = 33;
treeNode13.Name = "Recycled";
treeNode13.SelectedImageIndex = 34;
treeNode13.Text = "Recycled";
treeNode14.ImageIndex = 18;
treeNode14.Name = "RECYCLER";
treeNode14.SelectedImageIndex = 20;
treeNode14.Text = "RECYCLER";
treeNode15.ImageIndex = 18;
treeNode15.Name = "System Volume Information";
treeNode15.SelectedImageIndex = 20;
treeNode15.Text = "System Volume Information";
treeNode16.ImageIndex = 18;
treeNode16.Name = "temp";
treeNode16.SelectedImageIndex = 20;
treeNode16.Text = "temp";
treeNode17.Name = "";
treeNode17.Text = "";
treeNode18.ImageIndex = 18;
treeNode18.Name = "WINDOWS";
treeNode18.SelectedImageIndex = 20;
treeNode18.Text = "WINDOWS";
treeNode19.ImageIndex = 23;
treeNode19.Name = "Локальный диск (C:)";
treeNode19.SelectedImageIndex = 24;
treeNode19.Text = "Локальный диск (C:)";
this.browser.SelectedNode = treeNode19;
this.browser.ShowFoldersButton = false;
this.browser.ShowNavigationBar = false;
this.browser.Size = new System.Drawing.Size(962, 432);
this.browser.SplitterDistance = 398;
this.browser.StartupDirectoryOther = "C:\\\\";
this.browser.TabIndex = 0;
this.browser.Load += new System.EventHandler(this.browser_Load);
//
// testButton
//
this.testButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
this.testButton.Image =
global::RecoveryStar.Properties.Resources.table_sql_view32451;
this.testButton.ImageAlign =
System.Drawing.ContentAlignment.TopCenter;
this.testButton.Location = new System.Drawing.Point(111, 27);

```

```

        this.testButton.Name = "testButton";
        this.testButton.Size = new System.Drawing.Size(100, 97);
        this.testButton.TabIndex = 4;
        this.testButton.Text = "Перевірка цілісності";
        this.testButton.TextAlign =
System.Drawing.ContentAlignment.BottomCenter;
        this.testButton.UseVisualStyleBackColor = true;
        this.testButton.Click += new
System.EventHandler(this.testButton_Click);
        //
        // repairButton
        //
        this.repairButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
        this.repairButton.Image =
global::RecoveryStar.Properties.Resources.medical_bag465471;
        this.repairButton.ImageAlign =
System.Drawing.ContentAlignment.TopCenter;
        this.repairButton.Location = new System.Drawing.Point(210, 27);
        this.repairButton.Name = "repairButton";
        this.repairButton.Size = new System.Drawing.Size(100, 97);
        this.repairButton.TabIndex = 3;
        this.repairButton.Text = "Відновлення цілісності";
        this.repairButton.TextAlign =
System.Drawing.ContentAlignment.BottomCenter;
        this.repairButton.UseVisualStyleBackColor = true;
        this.repairButton.Click += new
System.EventHandler(this.repairButton_Click);
        //
        // recoverButton
        //
        this.recoverButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
        this.recoverButton.Image =
global::RecoveryStar.Properties.Resources.redo6786987;
        this.recoverButton.ImageAlign =
System.Drawing.ContentAlignment.TopCenter;
        this.recoverButton.Location = new System.Drawing.Point(309, 27);
        this.recoverButton.Name = "recoverButton";
        this.recoverButton.Size = new System.Drawing.Size(100, 97);
        this.recoverButton.TabIndex = 2;
        this.recoverButton.Text = "Скидання дисків до початкового стану";
        this.recoverButton.TextAlign =
System.Drawing.ContentAlignment.BottomCenter;
        this.recoverButton.UseVisualStyleBackColor = true;
        this.recoverButton.Click += new
System.EventHandler(this.recoverButton_Click);
        //
        // protectButton
        //
        this.protectButton.BackColor = System.Drawing.SystemColors.Control;
        this.protectButton.FlatStyle = System.Windows.Forms.FlatStyle.Popup;
        this.protectButton.Font = new System.Drawing.Font("Microsoft Sans
Serif", 8.25F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)204));
        this.protectButton.Image =
global::RecoveryStar.Properties.Resources.Database_1_64x64e65768;
        this.protectButton.ImageAlign =
System.Drawing.ContentAlignment.TopCenter;
        this.protectButton.Location = new System.Drawing.Point(12, 27);
        this.protectButton.Name = "protectButton";
        this.protectButton.Size = new System.Drawing.Size(100, 97);
        this.protectButton.TabIndex = 1;
        this.protectButton.Text = "Створення RaidSSD масиву";
        this.protectButton.TextAlign =
System.Drawing.ContentAlignment.BottomCenter;
        this.protectButton.UseVisualStyleBackColor = false;
        this.protectButton.Click += new
System.EventHandler(this.protectButton_Click);
        //
        // ToolStripMenuItem

```

```

//
    this.ToolStripMenuItem.Image =
global::RecoveryStar.Properties.Resources.Exit;
    this.ToolStripMenuItem.Name = "ToolStripMenuItem";
    this.ToolStripMenuItem.Size = new System.Drawing.Size(152, 22);
    this.ToolStripMenuItem.Text = "Вихід";
        this.ToolStripMenuItem.Click += new
System.EventHandler(this.виходToolStripMenuItem_Click);
//
// ToolStripMenuItem
//
        this.тестБыстродействияToolStripMenuItem.Image =
global::RecoveryStar.Properties.Resources.StartBenchmark;
    this.тестБыстродействияToolStripMenuItem.ImageScaling =
System.Windows.Forms.ToolStripItemImageScaling.None;
    this.ToolStripMenuItem.Name = "ToolStripMenuItem";
    this.ToolStripMenuItem.Size = new System.Drawing.Size(161, 22);
    this.ToolStripMenuItem.Text = "Тест швидкодії";
        this.ToolStripMenuItem.Click += new
System.EventHandler(this.тестБыстродействияToolStripMenuItem_Click);
//
// MainForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(986, 576);
this.Controls.Add(this.coderConfigGroupBox);
this.Controls.Add(this.testButton);
this.Controls.Add(this.repairButton);
this.Controls.Add(this.recoverButton);
this.Controls.Add(this.protectButton);
this.Controls.Add(this.browser);
this.Controls.Add(this.menuStrip);
        this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedDialog;
this.Icon =
((System.Drawing.Icon) (resources.GetObject("$this.Icon")));
this.MainMenuStrip = this.menuStrip;
this.MaximizeBox = false;
this.Name = "MainForm";
        this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Гарантоване збереження інформації ";
this.Load += new System.EventHandler(this.MainForm_Load);
    this.FormClosing += new
System.Windows.Forms.FormClosingEventHandler(this.MainForm_FormClosi
ng);
    this.menuStrip.ResumeLayout(false);
    this.menuStrip.PerformLayout();
    this.coderConfigGroupBox.ResumeLayout(false);
    this.redundancyGroupBox.ResumeLayout(false);
    this.redundancyGroupBox.PerformLayout();
    this.allVolCountGroupBox.ResumeLayout(false);
    this.allVolCountGroupBox.PerformLayout();
    this.ResumeLayout(false);
    this.PerformLayout();

}

#endregion

private System.Windows.Forms.MenuStrip menuStrip;
private System.Windows.Forms.ToolStripMenuItem ToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem ToolStripMenuItem;
private System.Windows.Forms.Button protectButton;
private System.Windows.Forms.ToolStripMenuItem ToolStripMenuItem;
    private System.Windows.Forms.ToolStripSeparator
separatorToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem ToolStripMenuItem;

```

```
private System.Windows.Forms.Button repairButton;
private System.Windows.Forms.Button testButton;
private System.Windows.Forms.GroupBox coderConfigGroupBox;
private System.Windows.Forms.GroupBox redundancyGroupBox;
private System.Windows.Forms.GroupBox allVolCountGroupBox;
private EConTech.Windows.MACUI.MACTrackBar allVolCountMacTrackBar;
private EConTech.Windows.MACUI.MACTrackBar redundancyMacTrackBar;
private System.Windows.Forms.ToolTip toolTip;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
internal FileBrowser.Browser browser;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
private System.Windows.Forms.ToolStripItem ToolStripMenuItem;
private System.Windows.Forms.Button recoverButton;
}
}
```

K6713-2023