Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Автоматизована система керування розумним будинком»

Виконав: студент 2 курсу, групи 2АКІТР-23м спеціальності 174 – Автоматизація, комп'ютерно-інтегровані технології, та робототехніка
Віталій Бажан
Керівник: Зав. кафедри КСУ, проф.
В'ячеслав КОВТУН
«11» лютого 2025 р.
Опонент: к.т.н., доцент кафедри АІІТ
Юрій ІВАНОВ
«12» лютого 2025 р.

Допущено до захисту
Зав. кафедри КСУ
В'ячеслав КОВТУН
«14» лютого 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти другий (магістерський)
Галузь знань – 17 – Електроніка, автоматизація та електронні комунікації
Спеціальність – 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Освітньо-професійна програма – Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ
Зав. кафедри КСУ

В'ячеслав КОВТУН

"14" жовтня 2024 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Бажану Віталію Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Автоматизована система керування розумним будинком.
керівник роботи Ковтун В.В.
затверджені наказом ВНТУ від "17" вересня 2024 року №310
2. Термін подання студентом роботи "1" лютого 2025 року
3. Вихідні дані до роботи: вихідні дані до роботи представлені у структурованому csv файлі. Формуючи вихідні дані автор використовував такі інформаційні джерела:

1. O. Popov, A. Yatsyshyn, and I. Ivanov, "Designing a Smart Home System Based on IoT Technologies," Electronics and Control Systems, vol. 2, no. 60, pp. 80-88, 2021. [Online]. Available: https://doi.org/10.18372/1990-5548.60.15633. (9 стор.)

2. M. Smith and K. Johnson, "Mobile-Oriented Cloud-Based Smart Home Data Processing," IEEE Access, vol. 8, pp. 24012-24022, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2973345. (11 стор.)

3. P. Borysov, "Internet of Things and Mobile Application Design for Smart Home Data Management," Scientific Bulletin of the National Technical University of Ukraine, no. 5, pp. 45-52, 2021. (8 стор.)

4. Зміст текстової частини: текстова частина включає анотацію, зміст, вступ, основну частину з першим, оглядовим, розділом, висновки, список посилань та додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): об'єкт, предмет та мета дослідження, структурна схема запропонованої програмно-апаратної системи, алгоритми роботи програної складової розробки, результати тестуваннння створеної системи

1. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
4	Кавецький В.В. – доцент кафедри економіки підприємства і виробничого менеджменту		

2. Дата видачі завдання "14" жовтня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1.	Аналітичний огляд переваг та недоліків, які характеризують досліджуваний технологічний процес	09.12.24	11.12.24	
2.	Дослідження стану розв'язання технічної задачі	23.12.24	29.12.24	
3.	Розробка та тестування системи	01.01.25	19.01.25	
4.	Розрахунок економічної частини	21.01.25	22.01.25	
5.	Оформлення матеріалів до захисту	26.01.25	28.01.25	
6.	Захист МКР	18.02.25	18.02.25	

Студент Віталій Бажан
(підпис) (*Ім'я ПРІЗВИЩЕ*)

Керівник роботи В'ячеслав КОВТУН
(підпис) (*Ім'я ПРІЗВИЩЕЕ*

АНОТАЦІЯ

УДК 669.05

Бажан В.В. Автоматизована система керування розумним будинком. Магістерська кваліфікаційна робота зі спеціальності 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка. Вінниця: ВНУ, 2025. 148 с.

Українською мовою, 4 розділи, 27 джерел, рис.: 25, табл.: 15.

Мета роботи – розробка та впровадження автоматизованої системи керування розумним будинком, яка забезпечує інтеграцію різних пристроїв та підсистем для підвищення енергоефективності, безпеки та комфорту мешканців. Система має використовувати сучасні технології Інтернету речей (IoT), штучного інтелекту та хмарних обчислень для оптимізації керування освітленням, опаленням, безпекою та іншими інженерними системами будинку.

У магістерській кваліфікаційній роботі розроблено автоматизовану систему керування розумним будинком, що забезпечує підвищення комфорту, енергоефективності та безпеки житлових приміщень. У загальній частині роботи проаналізовано сучасні тенденції розвитку систем автоматизації, визначено основні функціональні вимоги до розумного будинку та обґрунтовано доцільність впровадження таких систем.

У розрахунково-конструкторській частині виконано проектування архітектури системи, розроблено алгоритми керування основними компонентами (освітленням, клімат-контролем, системою безпеки), а також проведено розрахунок надійності системи. У технологічній частині представлено процес інтеграції системи з різними пристроями «Інтернету речей» (IoT), методики налаштування мережевих підключень та пошуку можливих несправностей.

Графічна частина містить схеми взаємодії компонентів, результати моделювання роботи системи, а також інтерфейс користувача. У розділі охорони праці розглянуто питання безпеки експлуатації системи, можливі техногенні

ризики, розроблено рекомендації щодо забезпечення електромагнітної сумісності та відповідності нормам пожежної безпеки.

Ключові слова: розумний будинок, автоматизація, IoT, енергоефективність, система безпеки, клімат-контроль.

ANNOTATION

UDC 669.05

Bazhan V.V. Automated smart home control system. Master's qualification work in specialty 174 - Automation, computer-integrated technologies and robotics. Vinnytsia: VNTU, 2025. 148 p.

In Ukrainian, 4 chapters, 27 sources, fig.: 25, tab.: 15.

The purpose of the work is to develop and implement an automated smart home control system that provides integration of various devices and subsystems to increase energy efficiency, safety and comfort of residents. The system should use modern technologies of the Internet of Things (IoT), artificial intelligence and cloud computing to optimize the control of lighting, heating, security and other engineering systems of the house.

In the master's qualification work, an automated smart home control system has been developed that provides increased comfort, energy efficiency and safety of residential premises. The general part of the work analyzes modern trends in the development of automation systems, determines the main functional requirements for a smart home, and justifies the feasibility of implementing such systems.

In the design and calculation part, the system architecture is designed, algorithms for controlling the main components (lighting, climate control, security system) are developed, and the system reliability is calculated. The technological part presents the process of integrating the system with various Internet of Things (IoT) devices, methods for configuring network connections and searching for possible malfunctions.

The graphic part contains diagrams of component interaction, results of system operation modeling, and the user interface. The occupational health and safety section considers the issues of system operation safety, possible man-made risks, and develops recommendations for ensuring electromagnetic compatibility and compliance with fire safety standards.

Keywords: smart home, automation, IoT, energy efficiency, security system, climate control.

ЗМІСТ

ВСТУП

Актуальність: автоматизація управління розумним будинком та створення зручних інтерфейсів для користувачів є актуальними питаннями на сьогоднішній день. Багато компаній розробляють інтерфейси, які не завжди є інтуїтивно зрозумілими для користувачів, особливо для тих, хто не має глибоких знань в області автоматизації. Головною метою роботи є розробка графічного інтерфейсу управління системою розумного будинку таким чином, щоб кожен користувач міг без проблем створювати та впевнено використовувати систему.

Системи автоматизації розумного будинку значно полегшують процес управління та моніторингу різноманітних функцій, таких як освітлення, безпека, вентиляція і температурний режим. Раніше ці системи дозволяли користувачеві контролювати функції будинку, виконуючи команди, які надавалися вручну. Однак на сьогоднішній день завдяки алгоритмам штучного інтелекту, орієнтованим на енергозбереження, безпеку, автоматичне управління та інші аспекти, ці системи можуть працювати автономно. Багато наукових установ та приватних компаній активно працюють над створенням нових алгоритмів для вдосконалення продуктів у цій галузі. Алгоритми штучного інтелекту потребують регулярного тестування та вдосконалення в різних умовах, щоб гарантувати їхню ефективність та правильне виконання заданих функцій. Проблеми виникають під час тестування алгоритмів у реальних умовах, що може призвести до зниження точності результатів. Тому для розробки та вдосконалення алгоритмів часто використовують модельовані середовища. Проте важко врахувати всі змінні реального життя при моделюванні. Враховуючи це, у рамках цього дослідження запропоновано гібридне моделювання розумного будинку, яке поєднує реальні та віртуальні системи для тестування алгоритмів штучного інтелекту, призначених для управління розумним будинком. Реальна система розумного будинку була вперше розроблена та встановлена в приміщенні для гібридного моделювання.

Потім створено змодельований будинок, що містить потрібну кількість кімнат та компонентів системи з різними функціями. Кімната з реальною системою розумного будинку перетворюється на кімнату віртуального будинку за допомогою змін у кодах моделювання. Людина, яка використовує справжню систему, стає мешканцем віртуального будинку, а також для тестування створено віртуальних мешканців, які взаємодіють з реальними пристроями. Таким чином, у процесі гібридного моделювання реальний та віртуальний будинки, а також їхні мешканці, функціонують спільно. Гібридне моделювання працює стабільно протягом двох місяців при використанні різних алгоритмів штучного інтелекту в різноманітних умовах.

Об'єкт дослідження: процеси автоматизованого керування системами розумного будинку з використанням IoT, сенсорних мереж та програмного забезпечення.

Предмет дослідження: інтерфейс управління життєдіяльністю людини в умовах розумного будинку.

Предмет дослідження: методи, алгоритми та технічні засоби для побудови автоматизованої системи керування розумним будинком з урахуванням енергоефективності, безпеки та зручності для користувачів.

Мета роботи: адаптація та моделювання роботи всіх пристроїв розумного будинку, перевірка сумісності приладів системи, а також оптимізація процесу управління та моніторингу всієї системи.

Завдання роботи: провести аналіз існуючих систем розумного будинку та визначити їх переваги та недоліки. Дослідити сучасні технології IoT, сенсорні мережі та алгоритми керування. Розробити архітектуру автоматизованої системи керування. Реалізувати програмно-апаратний комплекс для інтеграції компонентів розумного будинку. Виконати тестування запропонованої системи та оцінити її ефективність. Запропонувати рекомендації щодо подальшого розвитку та вдосконалення системи.

Наукова новизна: розробка умов для забезпечення життєдіяльності мешканців будинку, а також інтеграція системи охорони в загальну структуру.

Практичне значення: оптимізації енергоспоживання в житлових та комерційних будівлях. Підвищення рівня комфорту та безпеки мешканців. Зниження витрат на утримання будівлі за рахунок автоматичного регулювання ресурсів. Використання у сфері будівництва та інженерії для впровадження інтелектуальних систем у новобудовах.

Апробація результатів дослідження: представлені в роботі результати апробовані в результаті участі в конференції «LIII Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024)».

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд наявних рішень

Гібридна система автоматизації. Процес створення гібридного моделювання для розумного будинку можна умовно поділити на три основні етапи. Спершу була розроблена базова система розумного будинку, після чого було створено модель, яка відповідає архітектурі цього будинку та організації даних. Заключний етап полягав у поєднанні цих двох частин в єдину гібридну систему, здатну функціонувати в реальному часі [1].

Базова модель розумного будинку. У рамках дослідження спочатку була розроблена хмарна система для розумного будинку, яка дозволяє здійснювати контроль над температурою, освітленням та електричними розетками в кімнатах через Інтернет, а також реєструє всі дії, які здійснюються в системі. Цей розумний будинок складається з термостатного пристрою для управління температурою, розеток з вбудованими датчиками електричного струму, температури, вологості та освітленості, кімнатного контролера для моніторингу та управління усіма компонентами, а також хмарного сервера для керування та налаштування системи через веб-інтерфейс. Механізм роботи системи можна описати наступним чином: користувач має можливість переглядати дані зі системи через веб-інтерфейс та здійснювати управління компонентами. Іншими словами, коли користувач бажає увімкнути освітлення, вимкнути електричну розетку або відрегулювати температуру, він виконує ці дії через веб-інтерфейс, який виступає як основна точка управління для всієї системи. Веб-інтерфейс, представлений на рисунку 1, працює на основі програмного забезпечення веб-сервера (Apache), яке інстальовано на центральному сервері.

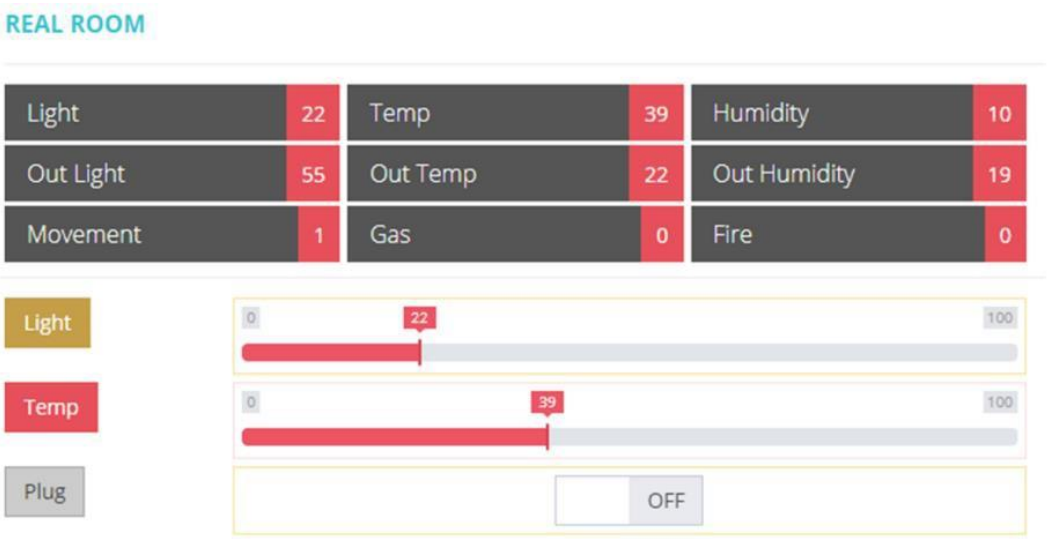


Рисунок 1.1 – Веб-додаток для управління «базовим» розумним будинком.

Центральний сервер являє собою серверний комп'ютер, який постійно функціонує, підключений до Інтернету та може бути фізично розміщений у будь-якому місці (у поточній системі цей сервер розташований в серверній кімнаті університету). Цей сервер здійснює передачу транзакцій, виконаних користувачем через веб-інтерфейс, до відповідного контролера кімнати, який підключений до мережі Інтернет. Веб-інтерфейс, який працює на центральному сервері, сумісний із мобільними пристроями, завдяки використанню мов HTML, CSS та Javascript. Тому окремий мобільний додаток для цієї системи не розроблявся. Під час першого входу в веб-інтерфейс користувачі виконують процес авторизації. Після цього вони отримують доступ до екрана, на якому відображаються дані про температуру, вологість та рівень освітлення в приміщеннях, а також мають можливість регулювати компоненти розумного будинку, такі як термостат та освітлення (див. Рис. 1.1). На сервері працює MQTT сервер, що відповідає за передачу транзакцій до розумного будинку та відображення даних від датчиків на веб-інтерфейсі. MQTT є популярним міжмережевим (M2M) протоколом для передачі повідомлень через Інтернет. Сервер виконує операції, пов’язані з

управлінням розумним будинком, через підключений контролер кімнати. Крім того, система управління базами даних MySQL на сервері реєструє всі транзакції, здійснені користувачем, дані від датчиків, а також інформацію про користувача, пристрій та інші важливі відомості, які використовуються в процесах штучного інтелекту.

У кожній кімнаті розумного будинку встановлений контролер, який передає запити з веб-інтерфейсу на сервер для виконання завдань щодо термостата, освітлення та розеток. Ці компоненти реагують відповідно до заданих параметрів. Крім того, контролер кімнати передає зібрані ним дані від датчиків на сервер для подальшої обробки.

Термостат є ключовим елементом для підтримки стабільної температури в межах заданого діапазону. У даній системі його основна функція полягає у забезпеченні підтримки температури кімнати на рівні, визначеному користувачем. Компонент освітлення відповідає за регулювання яскравості світла в кімнаті відповідно до налаштувань користувача. Компонент розетки дозволяє вмикати та вимикати електричні розетки. Всі ці компоненти бездротово підключені до контролера через Bluetooth, що дозволяє розміщувати їх у будь-якому зручному місці кімнати. Автори цієї гібридної системи моделювання управління розумним будинком — Сабрі Бічакчі та Хусейн Гюнес.

1.2 Інструменти для створення компонентів розумного будинку

У ході дослідження був розроблений спеціальний інструмент, що дозволяє створювати необхідну кількість кімнат, тим самим забезпечуючи можливість віртуального формування будинку. За допомогою цього інструменту створення кімнати відбувається шляхом введення її назви та короткого опису. Після цього був розроблений інший інструмент для додавання компонентів системи «розумного» будинку до вже створених кімнат. Цей інструмент дозволяє користувачам додавати різні компоненти, вказуючи тип пристрою, відповідну

кімнату та найменування самого пристрою. Скріншот інтерфейсу для додавання кімнат та пристроїв представлений на рис. 1.2.

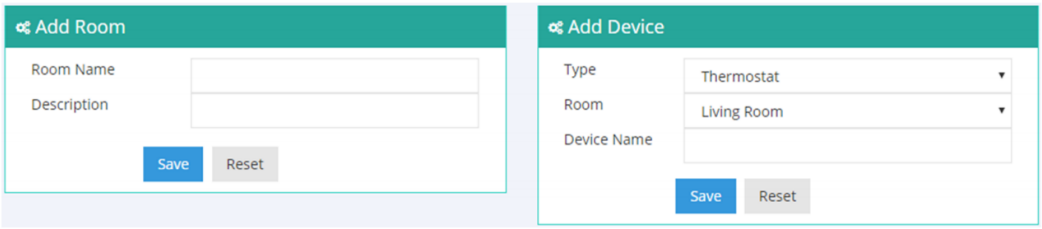


Рисунок 1.2 – Вікна додатку для додавання кімнат та пристроїв у систему «розумного» будинку.

Для моделювання віртуальних мешканців, які заміняють реальних людей, був розроблений спеціальний інструмент створення віртуальних осіб [2]. Під час створення кожної особи в системі необхідно вказати коротке ім’я, яке буде відображатися в процесі моделювання, а також надати стислу інформацію, що ідентифікує цю людину. Після того як віртуальна особа була створена, став можливим доступ до інструменту, що дозволяє налаштувати життєві сценарії для кожної особи. Цей інструмент визначає, коли віртуальна людина входить чи виходить із кімнати, коли вона приходить додому або покидає його, які пристрої вона вмикає або вимикає, а також вказує значення температури та яскравості, що є оптимальними для її перебування в приміщенні. Сценарії змінюються щотижня, враховуючи поведінку віртуального мешканця. На рис. 1.3 представлений інтерфейс для створення сценаріїв.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Wake Up	6:25	6:35	6:45	6:40	6:30	10:00	10:00
Sleep	23:10	23:15	23:10	23:15	0:00	2:00	23:10
Home Enter	17:31 [09:00-17:30-21:50]	17:30-22:10 [09:00-17:30-21:50]	17:30 [09:00-17:30-21:50]	17:30-22:10 [09:00-17:30-21:50]	17:30 [09:00-17:30-21:50]	17:16-19:00-I [09:00-17:30-21:50]	17:20 [09:00-17:30-21:50]
Home Exit	07:30 [09:00-17:30-21:50]	07:30-19:50 [09:00-17:30-21:50]	07:30 [09:00-17:30-21:50]	07:30-19:50 [09:00-17:30-21:50]	07:30 [09:00-17:30-21:50]	17:00-23:50 [09:00-17:30-21:50]	12:00 [09:00-17:30-21:50]
Living Room Enter	21:00-18:30 [09:00-17:30-21:50]	20:55-18:30 [09:00-17:30-21:50]	20:50-18:30 [09:00-17:30-21:50]	21:10-18:30 [09:00-17:30-21:50]	21:15-18:30 [09:00-17:30-21:50]	20:40 [09:00-17:30-21:50]	20:55 [09:00-17:30-21:50]

Рисунок 1.3 – Засіб для щотижневого введення сценарію для віртуальної особи.

Незважаючи на те, що люди мають схожі звички, неможливо стверджувати, що вони виконують однакові дії в один і той самий час кожного дня. Наприклад, хоча людина часто повертається додому після роботи приблизно о 17:00, це не відбувається строго за годину, і можуть бути незначні відхилення від заданого часу. Такі відхилення є частими в контексті рутинних повсякденних завдань. Враховуючи ці варіації, для кожного параметра, як-от час входу чи виходу, а також для інших налаштувань, які визначають поведінку віртуальної особи, можна встановити допустимі межі відхилень, які враховують реалії повсякденного життя. Це дозволяє коригувати та адаптувати сценарії для віртуальних осіб з урахуванням цих варіативних факторів, застосовуючи інструмент сценарного введення.

Інфраструктура моделювання. Після створення моделі розумного будинку, його компонентів, мешканців та сценаріїв, було розроблено спеціалізоване програмне забезпечення, що дозволяє здійснювати моделювання на основі цих даних. Для реалізації використовувалася мова програмування JavaScript, яка працює в середовищі Node.js. Завдяки застосуванню PM2 (інструмент для

керування процесами в Node.js, що забезпечує розширену функціональність), моделювання може бути автоматично перезапущене у разі збоїв або перезавантаження системи, що гарантує безперервну роботу.

Веб-інтерфейс був реалізований за допомогою технологій CSS, JavaScript та HTML, що дозволяє здійснювати моніторинг моделювання та втручатися в процес за необхідності. Цей інтерфейс надає можливість в реальному часі та з висоти пташиного польоту відображати розумний будинок, його компоненти та індивідуумів (див. Рис. 1.4).

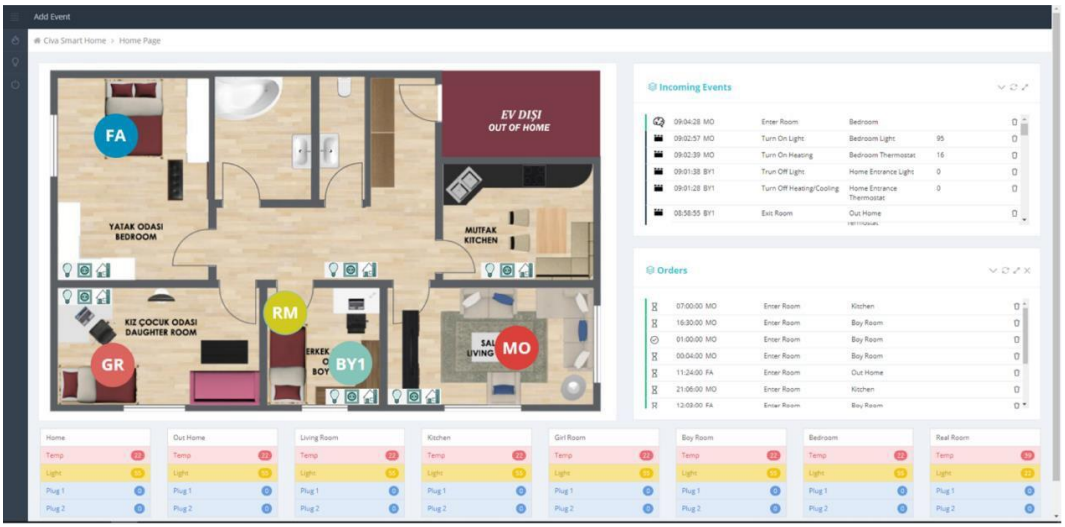


Рисунок 1.4 – Інтерфейс веб-моделювання

На Рис. 1.4 зображено інтерфейс веб-моделювання, де кола, що символізують віртуальних людей, відображаються з висоти пташиного польоту над розумним будинком. Ці кола автоматично змінюють своє місцезнаходження в залежності від переміщення людей між кімнатами. Положення значків розміщується випадковим чином, щоб уникнути їх перекриття. Значки ламп, температури та розеток в кімнатах позначають наявні компоненти розумного будинку в конкретному приміщенні.

Праворуч від інтерфейсу розташовані дві таблиці. Верхня таблиця містить інформацію про майбутні транзакції, що відображають події, які відбуватимуться

за заздалегідь визначеними сценаріями. Нижня таблиця показує розпорядження, які надаються менеджером поза сценарієм. Додатково, внизу екрану містяться маленькі таблиці, які демонструють дані про температуру, рівень освітлення та стан розеток у всіх кімнатах розумного будинку.

Меню розташоване в верхній та лівій частинах екрана, зокрема ліворуч. Меню з лівого боку включає блоки управління, де можна безпосередньо керувати компонентами розумного будинку в кожній кімнаті. Через це меню можна швидко переглянути та змінити налаштування температури, освітлення або розеток у вибраній кімнаті. Рисунок 6 ілюструє меню, через яке здійснюється налаштування температури для всіх кімнат.

За допомогою меню «Додати подію», розташованого вгорі екрану, можна додавати нові дії поза межами сценаріїв. Наприклад, через це меню, зображене на Рис. 7, можна додати нову подію, таку як переміщення особи між кімнатами або зміну налаштувань компонентів розумного будинку, вказавши точний час для виконання, після чого система автоматично виконає цю подію в зазначений момент.

1.2.1 Інструмент автоматизації zVirtualScenes zVirtualScenes

zVirtualScenes zVirtualScenes (версія 3.1) — це програмний контролер для управління сценами Z-Wave, що дає змогу створювати індивідуальні сцени Z-Wave, які можна відтворювати через графічний інтерфейс, HTTP-команди або мобільні додатки. Застосунок підтримує інтеграцію з іншими платформами автоматизації розумного будинку, такими як [назва аналога].

Рисунок 1.5 – Інтерфейс додатку zVirtualScenes на всіх платформах

– Отримуйте повний контроль над вашими сумісними пристроями Z-Wave (пристрої з бездротовим зв’язком для автоматизації будинку) за допомогою інтуїтивно зрозумілого графічного інтерфейсу [3].

– zVirtualScenes відслідковує зміни стану пристроїв Z-Wave та може надсилати сповіщення через Jabber / Google Talk (програми для миттєвого обміну повідомленнями). Наприклад, отримуйте миттєві сповіщення, якщо температура опускається нижче заданого рівня, або коли змінюється рівень перемикача або вмикається термостат.

– Керуйте вашими пристроями Z-Wave з мобільного телефону (Android та iPhone).

– Використовуйте мобільний додаток zVirtualScenes на базі HTML5 (мова розмітки для створення веб-сторінок) / JavaScript (мова програмування для створення інтерактивних веб-сторінок) для повного контролю за пристроями з будь-якої точки світу.

– Легко створюйте індивідуальні сцени для керування термостатами, перемикачами та іншими пристроями одним простим кроком. Активуйте сцени за допомогою смартфона, HTTP (протокол передачі гіпертекстових даних) або графічного інтерфейсу.

– Повністю протестовано з дистанційним терморегулятором Trane Z-Wave.

– Завдяки плагіну Web API (інтерфейс програмування додатків для взаємодії з веб-сервісами) ви можете взаємодіяти з усіма функціями zVirtualScenes через RESTful API (архітектура програмування для розробки веб-сервісів). Створюйте сцени, змінюйте імена пристроїв, виконуйте команди тощо.

– zVirtualScenes має розширену плагін-систему з повним API, що дозволяє розробникам створювати плагіни за допомогою Visual C# (мова програмування для створення додатків).

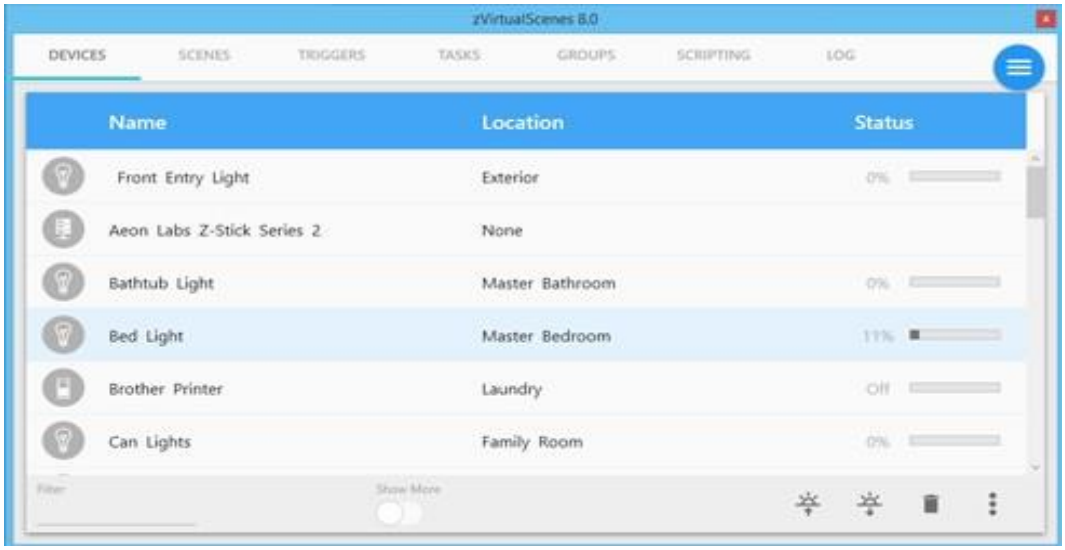


Рисунок 1.6 – Інтерфейс додатку zVirtualScenes для управління пристроями.

Станом на версію 3.5 додано підтримку JavaScript, який використовується для створення команд у сценах. Ця функція дозволяє створювати більш гнучкі та потужні сценарії для автоматизації. На цій сторінці наведено огляд того, як використовувати нову функцію.

Основний доступ до редагування сценаріїв можна знайти в головному меню, в розділі команд, де є пункт меню "Додати / редагувати JavaScript".

Щоб зв'язати конкретний сценарій із сценою, відредагуйте вашу сцену, і внизу праворуч ви побачите кнопку "Додати команди". Клацнувши на неї, виберіть "Додати команду JavaScript", що дозволить вам вибрати сценарій, який ви раніше редагували.

Існує два способи використання сценаріїв. Перший метод – це безпосереднє редагування сценарію в інтерфейсі, виконуючи всі необхідні дії. Базову бібліотеку, яку ми використовуємо, можна знайти за посиланням: http://jint.codeplex.com. Зверніть увагу, що ви маєте повний доступ до простору імен .NET, що дозволяє використовувати такі функції, як:

```csharp
logInfo(System.DateTime.Now);
```

Другим методом є використання функції `require()`. У папці "сценарії" в інсталяційній директорії програми розміщуйте файли .js, а потім імпортуйте їх у основний сценарій, наприклад:

```javascript
require("gmail.js");
```

1.2.2 Система автоматизації Ago Control

Ago Control — це система управління пристроями, що пропонує комплексне рішення для автоматизації будинку. Вона також має потенціал для застосування в різних інших галузях, таких як сільське господарство, де автоматизація допомагає керувати різноманітними процесами, від поливу до моніторингу погодних умов. Однією з ключових особливостей є використання шини повідомлень AMQP (Advanced Message Queuing Protocol), що дозволяє системі ефективно передавати дані між пристроями з високою надійністю та низьким рівнем затримки. Цей

легкий і відкритий протокол зручний як для людей, так і для машин, що дозволяє інтегрувати систему в різноманітні технологічні екосистеми.

Ago Control має сучасну модульну архітектуру, що дозволяє гнучко налаштовувати систему для конкретних потреб користувача. Хмарні функції забезпечують віддалене управління пристроями, що може бути особливо корисним для моніторингу та управління системами в режимі реального часу.

Система відзначається високою продуктивністю, працюючи навіть на невеликих вбудованих пристроях, таких як Raspberry Pi, а також підтримує кілька обчислювальних платформ, зокрема Sheevaplug, Guruplug та Pogoplug, що робить її доступною для широкого кола користувачів. Ця гнучкість дозволяє впроваджувати Ago Control в різноманітні середовища, від невеликих побутових автоматизацій до масштабних промислових рішень[3].

Завдяки такій універсальності, Ago Control підходить для використання як у побутових умовах, так і в складних інфраструктурних проектах, зокрема для автоматизації сільськогосподарських процесів, які можуть включати управління температурою, вологістю та іншими критичними параметрами.

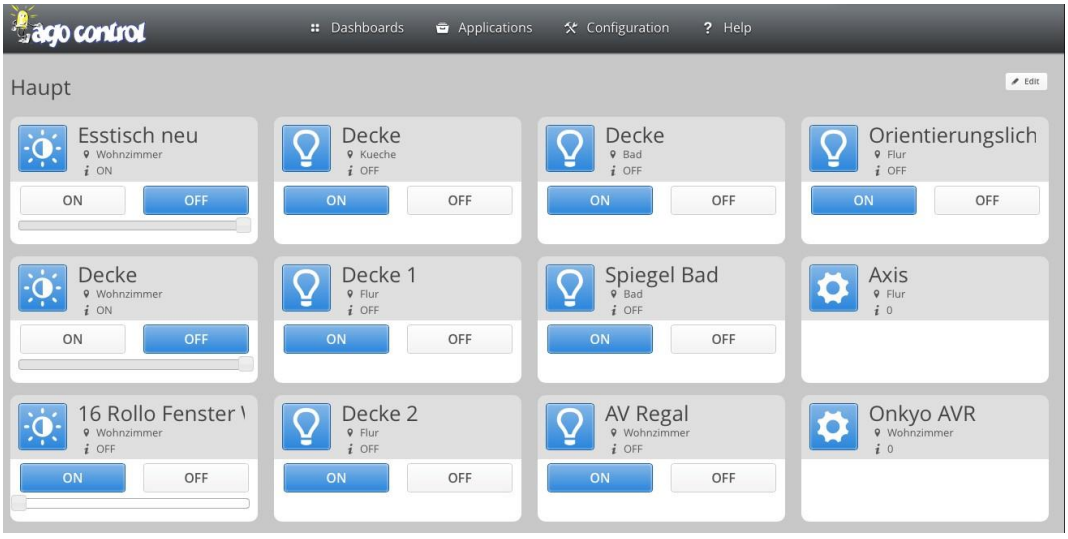


Рисунок 1.7 – Інтерфейс додатку Ago Control

Ago Control є потужною і масштабованою системою автоматизації, яка підтримує численні пристрої та протоколи. Це дозволяє інтегрувати різні технології та пристрої в єдину систему для автоматизації домашніх, офісних або навіть сільськогосподарських процесів. Ось деякі з підтримуваних пристроїв та протоколів:

- Z-Wave: Поширений стандарт для домашньої автоматизації, який дозволяє з'єднувати різноманітні бездротові пристрої.

- KNX: Протокол для автоматизації будівель, який забезпечує управління освітленням, отопленням, безпекою, енергоспоживанням тощо.

- 433 МГц трансивери: Використовуються для бездротового зв'язку на базі частоти 433 МГц, популярні в побутовій автоматизації.

- EnOcean: Протокол для бездротових сенсорів і пристроїв, який працює на енергії, отриманій від навколишнього середовища (наприклад, від сонячного світла).

- X10: Старіший стандарт для автоматизації будинку через електричні мережі.

- 1-Wire: Протокол для моніторингу та контролю температури та інших сенсорних даних.

- АТС Asterisk: Відкрита телефонна система, яка може бути інтегрована в систему автоматизації для керування дзвінками та голосовими повідомленнями.

- Dreambox / Enigma2: Телевізійні приймачі та приставки, які дозволяють інтегрувати телевізійні функції в систему автоматизації.

- Onkyo eISCP AVR: Аудіо-відео ресивери, які підтримують інтелектуальні функції управління через відкритий протокол.

- Chromoflex USP3 RGB: Світлодіодні диммери для управління кольоровим освітленням.

- APC (PDU): Блоки розподілу живлення для ефективного управління енергоспоживанням в мережах.

- OpenLightingArchitecture (OLA): Протокол для освітлення, який дозволяє управляти освітлювальними системами через DMX.

- Phillips (Jointspace): Інтерфейс для інтеграції телевізорів Phillips у систему автоматизації.

- Arduino Firmata: Підтримка для програмування та інтеграції пристроїв на базі Arduino в систему.

- IRTrans Ethernet: Інфрачервоні бластери для управління пристроями за допомогою інфрачервоного сигналу.

- Kwikwai Ethernet HDMI CEC: Міст Ethernet для управління пристроями через HDMI CEC (Consumer Electronics Control).

- Rain8net: Контролер для автоматизації зрошення, який дозволяє інтегрувати зрошувальні системи в екосистему автоматизації.

- BlinkM LED: Технологія управління світлодіодами, яка дозволяє створювати ефекти кольорового освітлення.

Розширення системи Ago Control:

Велика кількість підтримуваних пристроїв дозволяє користувачам налаштовувати систему автоматизації для задоволення конкретних потреб. Враховуючи, що Ago Control активно підтримує нові драйвери та пристрої, користувачі можуть самостійно розширювати функціональність системи, додаючи нові пристрої через спеціальні драйвери або скрипти. Це дозволяє створювати персоналізовані рішення для автоматизації будинку, офісу, чи навіть виробничих і сільськогосподарських процесів.

Основні компоненти системи Ago Control:

1. Resolver – відповідає за реєстрацію пристроїв і їх іменування. Це дозволяє системі визначати і взаємодіяти з пристроями на різних етапах їх роботи.

2. Таймер – цей компонент ініціює події на основі часу, що є важливим для автоматизації завдань, що повинні відбуватися в певний час (наприклад, увімкнення освітлення або кондиціонера).

3. Конфігурація подій – дозволяє налаштувати тригери для подій, що відбуваються за певними умовами. Це важливо для автоматизації, щоб дії виконувалися тільки при виконанні певних умов (наприклад, якщо температура підвищується вище за задану межу).

4. Конфігурація сценаріїв – дозволяє створювати сценарії для групування команд управління, що дозволяє ефективно автоматизувати складні дії (наприклад, "вечірній сценарій" для увімкнення освітлення, розкриття штор, і включення музики).

5. RPC інтерфейс – дає змогу віддалене керування системою через інтерфейси віддаленого виклику процедур (Remote Procedure Calls), що дає зручність у взаємодії з іншими системами чи додатками [4].

6. Веб-адміністратор – забезпечує зручне налаштування та моніторинг пристроїв через веб-браузер. Це дозволяє користувачам управляти системою без потреби у фізичному доступі до серверів або пристроїв.

Система Ago Control є високопродуктивною і гнучкою, що дозволяє забезпечити ефективну автоматизацію для широкого спектра застосувань. Користувачі можуть налаштовувати систему під свої індивідуальні потреби, використовуючи підтримувані протоколи та пристрої для створення власних сценаріїв автоматизації.

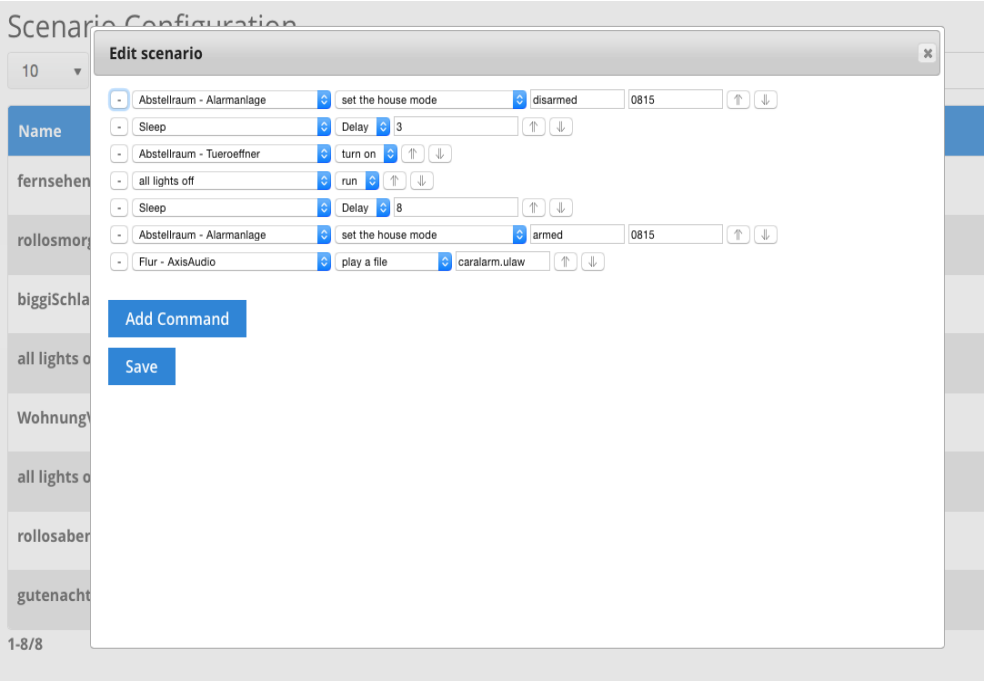


Рисунок 1.8 – Інтерфейс створення сценаріїв Ago Control.

– Реєстратор даних: компонент, який реєструє всі події, що надсилаються пристроями у системі. Це дає змогу відстежувати та зберігати інформацію про всі дії, виконувані в межах автоматизації, що важливо для аналізу та моніторингу.

– Blockly: інструмент для створення дій у системі Ago Control. Це візуальний інтерфейс, який дозволяє створювати складні сценарії без потреби знання програмування. Користувачі можуть будувати ланцюги дій за допомогою блоків, що спрощує процес налаштування автоматизації навіть для початківців.

– SecuritySystem: модуль, що надає функціональність сигналізації для системи. Він відповідає за безпеку автоматизованого будинку або іншого об'єкта, забезпечуючи тривогу та сповіщення у разі виявлення несанкціонованого доступу чи інших загроз.

Ці інструменти працюють разом, дозволяючи створювати ефективні, автоматизовані сценарії для управління пристроями, а також забезпечувати безпеку та моніторинг в реальному часі.

1.2.3 Система автоматизації ioBroker

ioBroker – це програмне рішення для інтеграції різних систем розумного будинку, що дозволяє об'єднати автономні рішення в єдину платформу. ioBroker є інтеграційною платформою для Інтернету речей (IoT), що має модульну структуру, що дозволяє з'єднувати понад 200 різних платформ, від голосових помічників Alexa до систем запису часу.

Система ioBroker забезпечує інтеграцію як з комерційними продуктами, так і з внутрішніми рішеннями. Оскільки кожна система має свої сильні і слабкі сторони, ioBroker дозволяє користувачам ефективно поєднувати різні технології, використовуючи ефекти синергії. Це дає змогу обирати найкраще з доступних рішень.

ioBroker сумісний із багатьма платформами і може бути встановлений на Windows, Linux, OSX або в контейнері Docker. Завдяки попередньо налаштованим установочним образам користувач може швидко налаштувати систему без зайвих зусиль.

Основні можливості ioBroker:

- Модульність та кросплатформеність: ioBroker підтримує інтеграцію з різними системами і забезпечує синергію між ними.

- Інтерфейси адаптерів: Платформа підтримує понад 200 різних пристроїв і платформ, що дозволяє легко налаштувати будинок за допомогою адаптерів, які підключають нові пристрої.

- Хмарний доступ: ioBroker може бути доступним 24/7 через хмарний сервіс, що дозволяє керувати системою з будь-якого місця.

- Розширення функцій: Користувачі можуть підключати нові пристрої та сервіси за допомогою додаткових адаптерів, що дозволяє системі розвиватися відповідно до вимог користувача.

- Масштабованість: Для великих систем можливе підключення кількох серверів ioBroker в систему Mutihost, а також інтеграція з потужними серверами для забезпечення високої продуктивності.

- Redis: Для високопродуктивних систем є можливість інтеграції з базою даних Redis, що забезпечує швидке збереження та обробку даних.

- Програмування та сценарії: Для реалізації додаткових функцій використовується мова програмування JavaScript, що дозволяє швидко реагувати на нові вимоги. Для початківців є також інтерфейс Blockly, який дозволяє створювати сценарії через перетягування блоків без глибоких знань програмування.

- Візуалізація даних: ioBroker надає потужні інструменти для створення індивідуальних візуалізацій, зокрема для відображення значень датчиків, історії процесів та зображень з камер спостереження. Візуалізація може включати контроль систем опалення, сигналізації та кондиціонування.

ioBroker дозволяє створювати інтегровані, масштабовані та персоналізовані рішення для автоматизації будинку та інших сфер, що робить його однією з найпопулярніших платформ для розумних будинків.

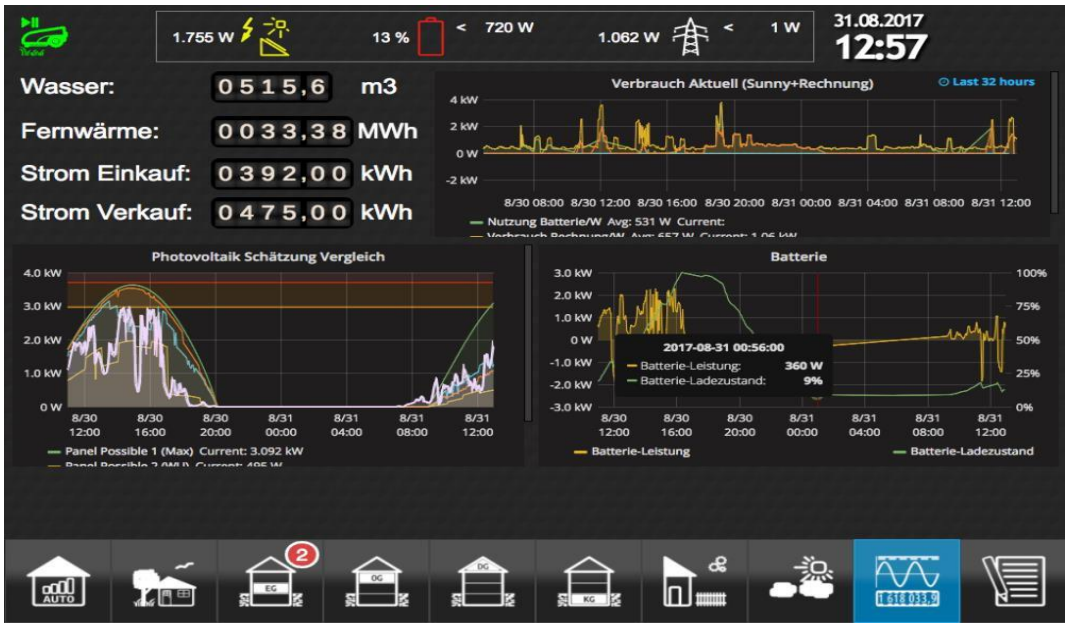


Рисунок 1.9 – Один із прикладів інтерфейсу додатку ioBroker.

ioBroker забезпечує максимальну свободу для користувачів при створенні інтерфейсів. Платформа пропонує готові модулі, які полегшують процес проектування та адаптації інтерфейсу до різних пристроїв. Це дозволяє користувачам створювати зручні та функціональні панелі управління для своєї системи автоматизації.

Однією з основних переваг є можливість керувати пристроями не тільки через відображення інформації, але й безпосередньо через інтерфейс візуалізації. Такий інтерфейс дозволяє швидко керувати різними елементами системи, такими як освітлення, опалення, сигналізація та інші пристрої [5].

Особливості інтерфейсу ioBroker:

- Адаптивний дизайн: Інтерфейси автоматично адаптуються до різних пристроїв, від смартфонів і настінних планшетів до персональних комп'ютерів, що дозволяє отримати зручний доступ до системи на будь-якому етапі.

- Простота використання: За допомогою простого перетягування елементів користувачі можуть легко створювати інтерфейси, що відповідають їхнім потребам.

- Інтерфейси для різних пристроїв: ioBroker підтримує адаптери для матеріалів, що дозволяють швидко створювати інтерфейси для різних типів пристроїв, а також використання HabPanel для більш складних візуалізацій.

Ці функції дозволяють користувачам без складних налаштувань створювати інтуїтивно зрозумілі та функціональні інтерфейси, що робить систему ioBroker доступною для широкого кола користувачів, незалежно від їхнього рівня технічних знань.

Рисунок 1.10 – Один із видів інтерфейсу, створених за допомогою ioBroker.

ioBroker – це гнучка програмна платформа для інтеграції різноманітних систем Інтернету речей (IoT) в єдину екосистему. Однією з основних особливостей ioBroker є модульна структура, яка дозволяє користувачам підключати різні пристрої та сервіси через адаптери (ioBroker Adapters). Система підтримує понад 250 адаптерів, що дають можливість інтеграції з різними типами обладнання, а також з отриманням інформації з різних джерел, таких як прогнози погоди, календарі, фінансові сервіси та інші.

Ключові характеристики ioBroker:

- Модульність і адаптери: Ви можете встановити тільки ті адаптери, які відповідають вашим потребам. Це дозволяє оптимізувати використання пам'яті та обчислювальних потужностей.

- Екземпляри адаптерів: Для кожного адаптера можна створювати "екземпляри", що дозволяє відрізняти різні підсистеми або вирішувати різні завдання в межах однієї платформи.

- Розподілені системи: Завдання можуть бути розподілені на кілька серверів, що дозволяє створювати многоузлові (multi-host) системи. Це корисно для масштабування системи або розподілу навантаження в залежності від просторових вимог або потреб у потужностях.

Переваги використання ioBroker:

1. Гнучкість і масштабованість: Платформа може працювати як на окремому сервері, так і в мережі з кількох серверів, що дозволяє забезпечити високу надійність і продуктивність.

2. Індивідуальна настройка: Користувач може налаштувати систему під свої індивідуальні вимоги, вибираючи необхідні адаптери і конфігуруючи їх для конкретних завдань.

3. Легкість в налаштуванні: Завдяки модульній структурі і великій кількості готових адаптерів, користувачі можуть легко інтегрувати нові пристрої або сервіси без необхідності переписувати великий обсяг коду.

Ця платформа ідеально підходить для тих, хто хоче створити свою унікальну систему автоматизації будинку, що інтегрує різні технології та пристрої в одному місці.

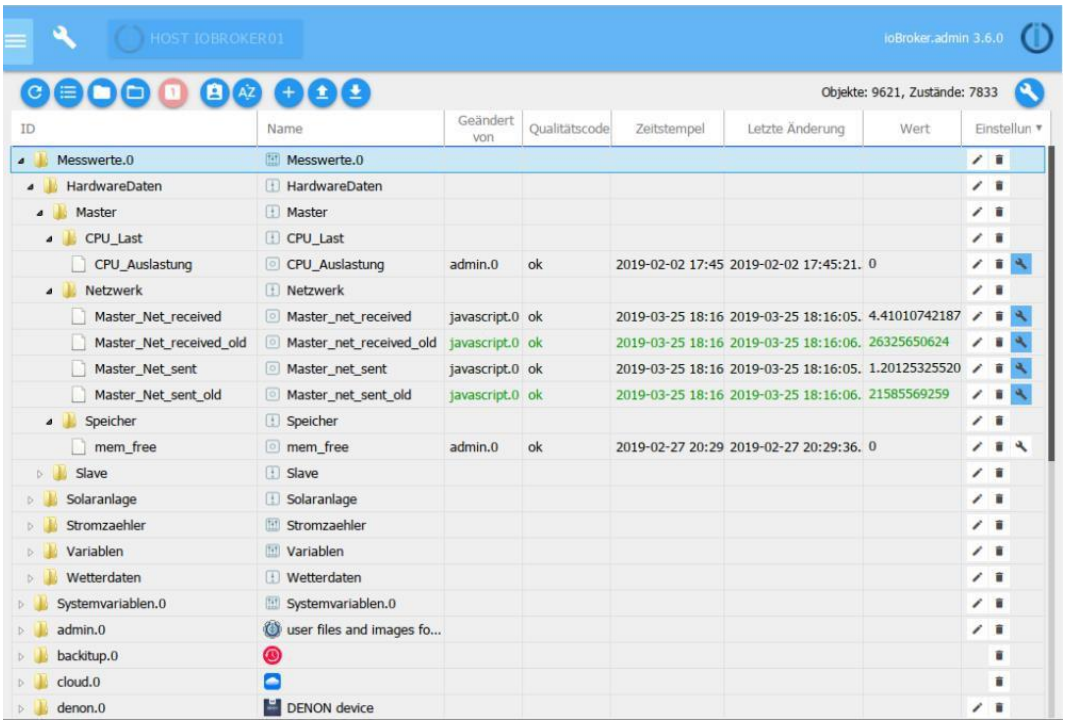


Рисунок 1.11 – Панель адміністратора ioBroker.

Вимоги до обладнання.Сервер ioBroker має високу сумісність з різними типами обладнання, завдяки чому його можна встановити практично на будь-якому пристрої. Однак є кілька мінімальних вимог:

- Операційна система: Потрібна актуальна версія Node.js для відповідної операційної системи.

- Оперативна пам'ять (ОЗУ): Для більш великих і складних установок рекомендується мати не менше 2 ГБ ОЗУ. Для базових налаштувань або тестування цілком достатньо пристроїв, таких як Raspberry Pi 2/3 з 1 ГБ ОЗУ, особливо в умовах використання міні-комп'ютерів у середовищах з кількома хостами.

Це дає змогу зберігати достатній рівень продуктивності навіть на менш потужних пристроях при налаштуванні або тестуванні базових функцій ioBroker.

ioBroker використовує базу даних для управління даними, що дозволяє організувати структуру даних для ефективного зберігання і обробки інформації.

Платформа підтримує інтеграцію з різними системами управління даними і дає змогу зберігати значні обсяги даних для автоматизації та моніторингу розумного будинку чи IoT-системи.

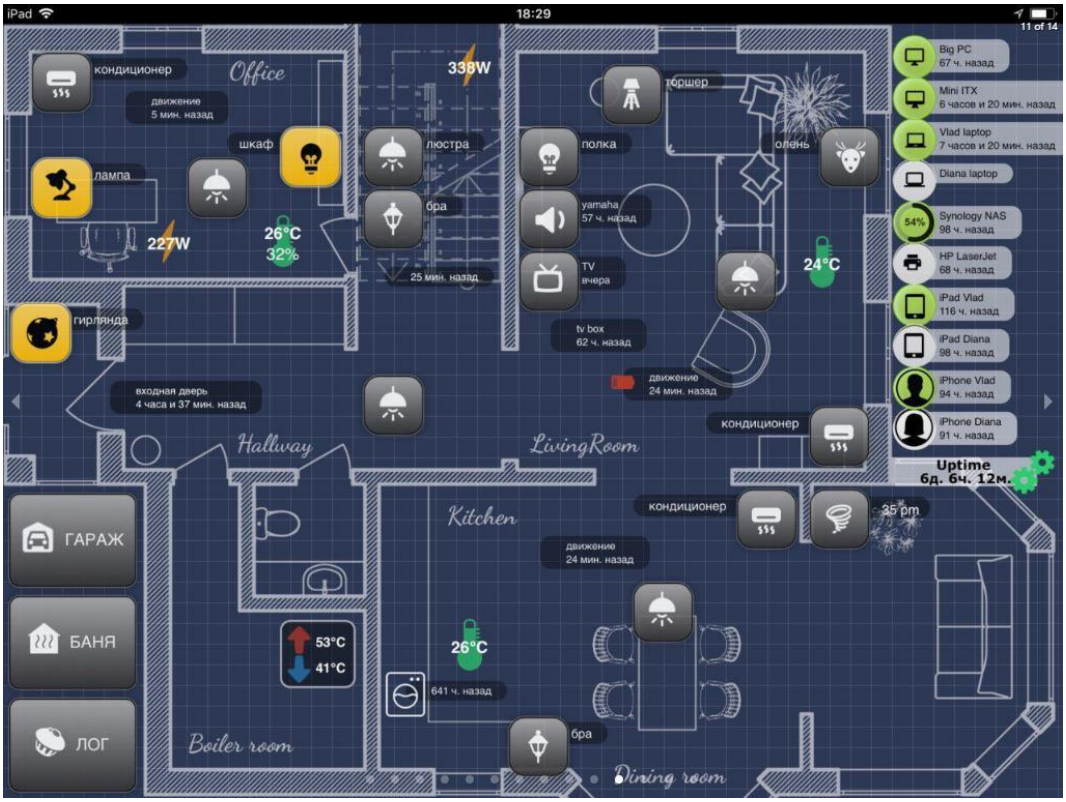


Рисунок 1.12 – Приклад інтерфейсу додатку ioBroker.

Кожен адаптер у системі ioBroker має так зване простір імен, яке містить усі дані про екземпляр адаптера. Простір імен зазвичай має вигляд, наприклад: AdapterName.0, де AdapterName — це ім'я адаптера, а 0 — індекс або версія екземпляра [6].

У межах цього простору імен ioBroker створює пристрої, їх канали та точки даних з відповідними значеннями (станами). Це дозволяє організувати структуру даних і керувати різними підсистемами та їх параметрами. Для зручності адміністрування та налаштування можна створювати кілька таких просторів імен, кожен з яких відповідає окремому підключеному пристрою або групі пристроїв.

Цей приклад показує, як може виглядати самостійно створений простір імен для ваших власних метрик. Наприклад, якщо ви використовуєте датчики температури або системи освітлення, то створений простір імен дозволяє керувати даними таких пристроїв, відстежувати їх стан та інтегрувати їх у загальну систему автоматизації будинку.

ioBroker дозволяє також реалізувати унікальні функції адаптерів за допомогою мови програмування JavaScript. Для новачків доступна опція використання Blockly, яка дозволяє створювати сценарії за допомогою перетягування блоків, що значно полегшує процес налаштування і автоматизації.

Система автоматизації Domoticz**.** Domoticz – це легка система домашньої автоматизації, що дозволяє ефективно контролювати та налаштовувати різноманітні пристрої в рамках розумного будинку. Завдяки своїй гнучкості, система підтримує управління такими пристроями, як ліхтарі, вимикачі, датчики температури, кількості опадів, сили вітру, ультрафіолетового (УФ) випромінювання, а також пристрої для моніторингу використання енергії, води та газу.

Однією з ключових функцій Domoticz є можливість налаштування сповіщень і попереджень, які можуть бути надіслані на мобільні пристрої користувача в разі виникнення важливих подій. Наприклад, якщо система фіксує надмірне споживання електроенергії або нестандартну температуру в приміщенні, Domoticz може миттєво повідомити користувача через мобільний додаток, що дозволяє оперативно реагувати на ситуацію.

Domoticz підтримує безліч протоколів і пристроїв, що дозволяє інтегрувати в систему різноманітні компоненти автоматизації. Ця система є відкритою та безкоштовною, що робить її популярним вибором серед любителів домашньої автоматизації та користувачів, які прагнуть створити власну розумну екосистему для дому.

Додаткові особливості:

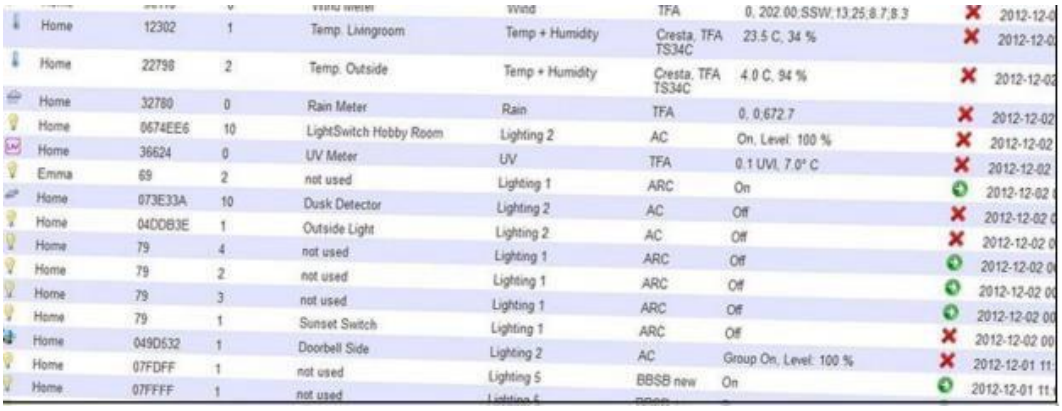
- Підтримка численних датчиків та лічильників (наприклад, для моніторингу вологості, рівня CO2, руху тощо).

- Інтеграція з різноманітними платформами IoT, такими як Z-Wave, Zigbee, MQTT тощо.

- Простий інтерфейс користувача, що дозволяє швидко налаштовувати та змінювати параметри пристроїв.

- Можливість налаштування сценаріїв автоматизації для різних ситуацій.

Система надає можливість для створення потужних рішень для управління інфраструктурою будинку та енергозбереження, що робить її важливим елементом концепцій "розумних будинків".



				Wind	TFA	0, 202.00;SSW;13.25;8.7;8.3	2012-12-0
Home	12302	1	Temp. Livingroom	Temp + Humidity	Cresta, TFA TS34C	23.5 C, 34 %	2012-12-0
Home	22798	2	Temp. Outside	Temp + Humidity	Cresta, TFA TS34C	4.0 C, 94 %	2012-12-02
Home	32780	0	Rain Meter	Rain	TFA	0, 0.672.7	2012-12-02
Home	0674EE6	10	LightSwitch Hobby Room	Lighting 2	AC	On, Level: 100 %	2012-12-02
Home	36624	0	UV Meter	UV	TFA	0.1 UVI, 7.0° C	2012-12-02
Emma	69	2	not used	Lighting 1	ARC	On	2012-12-02
Home	073E33A	10	Dusk Detector	Lighting 2	AC	Off	2012-12-02
Home	04D0B3E	1	Outside Light	Lighting 2	AC	Off	2012-12-02
Home	79	4	not used	Lighting 1	ARC	Off	2012-12-02
Home	79	2	not used	Lighting 1	ARC	Off	2012-12-02
Home	79	3	not used	Lighting 1	ARC	Off	2012-12-02
Home	79	1	Sunset Switch	Lighting 1	ARC	Off	2012-12-02
Home	049D532	1	Doorbell Side	Lighting 2	AC	Group On, Level: 100 %	2012-12-01
Home	07FDFF	1	not used	Lighting 5	BBSB new	On	2012-12-01
Home	07FFFF	1	not used	Lighting 5			

Рисунок 1.13 – Інтерфейс додатку для редактора пристроїв Domoticz

Domoticz використовує бібліотеку з відкритим кодом "OpenZWave" (скорочено OZW) для інтеграції пристроїв, що підтримують протокол Z-Wave. Протокол Z-Wave є популярним у сфері домашньої автоматизації, однак він не є повністю безкоштовним, і тому не всі пристрої чи функції підтримуються в OpenZWave. Це може обмежувати сумісність певних пристроїв із цією системою.

Більшість пристроїв, таких як Fibaro, BeNext та Aeon Labs, повністю підтримуються і працюють з OpenZWave. Однак деякі пристрої, наприклад

Fortrezz та інші менш відомі моделі, можуть бути лише частково підтримувані або не працювати зовсім.

Щоб забезпечити сумісність з Domoticz, спочатку необхідно перевірити, чи підтримуються ваші пристрої Z-Wave в бібліотеці OpenZWave. Якщо пристрої не відомі в базі даних OpenZWave, існує можливість додати нові пристрої шляхом включення їх у базу даних за допомогою спеціальних інструкцій або налаштувань.

Цей процес включає в себе:

- Перевірку сумісності пристрою з OpenZWave.

- Додавання пристрою до бази даних OpenZWave.

- Використання відповідних інтерфейсів для налаштування та контролю пристроїв через Domoticz.

Таким чином, протокол Z-Wave дозволяє інтегрувати в Domoticz різноманітні пристрої для автоматизації дому, але важливо враховувати сумісність і можливість додавання нових пристроїв.

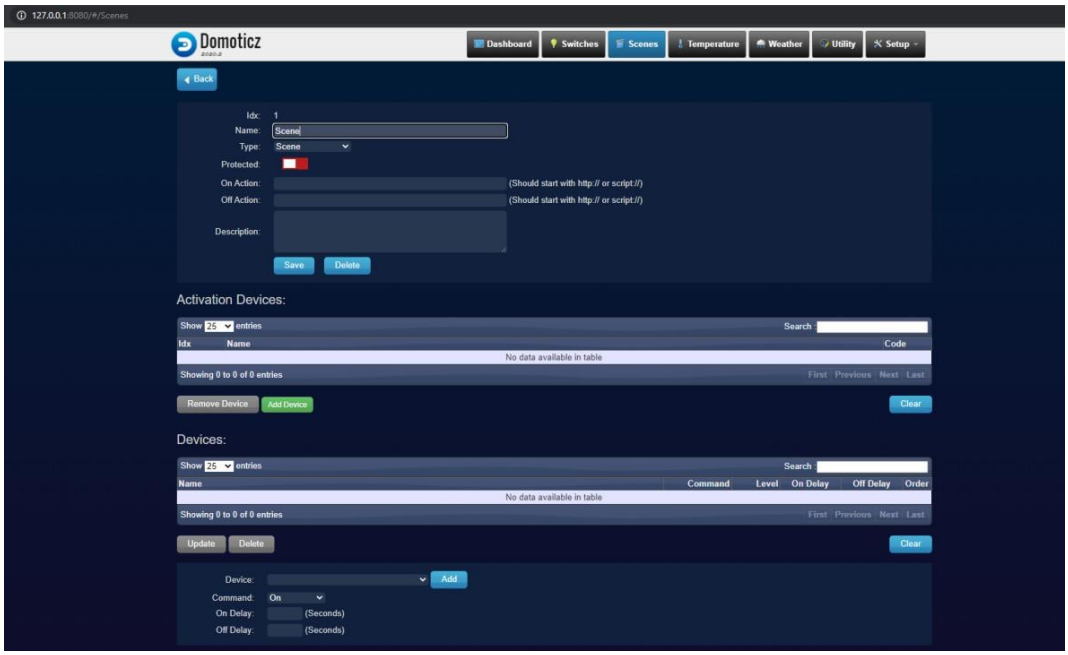


Рисунок 1.14 – Інтерфейс редактора сценаріїв додатку Domoticz

Екран інформації про мережу Z-Wave можна знайти в розділі Налаштування > Апаратне забезпечення в системі Domoticz. Якщо в системі присутній Z-Wave контролер (наприклад, USB-накопичувач), поряд з ним буде кнопка "Налаштування". Клацнувши на неї, відкриється список всіх підключених вузлів. У верхньому правому куті екрана знаходиться кнопка "Групи та мережа", яка дозволяє переглядати деталі мережі [7].

Зліва відображається хордова діаграма, яка наочно показує, які вузли можуть взаємодіяти між собою. Важливою рисою сітчастої мережі Z-Wave є те, що не всі пристрої мають пряме з'єднання між собою, але можуть комунікувати через проміжні вузли. Ця діаграма допомагає зрозуміти, як функціонує мережа в цілому. Якщо Z-Wave мережевий контролер нещодавно перезавантажувався або було запущено Domoticz, на оновлення з'єднань може знадобитися певний час. Тому після перезавантаження системи варто зачекати декілька хвилин перед поверненням до цього екрану. Клацнувши на вузол, можна побачити лише з'єднання, що стосуються саме цього вузла, що полегшує аналіз взаємодії між конкретними пристроями.

У правій частині екрана розміщена таблиця груп, що показує всі зареєстровані групи для кожного з вузлів і переліки пристроїв, які належать до цих груп. Групи Z-Wave є ключовим елементом роботи системи, оскільки вони дозволяють забезпечити взаємодію пристроїв навіть у разі відсутності активного Domoticz чи контролера. Варто зазначити, що групи не є універсальними, а існують для кожного вузла окремо. Для кожного пристрою можна створити одну або кілька груп, до яких можна додавати інші пристрої як слухачів. Якщо два пристрої мають однакову групу, це не означає, що вони належать до єдиної глобальної групи. Для видалення пристрою з групи необхідно натиснути на номер вузла. Якщо група позначена символом плюса, це свідчить про те, що вона активна для цього вузла. Клацнувши на цей символ, можна додати новий пристрій до групи.

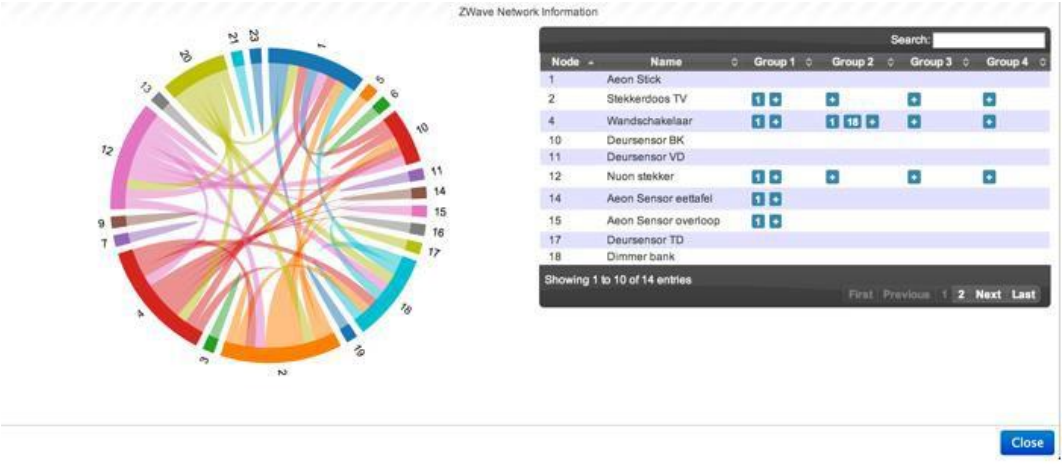


Рисунок 1.15 – Інформація про мережу та групи Domoticz

Приклад: настінний вимикач tz66-d оснащений двома лопатями (лівою та правою). Ліва лопать фізично керує лампою, яка підключена до нього, а права лопать лише відповідає за сигналізацію в межах груп. У tz66-d є чотири групи:

- Група 1: отримує сигнал при натисканні на ліву лопать.

- Група 2: отримує сигнал при натисканні на праву лопать.

- Група 3: отримує сигнал при подвійнім натисканні на праву лопать.

- Група 4: отримує сигнал, коли tz66-d отримує команду на включення або вимкнення.

Завдяки таблиці груп можна додавати інші пристрої (наприклад, лампи) в одну з груп tz66d. Наприклад, поставивши лампу в Групу 2, можна керувати нею за допомогою правої лопаті вимикача. Якщо лампу помістити в Групу 3, її можна вмикати подвійним натисканням правої лопати.

Групи відіграють важливу роль не тільки у керуванні пристроями, але й у зборі даних з вимірювальних приладів, таких як термометри. Наприклад, пристрої на зразок Aeon 4in1 можуть бути налаштовані на відправку даних в Групу 1 (де знаходиться контролер, а отже, Domoticz). Інші пристрої можуть надсилати свої дані в певну, фіксовану групу, до якої потрібно додавати приймаючий пристрій.

Інтерфейс цього екрану з часом буде вдосконалюватися і доповнюватися новими функціями. На даний момент він містить лише найосновніше. Наприклад, управління сценами наразі не реалізоване, оскільки це можна робити через саму платформу Domoticz.

1.3 Постановка задачі дослідження

Основними задачами магістерської дисертації є:

- Вивчення актуальності та особливостей впровадження автоматизації в управління виробничими системами, зокрема в умовах гнучких виробничих середовищ, що постійно змінюються. Це включає дослідження існуючих підходів до автоматизації і адаптації до нових вимог, таких як інтеграція Інтернету речей (IoT) та штучного інтелекту (ШІ) для покращення гнучкості та ефективності.

- Аналіз ефективності автоматизації в гнучких виробничих системах, що передбачає оцінку впливу автоматизованих рішень на підвищення продуктивності, зниження витрат та покращення якості продукції. Цей етап також включає вивчення результативності різних рівнів автоматизації, від часткової автоматизації до повного керування виробничими процесами.

- Дослідження підходів до проектування автоматизованих систем, зокрема тих, що використовують сучасні програмно-апаратні засоби. У цьому контексті важливо враховувати специфіку підприємства та типу виробництва, а також вимоги до зручності та масштабованості системи.

- Аналіз структури автоматизованих систем з точки зору організації взаємодії між різними елементами системи: від датчиків і виконавчих механізмів до інтеграції з управлінськими платформами та системами моніторингу.

- Дослідження можливостей інтеграції штучного інтелекту в управлінські системи для автоматичного аналізу та оптимізації виробничих процесів. Це дозволить системі самостійно навчатися, адаптуватися до змін у виробничому середовищі і пропонувати оптимальні варіанти для прийняття рішень.

- Аналіз гнучких виробничих систем, орієнтуючись на їх здатність адаптуватися до змін в умовах виробництва, вимогах замовників, а також до впливу технологічних змін і зовнішніх факторів.

- Розробка програмного додатку, який відповідатиме вимогам сучасних систем автоматизації. Для цього необхідно створити ефективний і зручний графічний інтерфейс, який буде інтегрувати всі ключові функції системи, забезпечувати зручне керування і візуалізацію процесів.

Програмне середовище має відповідати наступним вимогам:

- Ініціативність користувача: Усі основні дії мають ініціюватися користувачем, що дозволяє уникнути непотрібної автоматизації та забезпечити максимальний контроль з боку оператора.

- Швидка реакція на команди: Програма повинна миттєво реагувати на введені команди, зменшуючи час на виконання операцій і покращуючи досвід користувача [8].

- Інтерактивність: Система повинна бути інтерактивною, щоб користувач міг змінювати параметри в реальному часі, отримувати зворотний зв'язок і вносити корективи без перерви у роботі.

- Наочність операцій: Всі дії та операції мають бути подані на графічному інтерфейсі таким чином, щоб користувач міг легко орієнтуватися та зрозуміти, які операції виконуються на даний момент.

- Самостійне введення даних: Користувач має можливість вводити власні назви операцій, що дозволяє адаптувати систему під конкретні виробничі вимоги та забезпечити зручність налаштування.

- Дружній інтерфейс: Інтерфейс має бути зрозумілим і доступним для користувачів з різним рівнем підготовки, з можливістю швидкого освоєння.

- Запобігання випадковим діям: Система повинна містити захист від випадкових помилок з боку користувача, таких як підтвердження операцій чи попередження перед важливими діями.

- Моделювання в 3D: Необхідність наявності функції для моделювання виробничих процесів у тривимірному просторі, що допоможе у візуалізації складних процесів і прийнятті рішень.

- Контроль доступу до об'єктів: Вибір об'єкта для взаємодії має відкривати відповідні інструменти управління і дозволяти змінювати його параметри, що забезпечить точність і контроль над процесами.

- Інформаційна наглядність: Важлива інформація повинна бути наочно подана на екрані, зокрема в момент маніпулювання об'єктами.

- Уніфікація методів роботи: Всі методи роботи з системою повинні бути однаковими і стандартизованими, що полегшить взаємодію з користувачем.

- Консистентність стилів інтерфейсу: Стилі та кольори інтерфейсу повинні бути узгоджені між усіма елементами системи для полегшення її використання.

- Збереження налаштувань: Система повинна підтримувати функцію збереження та завантаження налаштувань, що дозволить користувачеві зберігати персоналізовані конфігурації.

- Зворотний зв'язок: Інтерфейс має забезпечувати зворотну інформацію щодо ходу процесів, щоб користувач міг відслідковувати поточний стан системи.

- Легкість освоєння: Користувач повинен мати можливість швидко освоїти систему та почати працювати з нею, що важливо для ефективного застосування в умовах виробництва.

- Баланс між простотою та доступністю функцій: Важливо знайти баланс між простотою використання інтерфейсу і забезпеченням доступу до всіх необхідних функцій і даних.

У розділі розглянуто предмет і об'єкт дослідження, а також проаналізовано існуючі підходи до створення систем автоматизації та візуалізації інтерфейсу користувача. Детально описано процес створення інтерфейсу за допомогою серверної частини на базі Apache та управлінських модулів, зокрема на платформі Arduino. Оцінено чотири платформи для автоматизації, які підтримують протокол

Z-Wave: zVirtualScenes, Ago Control, ioBroker та Domoticz. Після порівняння переваг і недоліків кожної платформи було обрано ioBroker, яка пропонує широкий спектр рішень для синхронізації та інтеграції різних протоколів, підтримує активну розробку адаптерів та надає потужні інструменти для візуалізації інтерфейсу користувача.

2 АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз безпеки використання протоколу розумного будинку

Завдяки стрімкому технологічному прогресу, міста стають розумнішими. Складові розумного міста включають розумний спосіб життя, розумне середовище, розумну мобільність, розумну економіку, розумне управління та розумних людей. Інфраструктура, пов'язана з цими розумними технологіями, відома як Інтернет речей (IoT), Інтернет даних (IoD), Інтернет послуг (IoS) та Інтернет людей (IoP). У цьому контексті технології IoT мають важливе значення для розвитку розумних міст, оскільки забезпечують інтеграцію фізичних об'єктів із цифровими платформами через бездротові мережі.

Однак із поширенням технологій IoT виникають нові виклики, зокрема в сфері безпеки. Оскільки IoT стає звичною технологією, особлива увага приділяється цифровій безпеці. Технологія IoP, яка дозволяє взаємодіяти людей із цифровими системами через пристрої IoT, постала в центрі досліджень, оскільки в розумних містах особиста інформація збирається за допомогою численних машин, що призводить до необхідності забезпечення персоналізованого обслуговування приватних осіб.

Проте існують серйозні проблеми щодо загрози безпеці особистої інформації. У зв'язку з великим обсягом даних, які збираються, важливо зберігати конфіденційність і цілісність цих даних, а також захищати їх від несанкціонованого доступу. Це стало особливо актуальним у розумних містах, де кожен розумний будинок виступає основним джерелом збору даних про особисте життя мешканців. Враховуючи постійне підключення до глобальних мереж, важливо знижувати ризики кіберзагроз.

Дослідження в галузі безпеки протоколів для розумних будинків зосереджуються на кількох ключових аспектах:

- Захист особистої інформації: Технології збору даних, що використовуються в розумних будинках, повинні відповідати стандартам конфіденційності і безпеки, аби захистити приватність користувачів. Дані, що збираються розумними пристроями, можуть включати інформацію про місце розташування, побутові звички, використання електроенергії, температуру та інші аспекти, що дозволяють створювати точні профілі користувачів [9].

- Аутентифікація та авторизація: Розумні будинки потребують надійних механізмів аутентифікації, які можуть включати біометричні дані або багатоетапні системи перевірки доступу для забезпечення безпеки. Важливою частиною є також забезпечення належного рівня авторизації, щоб уникнути несанкціонованого доступу до конфіденційних даних або управління розумними пристроями.

- Шифрування та захист каналів зв'язку: Використання шифрування даних є критично важливим для захисту інформації під час передачі через відкриті або слабозахищені мережі. Протоколи зв'язку, що використовуються для обміну даними між пристроями розумного будинку, повинні підтримувати стандарти шифрування, щоб забезпечити цілісність та конфіденційність даних.

- Захист від атак на мережу: Розумні будинки, підключені до Інтернету, піддаються різним типам кіберзагроз, зокрема атакам DDoS (розподілене відмовлення в обслуговуванні), спробам несанкціонованого доступу до мережевих пристроїв, а також маніпулювання даними в системах управління. Для захисту необхідно впроваджувати ефективні засоби моніторингу та реагування на загрози.

- Інтеграція стандартів безпеки: Для забезпечення безпеки розумних будинків важливо інтегрувати перевірені стандарти безпеки на рівні протоколів зв'язку та управління пристроями. Це може включати використання популярних стандартів, таких як Zigbee, Z-Wave або Thread, які пропонують засоби для забезпечення безпечної передачі даних між пристроями.

Загалом, забезпечення безпеки в контексті протоколів розумних будинків є багатогранним завданням, яке включає в себе захист особистої інформації, надійні механізми аутентифікації, шифрування, а також заходи протидії атакам на мережу. Інтеграція сучасних технологій та підвищена увага до стандартів кібербезпеки є необхідними для створення ефективної та безпечної інфраструктури розумних будинків у розумних містах.

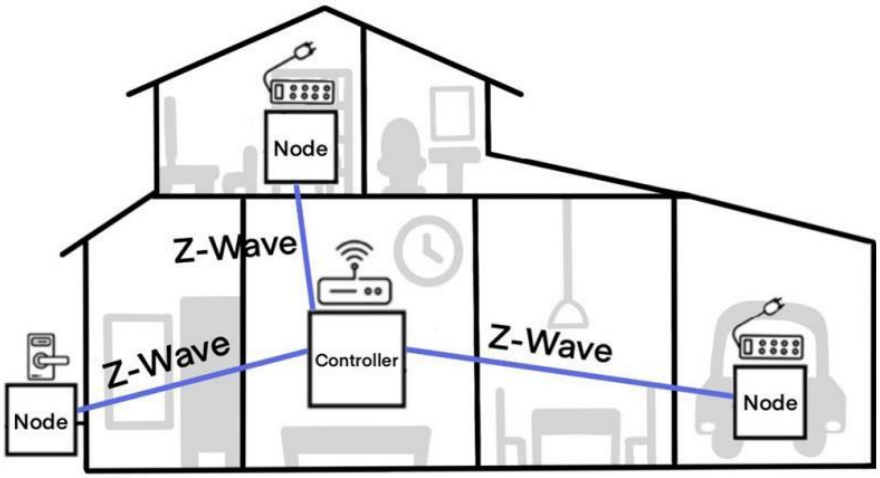


Рисунок 2.1 - Розумний будинок із Z-Wave

У сучасних розумних будинках різноманітні пристрої активно інтегруються для забезпечення комфортного середовища для мешканців. За допомогою мобільних телефонів та спеціальних контролерів користувачі можуть віддалено керувати елементами інфраструктури будинку, такими як газові клапани, освітлення або замки дверей. З метою забезпечення зручності та безпеки застосовуються різні бездротові технології, серед яких Wi-Fi, ZigBee та Z-Wave. Ці технології дозволяють створювати ефективні сенсорні мережі, що взаємодіють між пристроями та контролерами, що знаходяться в межах одного будинку. Wi-Fi та LTE забезпечують широкомасштабний зв'язок між пристроями та контролерами, тоді як ZigBee та Z-Wave, які призначені для короткодіапазонного

зв'язку, є основними технологіями для забезпечення обміну даними між пристроями в розумному будинку.

Z-Wave, розроблений компанією Zensys у 2001 році, став одним із провідних стандартів для бездротових пристроїв у розумних будинках. Ця технологія підтримується великим альянсом, що забезпечує сертифікацію пристроїв, що працюють на основі Z-Wave. Однак, незважаючи на широке застосування технології, дослідження безпеки Z-Wave залишаються обмеженими через приватний характер протоколу. Тому особливу увагу слід приділяти аналізу безпеки цього протоколу, адже він є основним засобом зв'язку в багатьох розумних будинках.

Як показано на рисунку 2.1, протокол Z-Wave складається з основного контролера, який приймає команди від користувачів через мобільні пристрої. Контролер надсилає ці команди до кожного вузла в мережі, який виконує необхідні дії, наприклад, включення освітлення або регулювання температури. Така архітектура робить систему зручною в управлінні, але також підвищує ризик можливих атак на мережу. Тому дослідження щодо безпеки Z-Wave є надзвичайно актуальними.

Для оцінки безпеки протоколу Z-Wave було проведено серію експериментів, які включали моделювання можливих сценаріїв атак. Метою цих експериментів було визначити потенційні загрози, пов'язані з використанням даного протоколу в реальних умовах. Особлива увага була приділена процесу обміну ключами та з’єднань між контролером і вузлами. Виявилось, що процес початкового обміну мережевими ключами між пристроями може бути вразливим для атак, що ставить під загрозу безпеку користувачів.

Для виявлення загроз було застосовано модель STRIDE, яка є однією з найбільш ефективних для виявлення та класифікації потенційних уразливостей в інформаційних системах. В результаті аналізу було визначено понад 40 потенційних загроз, серед яких особливу увагу слід приділити атакам типу

«відмова в обслуговуванні» (DoS), «прошивка в ефірі» (FOTA) та віддаленій атаці на керування режимом додаткового режиму. Ці атаки можуть призвести до серйозних наслідків, таких як порушення роботи важливих функцій розумного будинку, зокрема системи безпеки або енергоспоживання.

За допомогою методології STRIDE було здійснено детальний аналіз загроз на рівні 2 у DFD (діаграма потоку даних) для пристроїв Z-Wave. Виявлені загрози в результаті проведених експериментів показали, що сучасні розумні будинки, оснащені пристроями на основі Z-Wave, є вразливими до різноманітних атак. Наприклад, можливість перехоплення та модифікації команд між контролером і вузлами дозволяє зловмисникам отримати доступ до конфіденційних даних або навіть маніпулювати елементами управління в будинку.

Таким чином, хоча технології, такі як Z-Wave, значно покращують комфорт і ефективність розумних будинків, вони також створюють нові загрози для безпеки. Важливо розробляти та впроваджувати додаткові методи захисту і шифрування для захисту персональних даних і запобігання атакам на критичні системи розумного будинку. Це особливо актуально в умовах стрімкого розвитку Інтернету речей і зростання кількості підключених пристроїв, що збільшує потенційні ризики для користувачів.

Протокол Z-Wave став однією з найпопулярніших технологій для створення бездротових розумних мереж в розумних будинках, що дозволяють автоматизувати багато аспектів життєдіяльності будинку. Він використовує спеціалізовану технологію зв’язку для малих пристроїв, таких як сенсори, світильники, замки, термостати, детектори руху, і навіть побутову техніку. Z-Wave є частиною екосистеми Інтернету речей (IoT), в якій всі ці пристрої взаємодіють між собою для створення інтелектуальної системи управління будинком.

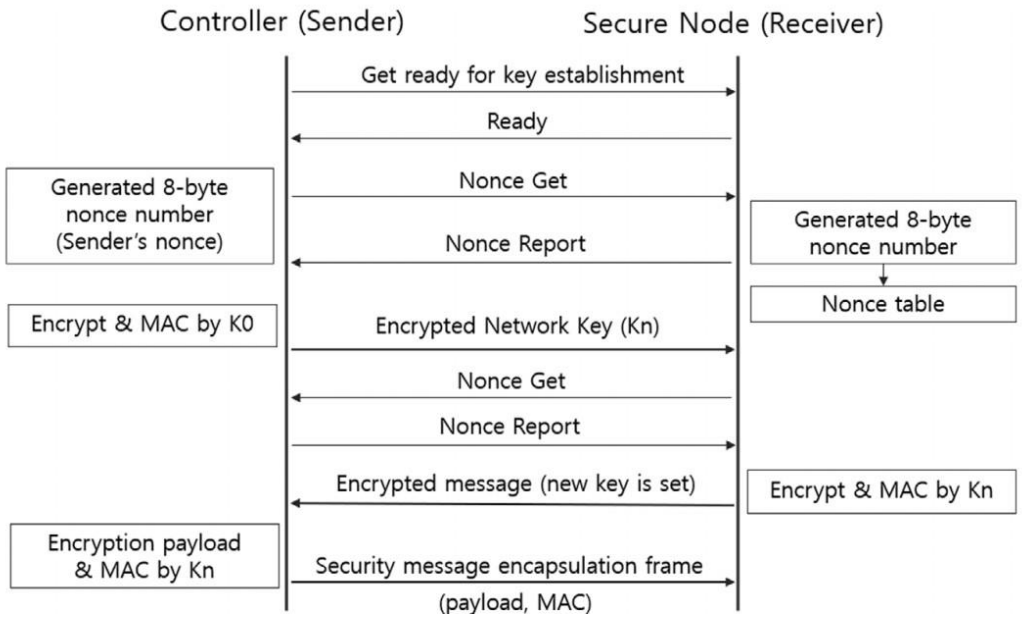


Рисунок 2.2 – Мережевий зв'язок Z-Wave

Протокол Z-Wave базується на чотирьох основних рівнях, кожен з яких відіграє важливу роль у забезпеченні ефективної роботи мережі:

1. Рівень додатків — цей рівень відповідає за роботу з даними, які передаються або приймаються між пристроями. Він керує розпізнаванням та обробкою команд, наприклад, для активації сенсорів або вмикання/вимикання освітлення. На цьому рівні також здійснюється обробка корисного навантаження в кадрі.

2. Мережевий рівень — цей рівень займається безпосередньо маршрутизацією даних між різними вузлами мережі. Він забезпечує виявлення шляхів до пристроїв, які знаходяться в межах мережі, а також підтримує таблиці маршрутизації для оптимізації передачі даних. Це дозволяє зберігати швидкість і ефективність передачі сигналів навіть у великих мережах, де кількість пристроїв значно зростає [10].

3. Рівень MAC / рівень передачі — цей рівень відповідає за фізичну передачу даних між вузлами через бездротовий зв'язок. Він здійснює перевірку

цілісності кадрів, підтверджує їх доставку та проводить повторні передачі в разі виникнення помилок. Це необхідно для забезпечення надійної комунікації між пристроями, навіть коли на шляху сигналу можуть бути перешкоди.

4. Фізичний рівень — на цьому рівні здійснюється безпосередня передача сигналу через радіочастотний канал (RF). Це включає використання різних частот для передачі і прийому даних, з урахуванням уникнення перешкод. Механізм модулювання та доступу до каналу забезпечує ефективне використання спектру радіочастот для зв'язку.

Протокол Z-Wave працює на частотах, визначених міжнародними стандартами, і використовує низькошвидкісні радіохвилі для передачі даних на відносно невеликі відстані, що робить його ідеальним для створення мереж автоматизації в будинках.

Хоча протокол Z-Wave спочатку не мав засобів шифрування та автентифікації, з плином часу він був адаптований для забезпечення більш високого рівня безпеки. Проблема безпеки була значною, оскільки у незахищеному режимі всі дані передавалися у вигляді простого тексту, що могло дозволити зловмисникам перехоплювати або маніпулювати інформацією.

Щоб вирішити ці проблеми, у Z-Wave була введена безпека рівня S0, яка забезпечує шифрування та автентифікацію повідомлень, що передаються між контролером та пристроями. У цьому режимі перед передачею даних здійснюється автентифікація вузлів мережі, і самі дані шифруються для захисту від несанкціонованого доступу. Це дозволяє зменшити ризики, пов'язані з можливими атаками на мережу, такими як:

- Перехоплення даних: Завдяки шифруванню зловмисники не можуть отримати доступ до приватної інформації.

- Атаки на пристрої: Використання автентифікації запобігає підключенню несанкціонованих пристроїв до мережі.

- Маніпуляції з командами: Аутентифікація та шифрування гарантують, що команди, що передаються через мережу, не можуть бути змінені або підроблені.

З появою режиму безпеки S2, цей протокол став ще більш захищеним, адже введено підтримку більш потужних алгоритмів шифрування, таких як AES-128, що забезпечують ще вищий рівень захисту даних.

Попри всі досягнення у сфері безпеки, Z-Wave все ще стикається з кількома проблемами. Ось деякі з них:

1. Вразливість до атак на фізичному рівні: Оскільки Z-Wave використовує бездротовий зв'язок, дані можуть бути перехоплені зловмисниками, що знаходяться в межах досяжності сигналу.

2. Слабка безпека деяких пристроїв: Багато старих пристроїв Z-Wave не підтримують нові протоколи безпеки або мають слабкі алгоритми шифрування, що робить їх вразливими до атак.

3. Захист від атак з боку користувачів: У разі неправильного налаштування або використання стандартних паролів, пристрої можуть стати мішенню для атак.

Протокол Z-Wave продовжує розвиватися, що дає можливість для більш безпечного і ефективного використання в розумних будинках та містах. Інтеграція з іншими технологіями, такими як IoT платформи, мобільні додатки та шлюзи для обміну даними, дозволяє розширювати функціональність протоколу, зберігаючи при цьому безпеку. Більше того, постійне вдосконалення методів шифрування та автентифікації дозволяє Z-Wave залишатися конкурентоспроможним у світі розумних технологій, де безпека є однією з основних вимог.

Враховуючи розвиток технологій та постійне зростання популярності розумних будинків, важливим є не лише функціональність пристроїв, але й гарантії безпеки даних, що передаються через мережу Z-Wave. Розуміння слабких місць цього протоколу і вдосконалення його безпеки є ключовим фактором для забезпечення надійності та захищеності розумних будинків у майбутньому.

2.1.1 Кадр Z-Wave

Кадр Z-Wave є основною одиницею передачі даних в мережі Z-Wave і складається з кількох важливих компонентів, що забезпечують правильну передачу і обробку інформації. Формат кадру організовано таким чином, щоб гарантувати точність і ефективність комунікації між різними пристроями в мережі. Кадр складається з заголовка, даних програми та циклічної перевірки надмірності (CRC), що дозволяє забезпечити цілісність і надійність передачі.

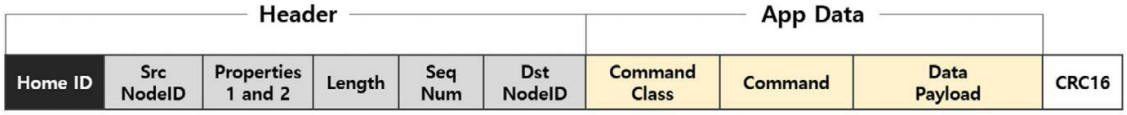


Рисунок 2.3 – Формат кадру Z-Wave

1. Заголовок кадру

Заголовок кадру є критично важливою частиною, яка дозволяє ідентифікувати та маршрутизувати повідомлення в мережі. Він містить кілька ключових елементів:

- Домашній ідентифікатор: Цей ідентифікатор використовується для ідентифікації персональної мережі (PAN) контролера. Кожен контролер в мережі має унікальний ідентифікатор, що дозволяє системі зрозуміти, від якого пристрою або мережі надходять дані [11].

- Вихідний ідентифікатор вузла: Це значення допомагає визначити, який саме вузол є відправником кадру. Це дозволяє мережі точно вказати джерело переданої інформації.

- Властивості 1 і 2: Ці властивості використовуються для додаткових налаштувань і характеристик кадру, які можуть змінюватися в залежності від конкретних вимог системи або додатка.

- Довжина кадру: Цей параметр вказує на загальну довжину кадру в байтах, що дозволяє приймаючому пристрою правильно інтерпретувати вхідні дані та визначити, коли кадр завершений.

- Порядковий номер: Це номер кадру, який забезпечує надійність зв'язку, особливо при передачі декількох кадрів. Він допомагає забезпечити правильний порядок отримання даних.

- Ідентифікатор вузла призначення: Цей параметр вказує на отримувача повідомлення в мережі. Завдяки цьому приймач може точно ідентифікувати пристрій, до якого надіслано повідомлення.

2. Дані програми

Ця частина кадру містить саму корисну інформацію, яка передається між пристроями. Дані програми поділяються на три основні елементи:

- Клас команд: Клас команд визначає групу команд, що відносяться до певної функціональності або типу пристроїв в мережі. Наприклад, клас команд може вказувати на управління освітленням або температурним контролем.

- Команда: Команда є конкретним набором інструкцій або запитів, який виконується певним пристроєм в рамках класу команд. Це може бути команда для включення/вимкнення пристрою, зміна налаштувань або інші дії.

- Корисне навантаження: Це параметри, що супроводжують команду і використовуються для детальнішого налаштування або виконання команди. Наприклад, це можуть бути значення для зміни яскравості лампи або температури термостата.

3. Циклічна перевірка надмірності (CRC)

Цей елемент кадру є частиною механізму перевірки на цілісність даних. CRC використовується для виявлення помилок у переданих даних, які могли виникнути внаслідок перешкод чи інших факторів під час передачі. При прийомі кадру пристрій перевіряє CRC, щоб переконатися, що передані дані не були змінені або пошкоджені.

Важливість і застосування кадрів Z-Wave

Кадри Z-Wave важливі для стабільної і надійної роботи мережі розумного будинку. Вони забезпечують точну маршрутизацію та виконання команд, зберігаючи при цьому цілісність і безпеку даних. Кожен кадр містить всю необхідну інформацію для ідентифікації пристроїв, виконання команд та забезпечення захищеності при передачі.

Завдяки своїй структурі, Z-Wave кадри можуть підтримувати складні мережі з багатьма пристроями, що мають різні функції, і гарантувати ефективну і безпечну комунікацію між ними. Використання CRC для перевірки цілісності даних додає додатковий рівень надійності, що важливо для систем, де передача інформації є критично важливою для безпеки та стабільності функціонування всієї мережі.

2.1.2 Механізм шифрування та автентифікації

Режим S0 протоколу Z-Wave розроблений для забезпечення безпеки даних, що передаються між пристроями в мережі, через використання шифрування та автентифікації. Цей режим дозволяє не лише захистити інформацію, але й підтвердити достовірність вузлів мережі, щоб запобігти підключенню несанкціонованих пристроїв.

Процес шифрування та автентифікації починається з обміну мережевим ключем (Kn) між контролером і вузлом Z-Wave. Цей ключ генерується за допомогою апаратного генератора псевдовипадкових чисел (PRNG) у контролері Z-Wave, разом з тимчасовим ключем за замовчуванням. Псевдовипадкові числа використовуються для створення надійного та унікального ключа, що забезпечує високу стійкість до атак на безпеку.

Мережевий ключ, що генерується, зберігається в енергонезалежній пам'яті (NVRAM) на пристрої, що гарантує збереження цього ключа навіть при відключенні живлення. Важливо, що мережевий ключ не використовується

безпосередньо для шифрування даних або надсилання повідомлень для автентифікації. Це дозволяє підвищити рівень безпеки, оскільки ключ не передається у відкритому вигляді по мережі.

Після того як мережевий ключ створено, він надсилається до вузла Z-Wave на етапі обміну ключами, що є частиною початкової налаштування мережі. Це дозволяє вузлу та контролеру здійснити криптографічну автентифікацію один одного. Такий процес є основою для забезпечення безпечної комунікації, адже кожен вузол в мережі повинен бути належним чином автентифікований, перш ніж зможе здійснити передачу або прийом даних [12].

Мережевий ключ використовується для виконання процесів шифрування повідомлень, що передаються між контролером і вузлами. Кожен кадр, що передається, зашифрований за допомогою цього ключа, що забезпечує конфіденційність інформації. Процес автентифікації виконується за допомогою механізмів підтвердження дійсності повідомлень, що дозволяє запобігти атакам, таким як підміна або повторна передача повідомлень.

Однією з переваг шифрування в Z-Wave є можливість оновлення ключів, що робить систему більш стійкою до атак зловмисників. Ключі можуть бути змінені за допомогою спеціальних команд управління, що дозволяє проводити періодичну зміну ключів для підвищення безпеки мережі.

Механізм шифрування та автентифікації, реалізований у режимі S0, є критично важливим для мереж Z-Wave, особливо в контексті безпеки в інтернеті речей (IoT). Використання сильних методів шифрування і регулярних оновлень ключів дозволяє зменшити ймовірність атаки на мережу та забезпечити цілісність і конфіденційність переданої інформації.

Таким чином, механізм шифрування та автентифікації в протоколі Z-Wave гарантує безпечну та надійну комунікацію між пристроями, що є особливо важливим для застосувань у смарт-будинках, де безпека є одним з основних пріоритетів, як показано на рис. 2.2.

Frame	Element	Description
Header	Home ID	Identify PAN of controller
	Source Node ID	Identify a sender node
	Properties 1 & 2	Several options
	Length	Total frame length
	Sequence Number	Ensure reliability of communication
	Dest. Node ID	Identify a receive node
App data	Command class	Identify specific command set
	Command	Identify specific command
	Data payload	Parameters used in command

Рисунок 2.4 - Елементи та опис кадру Z-Wave

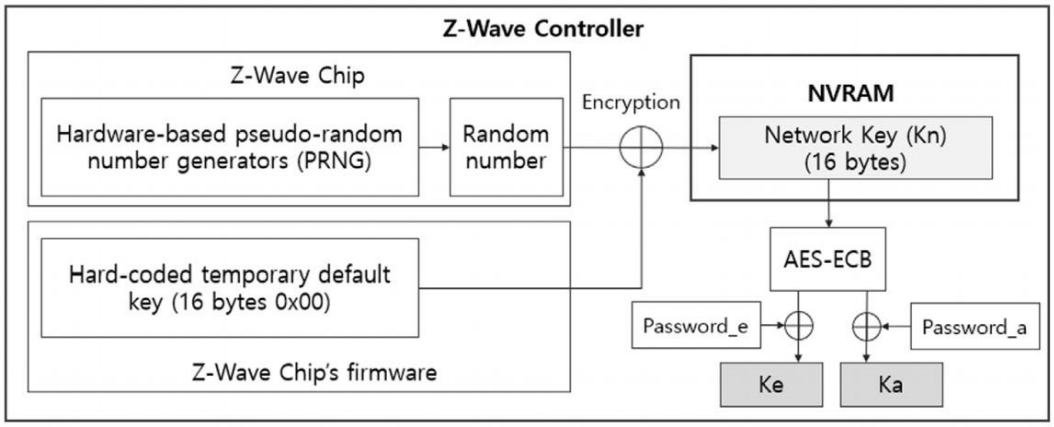


Рисунок 2.5 – Початкове створення ключа

1. Генерація ключів шифрування та автентифікації

Після встановлення зв'язку між контролером та вузлом мережі, система Z-Wave генерує два основні ключі для забезпечення безпеки:

- Ke (ключ шифрування): Цей ключ використовується для шифрування самого повідомлення, що передається. Шифрування даних важливе для забезпечення конфіденційності, адже навіть якщо дані будуть перехоплені, їх буде неможливо прочитати без відповідного ключа.

- Ka (ключ автентифікації): Використовується для підтвердження автентичності джерела повідомлення. Це критично важливо для захисту від атак

типу "man-in-the-middle", коли зловмисник може спробувати підробити або змінити повідомлення.

Ці ключі генеруються апаратним методом, використовуючи генератори псевдовипадкових чисел (PRNG). Вони зберігаються в енергонезалежній пам’яті (NVRAM) пристроїв, що гарантує їх збереження навіть після відключення живлення.

2. Шифрування повідомлення (Ke)

Шифрування повідомлення здійснюється з використанням алгоритму AES-OFB (Advanced Encryption Standard in Output Feedback mode). У цьому режимі шифрування кожен блок даних обробляється по черзі, і кожен з них залежить від попереднього. Вектор ініціалізації (IV) також використовується для уникнення однакових результатів шифрування при повторних передачах тих самих даних. Такий підхід покращує безпеку, оскільки кожне зашифроване повідомлення виглядає унікальним, навіть якщо його вміст однаковий.

Вектор ініціалізації (IV) містить nonce відправника і отримувача. Це важливо для захисту від повторних атак (replay attacks), де зловмисник може намагатися відправити старі повідомлення в мережу знову.

3. Генерація MAC (Ka)

MAC (Message Authentication Code) — це спеціальний код автентифікації, який генерується для перевірки того, що повідомлення не було змінено під час передачі. Це дозволяє переконатися, що дані не були підроблені або спотворені в процесі їх пересилання.

Для генерації MAC в Z-Wave використовується AES CBC-MAC (Cipher Block Chaining Message Authentication Code). Це метод, що дозволяє забезпечити як автентичність даних, так і їх цілісність. Використовуючи цей алгоритм, створюється спеціальний хеш, який залежить від вхідних параметрів, таких як:

- Ka — ключ автентифікації,

- IV — вектор ініціалізації,

- SH, SRC, DST, LEN, C — метадані та самі зашифровані дані.

MAC перевіряється на отримувачі: якщо хеш значення не співпадає, це свідчить про те, що повідомлення було змінено або підроблене.

4. Безпека і захист від атак

Механізм шифрування та автентифікації Z-Wave значно підвищує безпеку мережі, забезпечуючи:

- Конфіденційність: завдяки шифруванню повідомлення стають нечитаемими без правильного ключа [13].

- Цілісність: генерування MAC гарантує, що передані дані не були змінені.

- Автентичність: перевірка ключа автентифікації дозволяє переконатися, що повідомлення надійшло від авторизованого джерела.

Важливою частиною цього процесу є обмін ключами та їх захист під час передачі. Оскільки ключі генеруються випадковим чином і не використовуються безпосередньо для шифрування чи автентифікації до завершення налаштування, це ускладнює можливість атаки, що спрямована на отримання доступу до ключів.

5. Захист від атак типу "man-in-the-middle"

Зловмисники, які можуть бути в середині зв’язку між контролером і вузлом, не можуть змінити або спотворити передані повідомлення без відповідного ключа для автентифікації чи шифрування. Це робить атаку на шифрування або автентифікацію надзвичайно складною, оскільки зловмисник повинен знати правильний ключ для кожного з етапів.

Використання nonce та векторів ініціалізації (IV) є важливим для запобігання повторним атакам, коли зловмисник може повторно передати вже зібрані дані з попередніх сеансів комунікації. Ці елементи гарантують, що кожне повідомлення має унікальний код, навіть якщо воно ідентичне за вмістом.

Система Z-Wave використовує складні алгоритми шифрування та автентифікації для забезпечення безпеки. Це включає генерацію ключів шифрування та автентифікації, використання алгоритмів AES-OFB для

шифрування та AES CBC-MAC для автентифікації, а також перевірку цілісності та автентичності даних за допомогою MAC. Такі підходи значно зменшують ймовірність атаки та забезпечують безпеку мережі Z-Wave.

У реальних мережах передача пакетних даних, як показано на формулі 4, здійснюється у вигляді структурованого кадру, де кожен байт або група байтів має конкретне значення або функцію. Ось розбір кожного елементу пакету:

Формат кадру даних (рис. 2.3) та його складові:

1. F9 A6 C0 C8 (домашній ідентифікатор)

Це унікальний ідентифікатор домашньої мережі, який використовується для ідентифікації конкретної Z-Wave мережі. Цей ідентифікатор забезпечує, що тільки пристрої з відповідним ідентифікатором зможуть взаємодіяти у межах цієї мережі, захищаючи від несанкціонованого доступу з інших мереж.

2. 01 (ідентифікатор вузла джерела)

Це ідентифікатор вузла, який відправляє пакет. Важливий для вказівки на джерело повідомлення в мережі. Цей байт дає змогу отримувачеві правильно визначити, від кого надійшли дані.

3. 81 00 (властивості 1 та 2)

Тут зберігаються різні властивості кадру, які можуть включати інформацію про тип повідомлення, параметри налаштувань чи статуси команд. Ці байти можуть використовуватися для вказівки на додаткові характеристики передачі даних, наприклад, наявність підтвердження або спеціальних режимів роботи.

4. 23 (довжина)

Цей байт вказує на довжину корисного навантаження в пакеті, тобто скільки байтів даних слід обробити або передати. Це важливо для коректного прийому та обробки пакету, оскільки дозволяє отримувачу точно визначити, де закінчується корисне навантаження і де починаються інші частини пакету.

5. E5 (порядковий номер)

Це номер кадру або пакету в серії повідомлень, що допомагає отримувачу відслідковувати порядок пакетів у випадку їх розділення або повторної передачі. Цей номер важливий для підтримки послідовності кадрів при передачі великих обсягів даних.

6. 1D (ідентифікатор вузла призначення)

Ідентифікатор пристрою, до якого призначене повідомлення. Він дозволяє контролеру або вузлу визначити, на який пристрій або вузол потрібно направити повідомлення в мережі Z-Wave [14].

7. 98 81 (клас команди)

Це значення вказує на тип команди або запиту, який передається в кадрі. Клас команди визначає функціональність повідомлення, наприклад, чи це запит на зміну стану пристрою, отримання даних або інше.

8. FC A2 8B 0A 6E 38 73 1F (IV)

Це вектор ініціалізації (IV), що використовується для шифрування даних. IV необхідний для запуску алгоритму шифрування AES в режимі, що забезпечує уникнення однакових зашифрованих результатів для однакових вхідних даних при кожному шифруванні.

9. F6 F4 A7 1A (зашифровані команди даних та корисне навантаження даних)

Це частина, яка містить власне зашифровані дані або команди. Вони передаються в мережі у захищеному вигляді для запобігання доступу сторонніх осіб до конфіденційної інформації.

10. D2 (ідентифікатор відсутності приймача)

Цей байт позначає відсутність приймача, що може бути використано для індикації того, що пакет не був успішно доставлений або приймач не був доступний у момент передачі.

11. 44 69 4A F8 AD 28 D1 0C (код автентифікації повідомлень)

Це значення забезпечує автентифікацію пакету. Кожен пакет включає цей код для перевірки, що він надійшов від авторизованого джерела і не був змінений під час передачі. Код генерується за допомогою алгоритму хешування і ключа автентифікації (Ka), що гарантує цілісність даних.

12. DE CF (CRC, 16 байт)

Це код перевірки на циклічну коректність (CRC), який використовується для виявлення помилок передачі в пакеті. Якщо в процесі передачі були зміни (наприклад, через шум в каналі зв'язку), то CRC допоможе виявити ці помилки.

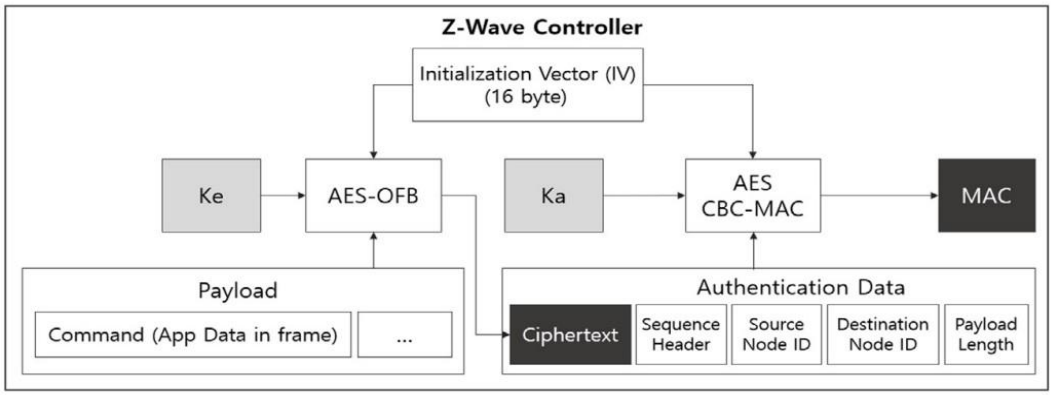


Рисунок 2.6 – Шифрування корисного навантаження та генерація MAC

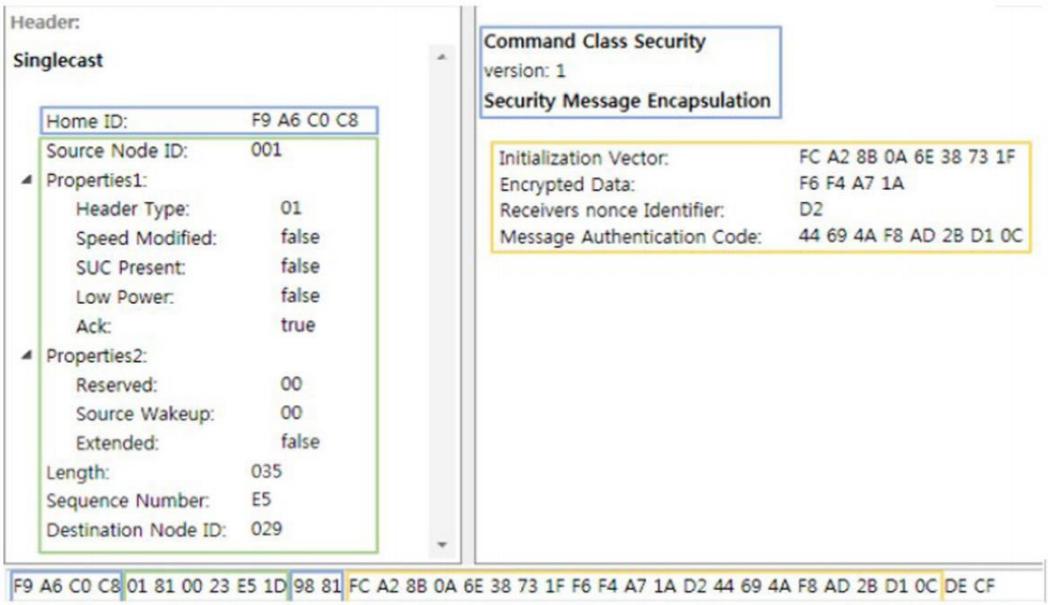


Рисунок 2.7 – Інформація про кадр Z-Wave S0

Цей детальний формат кадру показує, як Z-Wave використовує структуровану передачу даних для забезпечення безпеки, цілісності і правильної маршрутизації повідомлень в мережі. Кожен байт має чітко визначену роль, що дозволяє точно та надійно здійснювати передачу інформації в складних умовах реальних мереж.

2.1.3 Спільні ключові проблеми мережі

У мережі Z-Wave версії S0 обмін мережевим ключем (Kn) між контролером та вузлом є початковим кроком для генерації ключів шифрування (Ke) та автентифікації повідомлень (Ka). Це дозволяє забезпечити конфіденційність і автентичність даних, що передаються між пристроями в мережі. Для початкового обміну ключами використовуються три значення: мережевий ключ, Passworde та Passworda.

1. Початковий обмін ключами:

До обміну ключами мережевий ключ має значення 16 байт 0x00. Після завершення початкового обміну ключами генерується конкретне значення ключа, яке буде використовуватися для шифрування та автентифікації повідомлень. Це процес описано на рис. 2.8.

2. Проблеми з безпекою:

Основна проблема, пов'язана з цим методом обміну ключами, полягає в тому, що Passworde, Passworda та мережевий ключ можуть бути відомі через перехоплення на етапі початкового обміну ключами. Оскільки ці значення жорстко закодовані, зловмисники, які можуть перехопити передані дані під час цього процесу, можуть дістатися до важливих ключів. Це створює загрозу для безпеки, оскільки зловмисник може використовувати перехоплені значення для розшифровки зашифрованих даних.

Наприклад, якщо зловмисник перехоплює пакет даних, зібраний на початковому етапі обміну ключами, він може отримати доступ до мережевого ключа і використовувати його для дешифрування переданих даних. Це є потенційною вразливістю системи, особливо якщо цей процес відбувається в умовах, коли безпека каналу не гарантована.

3. Приклад атаки:

Як показано на рис. 2.9, зашифрований зразок даних 0x1E92A4 може бути успішно розшифрований до 0x009807. Зловмисник, маючи доступ до перехоплених даних, може додати додатковий байт (наприклад, 0x00) на початок даних, щоб сформувати новий зашифрований пакет. Це дозволяє змінювати вміст даних або створювати фальшиві повідомлення, що ставить під загрозу цілісність і достовірність даних у мережі.

Name	Before key exchange	After key exchange
Network key (K_n)	16-byte 0x00	446FBA73B03AD7456920F8D80BC1275D
$Password_e$	16-byte 0xAA	16-byte 0xAA
$Password_a$	16-byte 0x55	16-byte 0x55

Рисунок 2.8 – Дані ключа Z-Wave з жорстким кодуванням даних у S0 пристрої

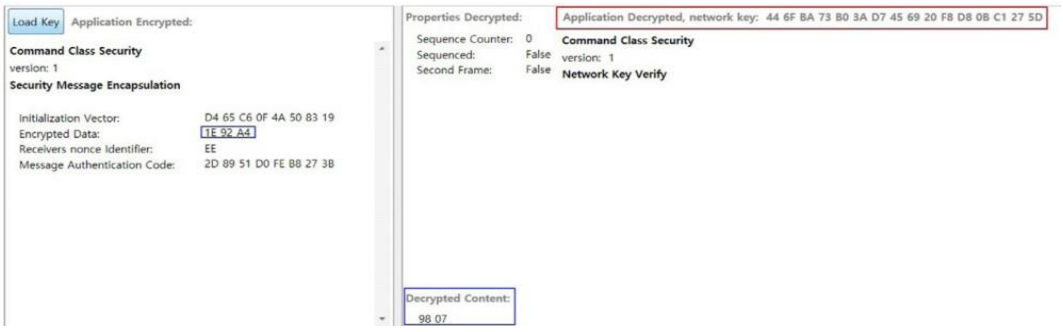


Рисунок 2.9 – Розшифровані дані S0 із зашифрованих даних за допомогою ключа (Kn)

Застосування жорстко закодованих значень (Passworde, Passworda та мережевого ключа) на початковому етапі обміну ключами у Z-Wave S0 мережі створює потенційні ризики для безпеки. Якщо зловмисник перехопить ці значення, він може здійснити атаки, розшифрувати зашифровані дані або змінити їх, що порушує безпеку мережі. Тому важливо використовувати більш надійні методи обміну ключами, щоб запобігти таким уразливостям.

2.2 Методологія дослідження системи

Методологія дослідження була побудована на чотирьох основних етапах, які дозволили комплексно оцінити вразливості пристроїв Z-Wave та розробити стратегії для їхнього аналізу та захисту. Ось основні етапи, що були реалізовані:

1. Підготовка тестового стенду для аналізу вразливостей пристроїв Z-Wave:

Для того щоб здійснити дослідження, спочатку був створений спеціальний тестовий стенд, що відтворює реальні умови роботи пристроїв Z-Wave. Це дозволило вивчити можливі вразливості та поведінку мережі в умовах атаки або спроби несанкціонованого доступу. Тестовий стенд включав в себе не лише фізичні пристрої, але й елементи програмного забезпечення для моніторингу та аналізу даних, що передаються в мережі Z-Wave [15].

2. Пояснення роботи методу DFD у мережі Z-Wave в реальному середовищі:

Далі було розглянуто метод діаграм потоку даних (DFD) для опису процесів передачі інформації в мережі Z-Wave. DFD допомагає візуалізувати, як дані проходять через систему, які пристрої взаємодіють, і як відбувається передача та обробка інформації на кожному етапі. Цей метод дозволив виявити можливі слабкі місця в процесах передачі даних та зрозуміти, де можуть виникати вразливості.

3. Застосування моделі загроз STRIDE для виявлення загроз та вразливостей:

Модель STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) була використана для ідентифікації

потенційних загроз та вразливостей в мережі Z-Wave. Ця модель допомогла систематично оцінити кожен компонент мережі з точки зору можливих загроз:

- Spoofing (підробка): можливість підробки ідентичності вузлів або контролера.

- Tampering (пошкодження): можливість зміни даних під час їх передачі.

- Repudiation (відмова від дій): можливість заперечення дій у мережі.

- Information Disclosure (розкриття інформації): ризик витоку конфіденційних даних.

- Denial of Service (відмова в обслуговуванні): атаки, які блокують або обмежують доступ до сервісів.

- Elevation of Privilege (підвищення привілеїв): можливість отримання несанкціонованого доступу до системи або її компонентів.

4. Створення дерев атак для сценаріїв атак Z-Wave:

Останнім етапом було створення дерев атак (Attack Trees) для опису можливих атак на систему Z-Wave та оцінки їхнього потенційного впливу. Дерева атак дозволяють детально описати різні сценарії, за яких зловмисник може спробувати зламати систему або викликати збої в роботі мережі. У процесі створення дерев атак було враховано різні типи загроз, виявлені за допомогою моделі STRIDE, а також вивчені конкретні методи та інструменти для реалізації атак на мережу Z-Wave.

Методологія, що включала етапи підготовки тестового стенду, застосування DFD, модель STRIDE та створення дерев атак, забезпечила глибоке розуміння вразливостей пристроїв Z-Wave. Ці етапи дозволили виявити потенційні слабкі місця в мережі та запропонувати шляхи їхнього усунення, що може підвищити безпеку мережі Z-Wave в реальних умовах експлуатації.

2.2.1 Підготовка пробного стенду

На рисунку 2.10 показано положення випробувального стенду та опис кожного об'єкта, що використовується для тестування та аналізу вразливостей пристроїв Z-Wave в рамках цього дослідження.

Пристрої Z-Wave, які були використані, працюють на частоті 923,10 МГц, що є стандартом для відповідного регіону. Це дозволяє забезпечити сумісність з іншими пристроями та мережами, що підтримують Z-Wave, та використовуються для передачі даних в умовах реального середовища.

Випробувальний стенд був побудований на основі «Z-Wave IoT Home Service», що є інтегрованою платформою для управління та аналізу пристроїв в Інтернеті речей (IoT). Ця платформа дозволяє здійснювати моніторинг мережі Z-Wave та тестувати безпеку пристроїв, що входять до її складу, в умовах, максимально наближених до реальних.

Опис елементів випробувального стенду:

- Контролер Z-Wave: Використовується для управління мережею пристроїв Z-Wave, налаштування ключів та комунікацій між вузлами.

- Вузли Z-Wave: Різні пристрої, такі як сенсори, замки, освітлення та інші, що взаємодіють з контролером для передачі даних або виконання команд.

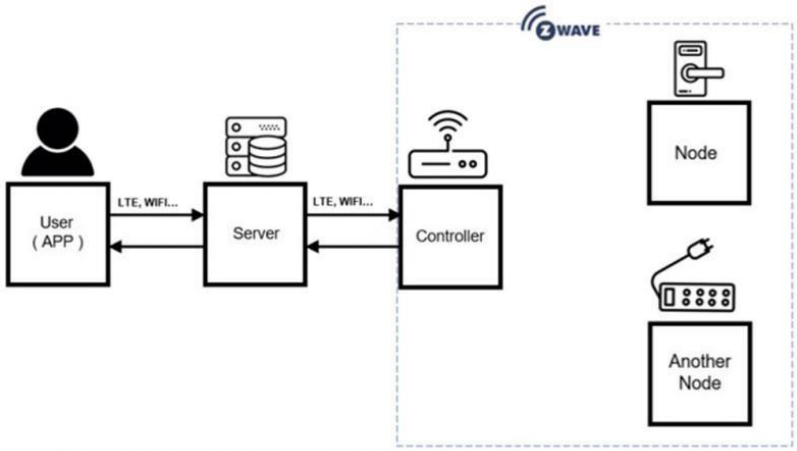
- Захищений канал зв'язку: Забезпечує передачу даних між пристроями на основі методів шифрування та автентифікації, що тестуються на стенді.

- Моніторинг безпеки: Інструменти для спостереження за передачею даних, збором статистики та виявленням можливих атак або вразливостей.

- Інтерфейс для тестування: Програмне забезпечення для контролю і аналізу результатів тестування, яке забезпечує візуалізацію мережі та її компонентів.

Цей стенд дозволяє здійснювати різноманітні тести, включаючи аналіз безпеки протоколів передачі, тестування вразливостей пристроїв до атак типу

«відмова в обслуговуванні» (DoS), перехоплення даних, а також аналіз інших аспектів мережі Z-Wave для виявлення потенційних загроз.



Components	Description
User(App)	Send command to Server by App
Server	Send command to Controller
Controller	Controls multiple nodes with Z-Wave
Node	Door lock controlled by Controller
Another node	Multi-tab controlled by Controller

Рисунок 2.10 – Навколишнє середовище та опис Z-Wave

2.2.2 Аналіз діаграми потоків даних

DFD (Data Flow Diagram) — це метод представлення або виведення потоку даних у процесі або системі. DFD є корисними для виявлення вразливостей продукту, оскільки вони дозволяють чітко розуміти, як дані пересуваються через систему і де можуть виникнути потенційні загрози безпеці. Багато дослідників безпеки використовують DFD для системного визначення вразливості цільових продуктів [16].

Для створення DFD спочатку розробляється контекстна діаграма. Контекстні діаграми є найбільш абстрактними діаграмами рівня для цільової екосистеми. Вищий рівень DFD (рівень 0) є більш абстрактним, де потік даних описується на загальному рівні, тоді як нижчий рівень DFD (рівень 2) є більш конкретним, де детально показано взаємодію компонентів системи.

Більшість досліджень, пов'язаних з DFD, аналізували діаграми рівня 0 або 1. В даній роботі було проаналізовано DFD рівня 1 і рівня 2 для реальних пристроїв Z-Wave в екосистемі розумного будинку, як показано на рис. 2.11 та в додатку А відповідно.

Опис елементів DFD:

1. Коло (процес): Коло на діаграмі представляє процес, який обробляє отримані дані відповідно до заздалегідь визначеної процедури та видає дані нового типу. У випадку з Z-Wave це може бути обробка даних від вузлів або контролера для подальшої комунікації в мережі.

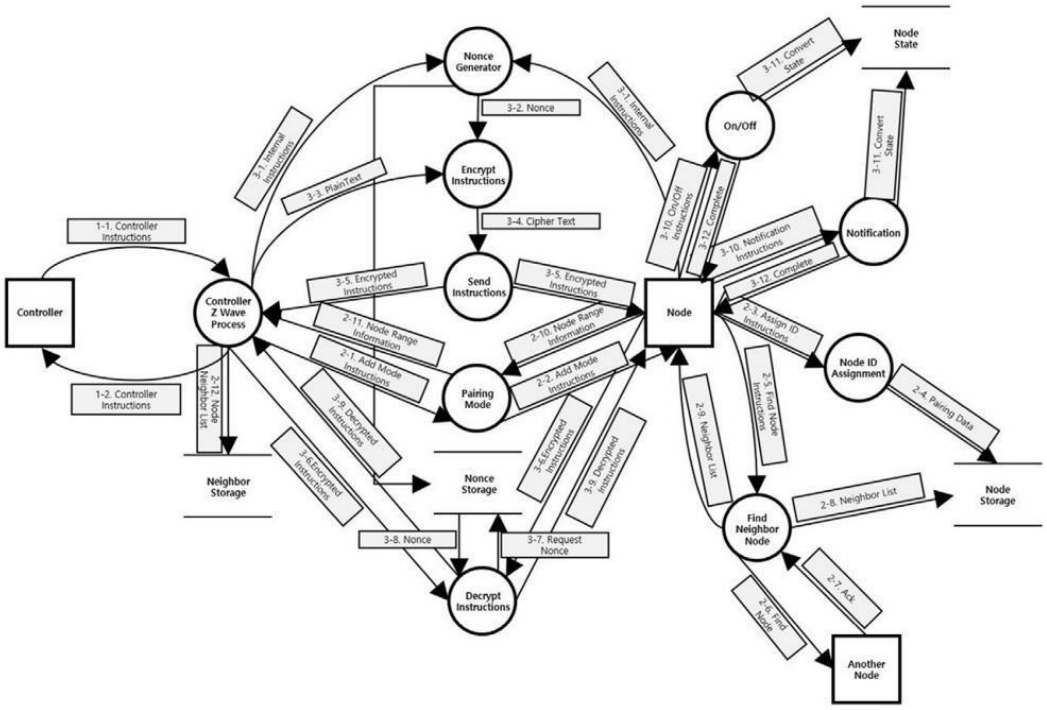
2. Квадрат (сутність): Квадрат позначає сутність, яка є абстрактним поняттям, де починається потік даних. Наприклад, вузли Z-Wave та контролер можуть бути представлені як сутності. Вони є джерелами чи отримувачами даних.

3. Стрілка (потік даних): Стрілка вказує на напрямок потоку даних, що передаються від одного процесу чи сутності до іншого.

4. Довірена межа: Це роздільна лінія, яка вказує область, в якій змінюються повноваження даних. Вона визначає межі доступу до даних між різними компонентами системи або між різними рівнями безпеки.

5. Зберігання даних: Зберігання даних представляє собою елемент, який зберігає вхідні дані та видає їх на запит процесу. Це може бути пам'ять, де зберігаються ключі шифрування, автентифікаційні дані чи інша важлива інформація для подальшої обробки.

Діаграма потоків даних дозволяє чітко побачити, як інформація рухається через систему Z-Wave, і допомагає виявити потенційні вразливості на етапах обробки та передачі даних, що є критичними для забезпечення безпеки мережі.



Рис

унок 2.11 – Діаграма протоколу Z-Wave рівень 1

2.2.3 Визначення моделі Z-Wave STRIDE

Метод STRIDE є потужним інструментом для систематичного виявлення загроз у системах, і в даному контексті він використовується для аналізу безпеки мережі Z-Wave. Модель STRIDE охоплює шість основних типів загроз, і її застосування до Z-Wave дозволяє детально виявити потенційні вразливості в мережі і системах, що використовують цей протокол. Давайте детальніше розглянемо кожну з цих загроз і механізми, за допомогою яких їх можна мінімізувати або усунути.

Підробка відноситься до ситуацій, коли зловмисник намагається видати себе за іншого користувача або пристрій. Це може бути особливо небезпечно в контексті мережі Z-Wave, де пристрої взаємодіють між собою через радіосигнали, які можуть бути підроблені. Щоб протистояти такій загрозі, необхідно забезпечити механізми автентифікації для пристроїв. Це може включати

використання криптографічних протоколів, які гарантують, що дані, які передаються по мережі, дійсно походять від авторизованих джерел.

Втручання — це тип атаки, за якого зловмисник змінює вміст переданих даних, щоб отримати вигоду або викликати збій системи. В мережах Z-Wave це може статися, якщо дані, що передаються між пристроями, не захищені належним чином. Для протидії цій загрозі використовується шифрування даних. Застосування криптографії на рівні даних (наприклад, AES) дозволяє гарантувати, що дані не будуть змінені або пошкоджені під час їх передачі.

Загроза відмови полягає в тому, що одна зі сторін може заперечити свою участь у транзакції, навіть якщо вона справді мала місце. Це може бути небезпечним у випадках, коли необхідно підтвердити справжність дій (наприклад, зміна налаштувань пристрою). Для захисту від цієї загрози використовуються цифрові підписи та механізми автентифікації, які дозволяють відслідковувати та підтверджувати дію користувачів або пристроїв в системі.

Розкриття інформації передбачає витік конфіденційних даних. У мережах Z-Wave це може включати несанкціонований доступ до персональних або чутливих даних користувачів. Для мінімізації цієї загрози необхідно використовувати шифрування переданих даних. Використання таких алгоритмів, як AES, дозволяє забезпечити конфіденційність комунікацій і запобігти витоку даних.

Атака DoS полягає в тому, що зловмисник намагається зробити систему або сервіс недоступними, наприклад, через вичерпання ресурсів або блокування передачі даних. У випадку з Z-Wave це може включати, наприклад, глушіння радіосигналу або створення надмірного навантаження на мережу. Для захисту від DoS атак важливо впроваджувати механізми захисту від глушіння та зловживань, такі як використання більш стійких до перешкод частот або розподілене управління ресурсами.

Ця загроза виникає, коли зловмисник отримує доступ до системи з нижчими правами, а потім використовує вразливості для підвищення своїх привілеїв. В

мережах Z-Wave це може бути пов'язано з доступом до налаштувань пристроїв або мережевих ресурсів. Для протидії цій загрозі важливо забезпечити належну систему прав доступу, щоб користувачі та пристрої могли виконувати лише ті дії, які їм дозволені.

Виявлення загроз за допомогою STRIDE вимагає ретельного аналізу кожного компонента екосистеми Z-Wave. Зокрема, за допомогою методів моделювання загроз, таких як діаграми потоку даних (DFD), можна виявити критичні точки в системі, де можуть виникнути уразливості. Аналіз рівня 2 DFD дозволяє створити деталізовану картину того, як дані переміщуються через систему, і де можуть бути виявлені загрози для кожного етапу обробки інформації.

Кожна з цих загроз вимагає відповідних заходів безпеки, і їх вплив на мережу Z-Wave можна мінімізувати за допомогою сучасних криптографічних методів, налаштування прав доступу та автентифікації, а також використання технологій захисту від DoS атак.

Метод STRIDE виявив 46 потенційних загроз для мережі Z-Wave, що підкреслює важливість вивчення системи та її вразливостей на кожному рівні. Визначення цих загроз є важливим етапом для покращення безпеки і надійності мережі Z-Wave, а також для розробки ефективних стратегій захисту від потенційних атак.

2.2.4 Створення векторів атак

Використовуючи методи DFD і STRIDE, було виведено 46 потенційних загроз, що дозволило створити дерево атак, яке має на меті забезпечити повний контроль над розумним будинком. Ключовою метою є захоплення контролю над мобільним телефоном користувача, контролером і кожним вузлом мережі Z-Wave. Однак, в цьому дослідженні акцент зроблений на вразливостях самого протоколу Z-Wave, при цьому атаки на мобільний телефон користувача були виключені з аналізу.

Для досягнення цієї мети було визначено п'ять основних векторів атак, які можуть призвести до повного контролю над розумним будинком:

1. Атака відтворення Z-Wave

2. Атака за допомогою пульта дистанційного керування Z-Wave

3. Атака DoS для мережі Z-Wave

4. Атака Z-Wave FOTA (Firmware Over-the-Air)

5. Атака дистанційного керування додатковим режимом

Для поглибленого аналізу було детально перевірено три з цих векторів атак, за винятком атаки відтворення та атаки пульта дистанційного керування, що дозволяє отримати більш конкретні дані щодо вразливостей в протоколі Z-Wave.

1. Атака відтворення Z-Wave (Replay Attack)

Атака відтворення відбувається, коли зловмисник перехоплює законний сигнал між двома пристроями та відтворює його для того, щоб здійснити несанкціоновану дію. Вона використовує принцип «запису та відтворення» сигналу, що дозволяє шахраю повторно виконати команду або операцію, яку передав законний користувач або пристрій. Протокол Z-Wave може бути вразливим до цього типу атак, якщо механізми аутентифікації або шифрування не застосовуються належним чином.

2. DoS-атака Z-Wave (Denial of Service Attack)

Атаки типу DoS спрямовані на блокування або обмеження доступу до ресурсу, що робить систему недоступною для законних користувачів. Для мережі Z-Wave DoS-атака може бути спричинена шляхом перевантаження мережі або блокування передачі даних, що може призвести до відмови в роботі пристроїв або систем. Такі атаки можуть бути виконані за допомогою технологій, які глушать радіосигнали або створюють інші перешкоди для нормальної роботи мережі.

3. Атака Z-Wave FOTA (Firmware Over-the-Air)

Атаки FOTA націлені на модифікацію або заміну мікропрограмного забезпечення пристроїв Z-Wave. За допомогою цих атак зловмисник може

здійснити несанкціоноване оновлення прошивки пристроїв у мережі Z-Wave, змінюючи їхню поведінку або вводячи вразливості, які можна використати для подальших атак. Це може бути дуже небезпечним, оскільки дозволяє віддалено змінювати програмне забезпечення на пристроях і потенційно отримувати доступ до конфіденційних даних або систем.

4. Дистанційне управління додатковим режимом (Remote Control Additional Mode)

Цей вектор атак націлений на використання незахищених або недостатньо захищених каналів дистанційного керування для активації додаткових режимів в пристроях Z-Wave. Використовуючи вразливості в механізмах управління, зловмисник може включити додаткові функції або отримати доступ до контрольованих пристроїв, що може призвести до несанкціонованого контролю над системою [17].

Attack vector	Description	Prerequisites
(i) Z-Wave DoS	With specific crafted data, it is possible to disable the features of Z-Wave IoT devices.	The attacker must be in the same network area as the controller.
(ii) Z-Wave FOTA	Vulnerabilities are identified when the Z-Wave firmware is updated via wireless connection.	The attacker must know the network key value by snipping during the pairing process.
(iii) Z-Wave remote add-mode control	With remote add-mode vulnerabilities, it is possible to control any Z-Wave device remotely.	The attacker must know the admin ID and password of the router connected the controller.

Рисунок 2.12 – Опис та передумови трьох векторів атаки

Цей аналіз надає глибоке розуміння потенційних загроз для протоколу Z-Wave, дозволяючи розробити стратегії для їх пом’якшення і підвищення безпеки розумних будинків, що використовують цей протокол.

Під час аналізу процесу зв'язку в мережі Z-Wave було виявлено, що довгі пакети Z-Wave діляться та передаються через кілька менших кадрів, щоб забезпечити їх ефективну передачу в межах обмеженої пропускної здатності каналу. Однак ця особливість архітектури може стати вразливою точкою для атак, таких як пінг-смерть (Ping of Death, PoD).

Пінг-смерть — це тип DoS-атаки, яка полягає в надсиланні спеціально сформованих пакетів з помилково великим розміром, що може призвести до збою в мережевих пристроях або системах, які не мають належного захисту від такого роду пакетів. В контексті Z-Wave, якщо зловмисник створює надмірно великі пакети, вони можуть бути поділені на кілька малих кадрів під час передачі, що може спричинити перевантаження системи і порушення зв’язку між пристроями.

У системах Z-Wave, де використовується командний клас транспортної служби для розподілу довгих пакетів на менші фрагменти, зловмисник може скористатися цією особливістю, щоб відправити непередбачено великі або неправильні пакети. Це може привести до:

1. Перевантаження приймальних пристроїв – коли пристрій отримує надмірно великі або фрагментовані пакети, він може не встигати обробляти їх належним чином, що спричинить збої в його роботі або навіть повний вихід з ладу.

2. Блокування комунікацій – атака може перервати нормальний процес зв'язку між пристроями, блокуючи їхню здатність здійснювати зв'язок між собою. Це може унеможливити передачу важливих даних або команд, які необхідні для функціонування розумного будинку.

3. Витрати ресурсів – надмірне навантаження на мережу може призвести до великих витрат енергоресурсів на пристроях, які постійно намагаються обробити або передати ці пакети.

Запобігання атаці

Для пом'якшення цієї вразливості протоколу Z-Wave можна застосувати кілька стратегій:

1. Перевірка розмірів пакетів – впровадження механізмів, які перевіряють розмір пакету перед його відправкою, дозволить запобігти надмірно великим або некоректним пакетам, які можуть спричинити збої.

2. Фільтрація та валідність даних – введення додаткових рівнів перевірки та валідації пакунків на рівні кожного вузла мережі дозволить виявити спроби відправлення небажаних або маніпульованих даних.

3. Адаптація до змін у трафіку – використання алгоритмів, які дозволяють мережі адаптуватися до змін у навантаженні та коригувати пропускну здатність для уникнення перевантаження.

Таким чином, хоча архітектура Z-Wave дозволяє ефективно передавати великі обсяги даних, вона також вимагає додаткових заходів безпеки для запобігання DoS-атакам, зокрема таких як пінг-смерть.

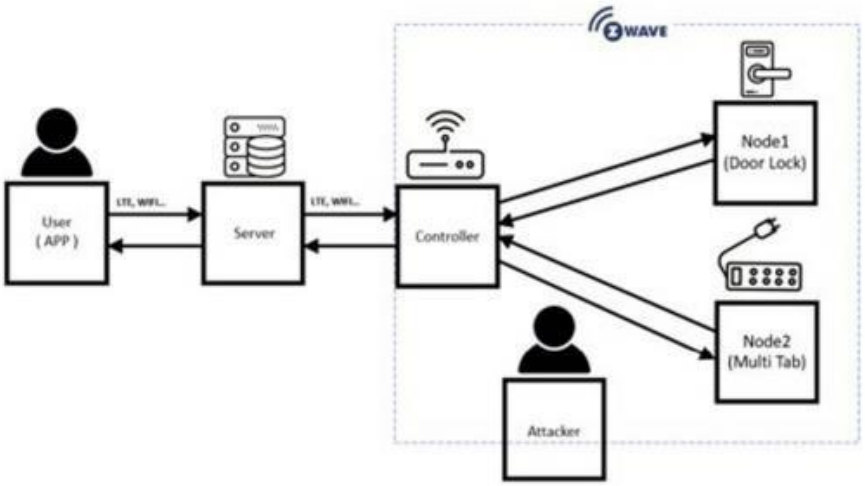


Рисунок 2.13 – Z-Wave DoS

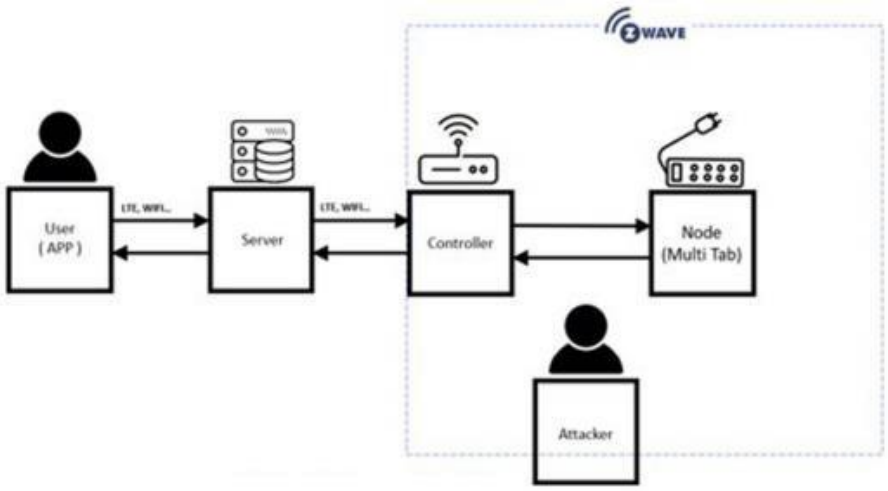


Рисунок 2.14 – Z-Wave FOTA

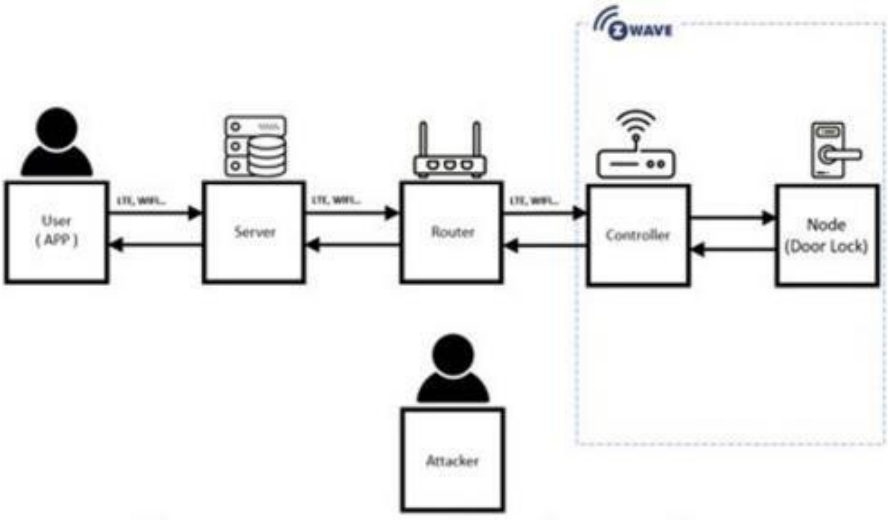


Рисунок 2.15 – Віддалене управління додатковим режимом Z-Wave

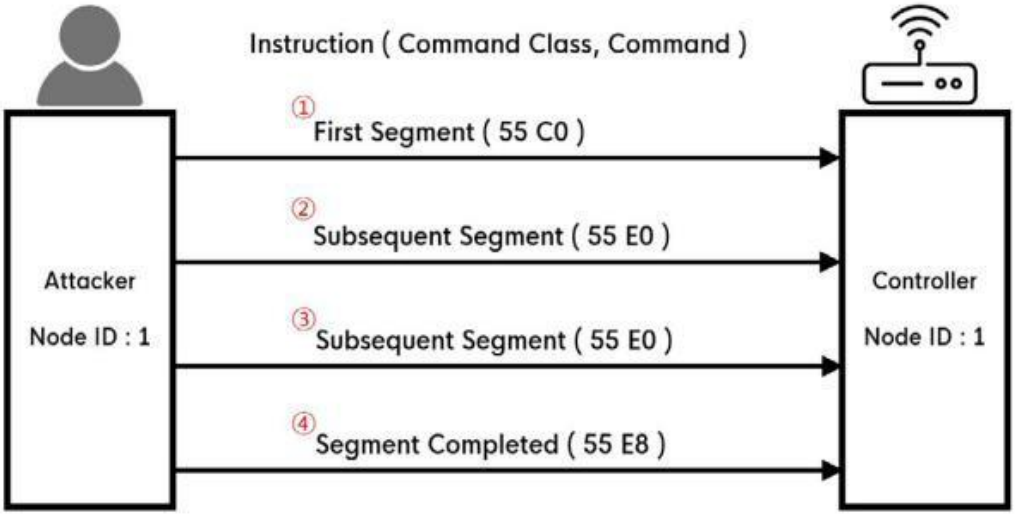


Рисунок 2.16 – Метод сегментації кадру Z-Wave

Процес використання команди транспортної послуги в Z-Wave для DoS-атак

1. Розділення пакету: Коли довгий пакет передається через мережу Z-Wave, він розділяється на менші сегменти для забезпечення ефективної передачі. Процес виглядає так:

- Перший сегмент довгого пакету (0x55 C0) приймається контролером.
- Два наступні сегменти (0x55 E0) отримуються контролером.
- Останній сегмент (0x55 E8) вважається завершальним сегментом пакету.

2. Обробка сегментів: Всі подальші сегменти після першого обробляються як наступні сегменти пакету. Це означає, що процес розділення та обробки даних передбачає валідацію та злиття сегментів після їх отримання.

3. Атака пінг-смерть (DoS): У разі атаки, зловмисник може надіслати закінчений сегмент пакету після його поділу, що призведе до того, що приймальний вузол виконає звичайні команди управління після процесу злиття. Цей процес можна спричинити неконтрольованим збільшенням навантаження на мережу або пристрої, що може призвести до збоїв у зв'язку між пристроями мережі.

Для проведення тестування DoS-атаки було створено спеціальне середовище моделювання, яке показано на рис. 2.13. В середовищі моделювання було використано WYP Z-Wave Spoofing Tool (рис. 2.17), яке дозволяє здійснювати атаку, підробляючи пакети. В середовищі було налаштовано два вузли — дверний замок та мульти-вкладинка, підключені до контролера.

Процес атаки включав такі етапи:

1. Розширення кадру nonce get: Кадр nonce get був розширений і постійно надсилався до контролера.

2. Надсилання розділеного кадру: Кадр nonce get був поділений на кілька сегментів і надсилався до вузлів. Цей процес використовував вразливість протоколу в його версії безпеки S0, коли контролер надсилає кадр nonce get до вузла, а вузол відповідає на нього у звіті nonce.

3. Атака DoS: Завдяки несанкціонованому надсиланню сегментів кадру nonce get, зловмисник може створити ситуацію, коли вузол буде постійно обробляти ці кадри, що призведе до відмови в обслуговуванні (DoS) для інших важливих комунікацій в мережі Z-Wave.

Кроки атаки Z-Wave DoS

Як показано на рис. 2.18, процес атаки складається з наступних етапів:

1. Зловмисник ініціює атаку, надсилаючи кадр nonce get.

2. Кадр nonce get розділяється на кілька менших пакунків.

3. Зловмисник постійно надсилає ці кадри, що викликає перевантаження в процесі з'єднання.

4. Контролер не може ефективно обробляти надмірні кадри, що призводить до збоїв або навіть до повного блокування комунікації з іншими пристроями.

Процес атаки Z-Wave DoS через підроблені кадри та розширення кадру nonce get показує, як важливо враховувати вразливості в архітектурі протоколу. Для забезпечення безпеки в мережах Z-Wave потрібно впроваджувати додаткові

заходи для захисту від таких атак, зокрема використання захищених версій протоколу та засобів для виявлення та запобігання підробці кадрів.

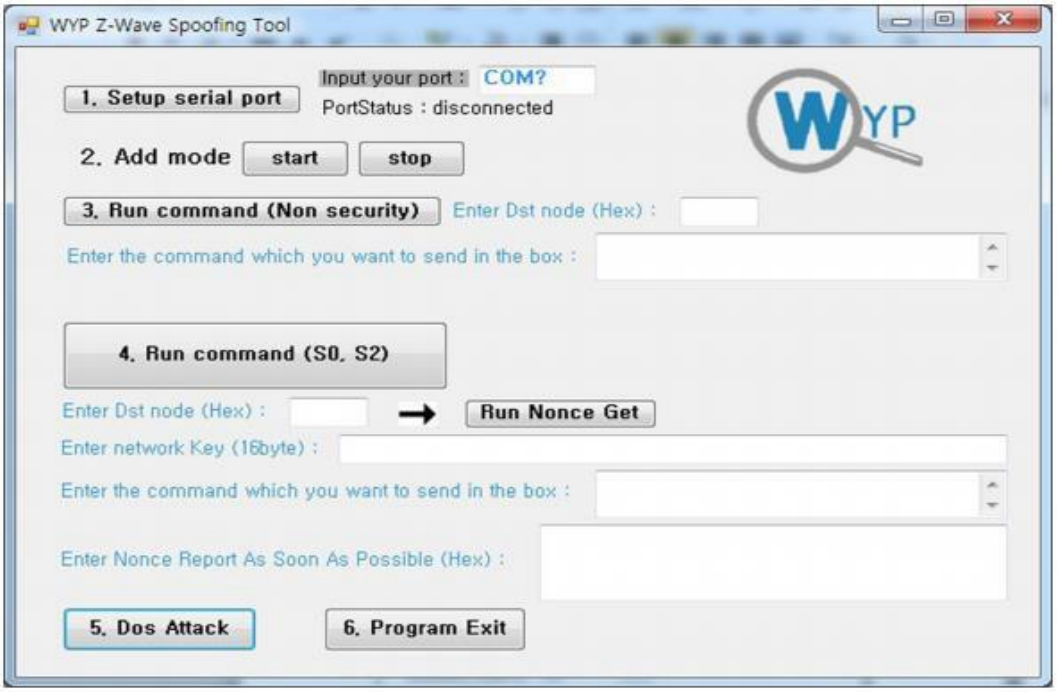


Рисунок 2.17 – Z-Wave Spoofing Tool

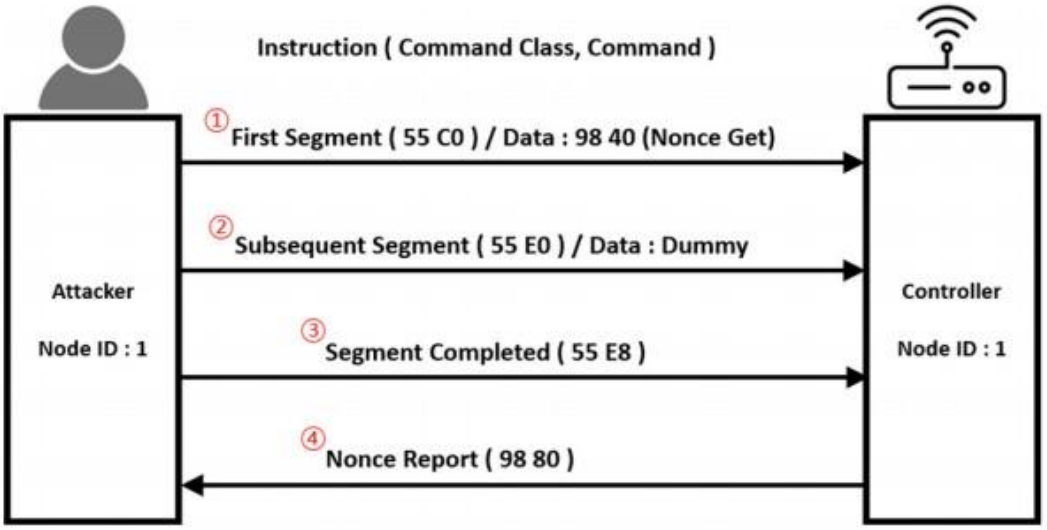


Рисунок 2.18 – Z-Wave процес атаки DoS

1. Надсилання першого сегмента:

Контролеру було надіслано перший сегмент пакету (0x55 C0) з даними 0x98 40 (nonce get). Цей сегмент являв собою початок процесу передачі даних.

2. Надсилання наступного сегмента:

Потім було надіслано другий сегмент (0x55 E0) з фіктивними даними, що не мали реальної інформації для обробки, але забезпечували продовження потоку передачі.

3. Завершення пакету:

Завершення пакету відбулося шляхом надіслання останнього сегмента (0x55 E8), що сигналізувало про закінчення передачі даних для цього пакету.

4. Відповідь контролера:

Контролер надіслав 0x98 80 зловмиснику як звіт про відсутність (відповідь на nonce get), що вказує на проблему з отриманими даними або їхню відсутність.

5. Надсилання довгих даних:

Після цього зловмисник надіслав довгі дані у вигляді численних значень 0x11 після 0x98 40, що використовується для вчинення DoS-атаки (відмова в обслуговуванні). Цей процес є частиною атаки на мережу, де ці дані не несуть корисної інформації для контролера, але створюють великий потік трафіку, що перевантажує систему [18].

6. Виявлення результату атаки:

Після виконання атаки було виявлено, що два вузли, вузол 1 (дверний замок) та вузол 2 (мульти-вкладинка), більше не змогли підключитися до контролера, оскільки його ресурси були заблоковані або перевантажені великою кількістю фальшивих даних.

Ця атака є прикладом того, як можна ефективно використати уразливості в протоколі Z-Wave для перевантаження мережі і блокування пристроїв. Завдяки маніпуляціям з пакунками і надсиланню фальшивих сегментів, зловмисник може заблокувати з'єднання між контролером і пристроями, що критично впливає на

функціонування розумного дому. Для протидії таким атакам необхідно використовувати механізми верифікації даних, покращення безпеки передачі інформації та захист від підроблених кадрів.

FOTA (Firmware Over-The-Air) є важливим процесом для оновлення мікропрограм пристроїв у бездротових мережах, зокрема, в мережах Z-Wave. Ця технологія дозволяє значно спростити процес оновлення програмного забезпечення пристроїв, оскільки вона дозволяє здійснювати оновлення мікропрограм без необхідності фізичного доступу до пристроїв. Однак, як і будь-яка технологія, вона має свої вразливості, особливо якщо не забезпечена належним рівнем захисту.

У Z-Wave процес FOTA включає кілька етапів, які детально описуються під час обміну даними між вузлом і контролером. Перш за все, вузол, який потребує оновлення мікропрограми, надсилає запит на отримання оновлення через команду Md Get. У відповідь на цей запит контролер надсилає звіт Md (Firmware Report), який містить шість основних елементів, включаючи ідентифікатор виробника, ідентифікатор прошивки, контрольну суму прошивки та інші параметри, необхідні для оновлення.

Коли запит на оновлення мікропрограми отримано та підтверджено, починається процес сегментації даних прошивки. Оскільки деякі мікропрограми можуть бути досить великими, дані поділяються на сегменти та передаються поетапно. Цей процес дає змогу доставити великі обсяги даних навіть у випадку обмеженої пропускної здатності мережі.

Однак незахищений процес FOTA створює вразливості в мережах Z-Wave. Якщо пристрій використовує незахищену версію FOTA, зловмисники можуть скористатися цією вразливістю, щоб атакувати систему. Одна з таких атак полягає в крадіжці мережевого ключа, що може статися під час процесу оновлення мікропрограми. Зловмисник може впровадити шкідливе програмне забезпечення в

одному з сегментів оновлення, а після цього контролер може передавати шкідливі дані, не помітивши підміни.

Однією з основних загроз є використання шкідливого програмного забезпечення для модифікації інформації, що передається між пристроєм і контролером під час процесу оновлення. Якщо зловмисник має змогу змінювати або підміняти пакети, що містять дані про мікропрограму, це дозволяє йому отримати доступ до критичних елементів системи. Наприклад, замінивши код, що міститься в пакеті, зловмисник може активувати невідоме шкідливе програмне забезпечення, яке може порушити нормальну роботу пристроїв або навіть вкрасти конфіденційну інформацію.

Однією з найбільш серйозних вразливостей є слабка автентифікація даних, що передаються під час оновлення мікропрограми. Якщо система не перевіряє належним чином автентичність даних, що надходять від контролера, зловмисник може втрутитися в цей процес і замінити або підробити необхідні сегменти. У такому випадку навіть правильно сегментовані та перевірені пакети можуть стати джерелом уразливості, якщо зловмисник має можливість змінити вміст цих пакетів до їх отримання вузлом.

Для усунення цих обмежень виявлені шляхи атаки потребують постійного вдосконалення механізмів захисту. Один із таких шляхів полягає в застосуванні більш складних методів шифрування для захисту даних, які передаються під час оновлення мікропрограми. Шифрування забезпечує конфіденційність і захист від несанкціонованого доступу, що ускладнює зловмисникам завдання.

Крім того, важливо удосконалити автентифікацію даних, щоб підтвердити автентичність мікропрограми перед її завантаженням на пристрій. Механізми перевірки автентичності повинні бути стійкими до підробки або маніпуляцій з даними, що надходять від контролера, що дозволяє запобігти атакам, спрямованим на підмінку або зміни цих даних.

У разі незахищених систем, де процес FOTA не містить належного рівня перевірки автентичності або шифрування, зловмисники можуть отримати можливість не тільки викрасти важливі дані, але й встановити шкідливе програмне забезпечення, що може серйозно порушити роботу пристроїв або призвести до повної компрометації всієї мережі.

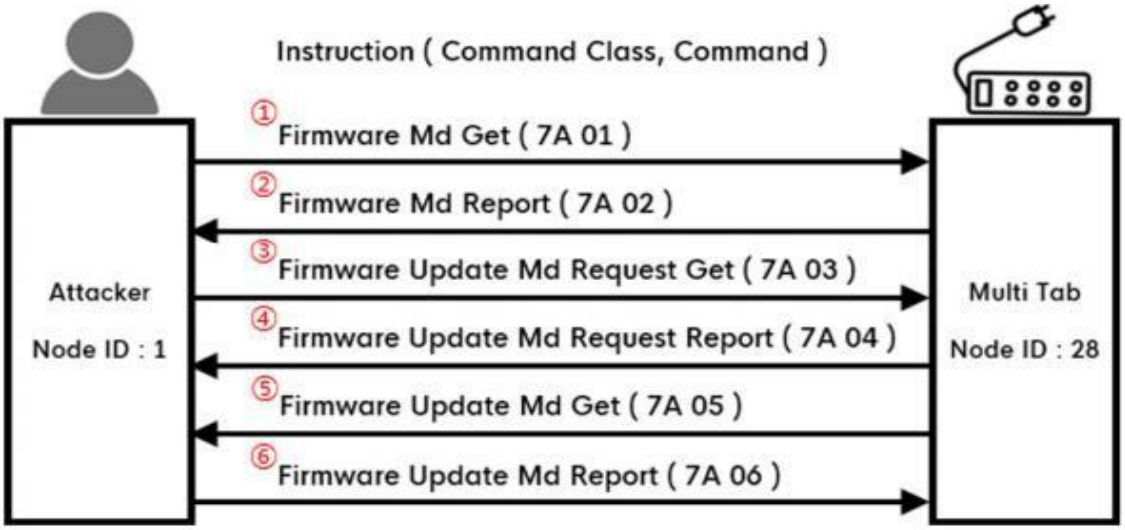


Рисунок 2.19 – Зв'язок Z-Wave FOTA між зловмисником та Multi Tab

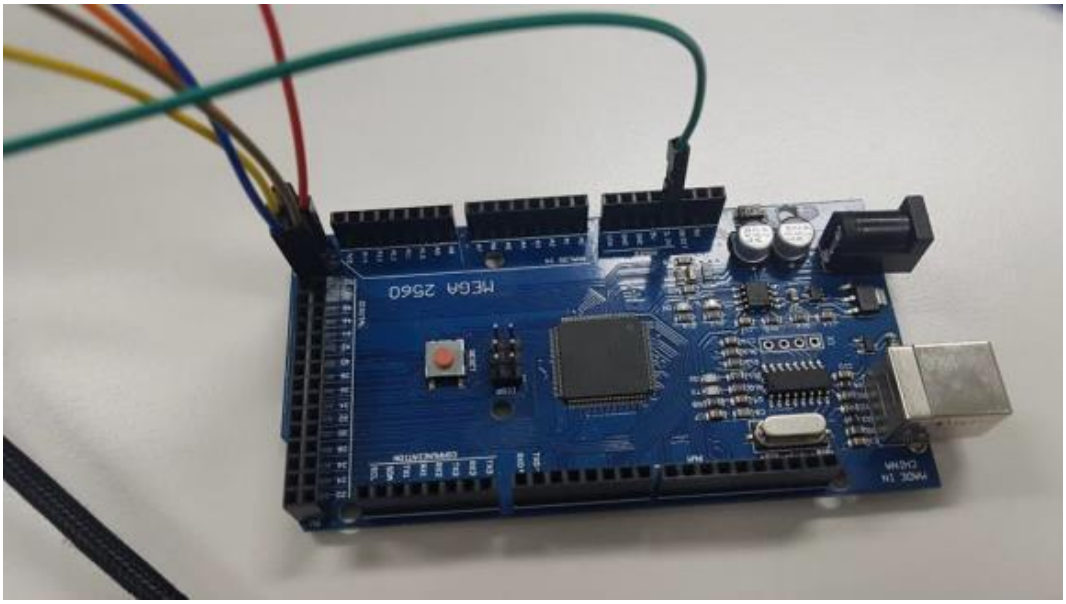


Рисунок 2.20 – Обладнання ATmega2560 для тестування FOTA

1. Аналіз вихідного коду контролера:

Оскільки Z-Wave дозволяє здійснювати оновлення мікропрограми через бездротові з'єднання, перший крок полягав у завантаженні і модифікації вихідного коду контролера «ZWaveController.sln», наданого компанією Silicon Labs. Цей код включає в себе компоненти для взаємодії з пристроями Z-Wave, а також для здійснення оновлень мікропрограм за допомогою FOTA. Після отримання доступу до цього коду, була проведена його модифікація, яка дозволяла змінювати параметри, зокрема ідентифікатор вузла, на який було спрямоване оновлення.

2. Модифікація ідентифікатора вузла призначення:

Важливим моментом в атаці є здатність змінювати ідентифікатор вузла призначення (Destination Node ID). Зазвичай цей ідентифікатор визначає, до якого пристрою має бути направлена прошивка під час оновлення. Зловмисники змогли змінити значення цього ідентифікатора в коді контролера, таким чином перенаправивши процес оновлення на інший пристрій, навіть якщо це не було передбачено в його початковій конфігурації.

3. Отримання інформації про мікропрограму:

Для того щоб змінити мікропрограму на зловмисну, потрібно було отримати інформацію про прошивку від цільового пристрою. Використовуючи команду «Md Get», можна було запитати і отримати звіт про поточну прошивку. Цей звіт містить ключові елементи, такі як ідентифікатор виробника, ідентифікатор прошивки, контрольну суму та інші важливі параметри, що дозволяють здійснити правильне оновлення.

4. Оновлення мікропрограми з модифікацією ідентифікатора:

Після того як отримано інформацію про поточну мікропрограму, стало можливим застосувати зміни. За допомогою модифікації коду було реалізовано перенаправлення оновлення мікропрограми на інший вузол у мережі Z-Wave, що

не був оригінально передбачений для оновлення. Це дозволило зловмисникам переписати мікропрограму на шкідливу, підвищуючи ризик атак на систему.

5. Виконання FOTA на модифікованому вузлі:

Оновлення мікропрограми, яке було спрямоване на вибраний вузол за допомогою модифікації, відбулося через FOTA-процес. Після того як ідентифікатор вузла був змінений, атака стала можливою, і відповідні зміни мікропрограми було завантажено на пристрій, де виконувались шкідливі дії.

Для підтвердження теоретичної моделі атаки на реальному обладнанні була проведена серія експериментів, що включала в себе витягування та модифікацію прошивки з використанням реальних пристроїв. Зокрема, було використано флеш-пам'ять типу M25PE20 та M25PE10, що використовуються в модулях зв'язку Z-Wave, щоб вилучити вбудовану програму з пристрою, наприклад, з розумного дверного замка.

1. Витягнення прошивки:

Використовувалась методика витягнення прошивки через Raspberry Pi та програму flashrom, але флеш-пам'ять M25PE20 не підтримувала повністю командне керування, тому був використаний додатковий інструмент Arduino Mega (ATmega2560), що дозволив успішно витягти прошивку. Витягнуті дані включали важливу інформацію, як-от домашній ідентифікатор мережі та мережевий ключ, які зберігалися в пам'яті.

2. Аналіз витягнутого коду:

За допомогою інструменту HeX були проаналізовані двійкові файли, отримані з флеш-пам'яті. Під час цього аналізу були виявлені важливі параметри, такі як домашній ідентифікатор та мережевий ключ, що підтверджує можливість витоку чутливих даних в результаті успішної атаки.

3. Створення підробленого файлу прошивки:

На основі отриманих даних був створений двійковий файл з підробленим підписом. Цей файл успішно встановлювався на пристрої, що підтвердило

можливість зміни мікропрограми в пристроях Z-Wave, навіть без автентифікації чи інших заходів безпеки.

Реалізація такої атаки може призвести до значних безпекових уразливостей у мережах Z-Wave. Зловмисник, що здобув контроль над процесом FOTA, може отримати доступ до конфіденційних даних, таких як мережеві ключі, а також отримати можливість переписати мікропрограми на шкідливі, що може привести до:

- Виконання несанкціонованих операцій на пристроях.

- Втрата контролю над пристроями, зокрема смарт-замками чи іншими пристроями розумного будинку.

- Можливість перехоплення або зміни даних, що передаються в мережі.

- Поширення шкідливого коду через оновлення мікропрограм на інші пристрої в мережі.

Виявлення вразливості FOTA в Z-Wave показує необхідність вжиття додаткових заходів безпеки при реалізації бездротових оновлень мікропрограм. Для захисту системи від подібних атак необхідно забезпечити:

1. Шифрування та автентифікація: Використання надійних методів шифрування для передачі мікропрограм і автентифікації даних під час оновлень.

2. Перевірка підпису: Встановлення перевірки цифрових підписів для всіх файлів мікропрограм перед їх завантаженням на пристрої.

3. Контроль доступу: Обмеження доступу до інтерфейсів оновлення мікропрограм, особливо для пристроїв, що знаходяться в мережах з високим рівнем безпеки.

Таким чином, атака FOTA через протокол Z-Wave може мати серйозні наслідки для безпеки розумних будинків, але з правильними заходами захисту можна значно зменшити ризики таких атак.

Дистанційне керування режимом додавання**.**Режим додавання — це режим роботи, який пристрої Z-Wave використовують для з'єднання між собою в мережі.

Щоб два пристрої, такі як контролер і вузол, могли з’єднатися, обидва повинні перебувати в цьому режимі. Зазвичай більшість вузлів можуть увійти в режим додавання, якщо натискати кнопку на вузлі протягом більше 3 секунд. Однак для зручності користувачів, багато контролерів підтримують можливість зміни режиму додавання дистанційно, без фізичного натискання кнопок, через спеціальні програми або інтерфейси користувача.

Атака віддаленого керування режимом додавання полягає в тому, що зловмисник примусово перемикає контролер в режим додавання (режим сполучення) без фізичного доступу до пристрою. Це дає змогу атакуючому перехопити мережевий ключ жертви та приєднати до мережі вузол, що контролюється зловмисником, таким чином ставши частиною мережі Z-Wave жертви.

Для демонстрації такого сценарію атаки був підготовлений тестовий стенд, зображений на рис. 2.14. У цьому сценарії атаки припускається, що маршрутизатор жертви знаходиться між сервером і контролером жертви. Зловмисник, маючи доступ до маршрутизатора та знаючи ідентифікатор адміністратора і пароль маршрутизатора, може виконати атаку Man-In-The-Middle (MITM).

Кроки атаки:

1. Отримання доступу до маршрутизатора жертви: Зловмисник використовує свої знання ідентифікатора та пароля для доступу до маршрутизатора, що розташований між сервером і контролером.

2. Атака MITM: Зловмисник виконує атаку MITM, перехоплюючи та модифікуючи пакети, що передаються між сервером і контролером жертви. За допомогою цього методу зловмисник може проаналізувати з’єднання та знайти корисну інформацію.

3. Перемикання контролера в режим додавання: Після того, як зловмисник отримав доступ до потрібних даних, він може надіслати команду, яка змінює

режим роботи контролера на режим додавання. Це дозволить зловмиснику додати свій вузол до мережі жертви.

4. Перехоплення мережевого ключа: Під час цього процесу зловмисник може також отримати мережевий ключ жертви, що дає йому можливість управляти іншими пристроями в мережі, а також передавати шкідливі команди.

Потенційні наслідки атаки:

- Перехоплення контролю над мережею: Зловмисник може отримати доступ до всіх пристроїв в мережі Z-Wave жертви, контролюючи їх або відправляючи небажані команди.

- Витік конфіденційної інформації: Отримавши мережевий ключ, зловмисник може розшифрувати захищену інформацію, яка передається по мережі.

- Шкідливі дії: Зловмисник може додавати свої власні пристрої до мережі, отримувати доступ до сенсорів, замків або інших важливих пристроїв, що значно підвищує ризик злому або використання мережі у власних інтересах.

Заходи для запобігання атаці:

- Посилення захисту маршрутизаторів: Використання складних паролів та двофакторної автентифікації для доступу до маршрутизаторів.

- Шифрування переданих даних: Використання сучасних методів шифрування для захисту даних, що передаються між пристроями в мережі.

- Виявлення аномалій в мережевому трафіку: Встановлення систем виявлення вторгнень для аналізу та блокування підозрілих команд, які можуть вказувати на спробу атаки.

Цей сценарій атакує основи безпеки Z-Wave, дозволяючи зловмисникам без фізичного доступу до пристроїв здійснити контроль над мережею, що підкреслює необхідність ретельного контролю та захисту таких систем.

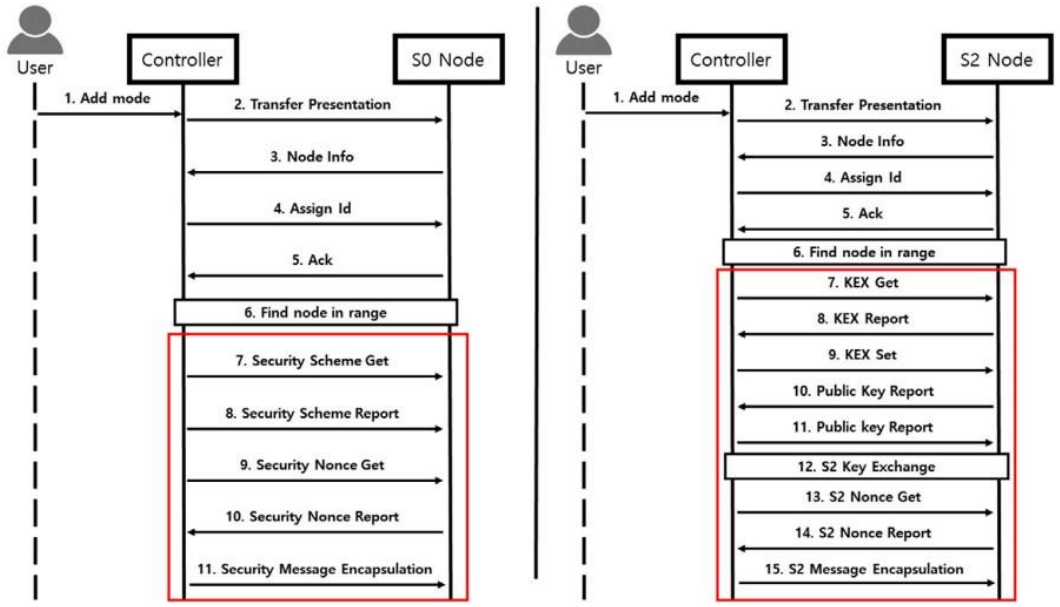


Рисунок 2.21 – Відмінності зв'язку протоколу S0 та S2

Після успішної відправки пакету для активації режиму додавання зловмисник може спостерігати, що контролер дійсно переходить у цей режим та надсилає відповідний кадр Z-Wave для створення пари з вузлом. Це означає, що контролер намагається здійснити сполучення з новим пристроєм, що дає зловмиснику можливість підключити свій вузол. У результаті:

1. Ініціація сполучення: Контролер надсилає запит на створення пари, який включає мережевий ключ.

2. Отримання мережевого ключа: Зловмисник отримує цей ключ разом з пакетом, що дозволяє йому підключити свій пристрій до мережі жертви.

3. Підключення пристрою зловмисника: Використовуючи мережевий ключ, зловмисник може додавати свої пристрої до мережі та контролювати їх.

4. Порушення безпеки: Мережевий ключ дозволяє зловмиснику дешифрувати комунікацію та маніпулювати іншими пристроями в мережі.

Ця атака може призвести до порушення безпеки, оскільки зловмисник отримує контроль над пристроями мережі.

Аналіз Z-Wave S2**.** Протокол Z-Wave S2 був проаналізований для виявлення вразливостей та оцінки рівня безпеки. Це дозволило порівняти два основних класи безпеки — S0 та S2, що відрізняються механізмами шифрування та обміну ключами між контролером і вузлом, особливо в контексті віддаленого додавання. Різниця між ними полягає в тому, що протокол S2 включає більш складний механізм безпеки для обміну ключами, який застосовується під час запиту на додавання нового вузла до мережі [19].

Протокол S2 значно покращує безпеку порівняно з S0, зокрема через використання сучасних алгоритмів шифрування для захисту даних, що передаються між пристроями. Однак навіть у такому випадку Z-Wave S2 має свої вразливості.

Виявлені вразливості:

1. Вразливість MITM (Man-in-the-Middle): Ця вразливість може бути використана зловмисником для перехоплення та модифікації даних між контролером та вузлом під час обміну ключами, що дозволяє здійснювати віддалене керування процесом додавання пристроїв до мережі.

2. DoS (Denial of Service) атака: Незважаючи на підвищений рівень захисту, протокол S2 все одно не є надійно захищеним від атак типу DoS. Це стало можливим через те, що основна відмінність між протоколами S2 та S0 полягає лише в методах обміну ключами та механізмах шифрування. Тому методи атаки на обидва протоколи є схожими, що робить пристрої, які використовують S2, вразливими до DoS-атак.

1. Уразливість протоколу S2: Хоча протокол S2 використовує більш високий рівень шифрування, його вразливості до атак MITM і DoS залишаються значними. Зловмисники можуть використовувати ці вразливості для здійснення атаки на мережу Z-Wave, що ставить під загрозу безпеку пристроїв.

2. Необхідність вдосконалення протоколу безпеки: Для досягнення ще більш високого рівня захисту слід звернути увагу на удосконалення алгоритмів

шифрування і механізмів автентифікації, щоб зменшити ризик атак на пристрої, що працюють з Z-Wave.

Таблиця 2.1 надає детальний огляд застосування протоколів S0 та S2 в кожному сценарії, що допомагає зрозуміти специфіку їх використання та наявні вразливості в контексті різних типів атак.

Таблиця 2.1 - Застосування протоколів S0 та S2 для кожного сценарію.

Сценарії	S0	S2
Сценарій 1: DoS	O	O
Сценарій 2: FOTA	O	*
Сценарій 3: Режим віддаленого додавання	*	*

2.3 Експерементальне дослідження

Для реалізації сценаріїв атаки Z-Wave було створено спеціалізований випробувальний стенд, що включає реальні пристрої. На рис. 2.22 зображено структуру стенду, що використовується для аналізу вразливостей та атаки на мережу Z-Wave. У цьому стенді зловмисник має фізичний доступ до мережі жертви, що дозволяє йому підключатися до внутрішньої мережі через Wi-Fi.

Зловмисник, отримавши доступ до Wi-Fi мережі, може маніпулювати пристроями Z-Wave, аналізувати мережевий трафік та здійснювати атаки на контролери та вузли, що використовуються в мережі. Для тестування використовуються як комерційні пристрої Z-Wave, так і спеціально налаштовані тестові вузли, що дозволяють симулювати реальні умови експлуатації та відтворювати різні типи атак на систему.

Цей випробувальний стенд забезпечує надійну платформу для перевірки вразливостей Z-Wave мереж та розробки методів захисту від можливих кіберзагроз.

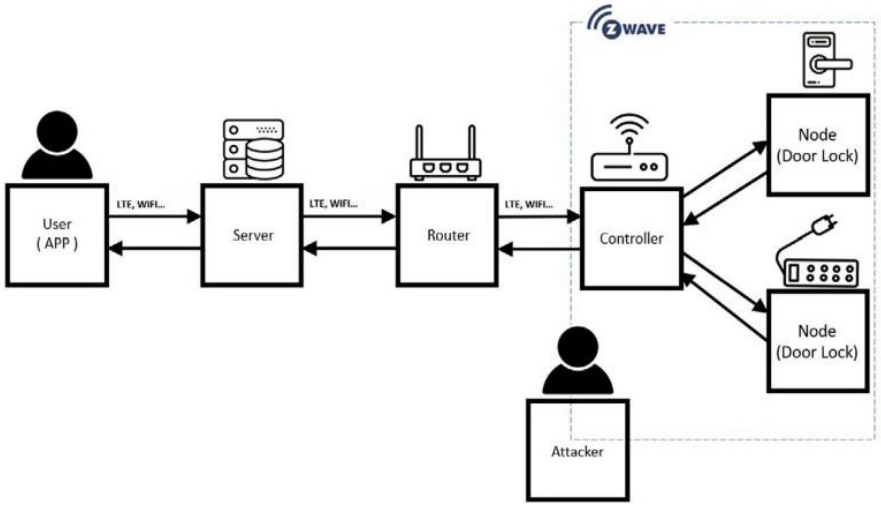


Рисунок 2.22 – Тест сценарію атаки Z-Wave

2.3.1 Результати дослідження

У першу чергу була здійснена атака MITM (Man-in-the-Middle) на цільовий маршрутизатор і контролер. Для реалізації віддаленої атаки на управління режимом додавання був налаштований проксі-сервер і клієнт, які взаємодіяли із сервером через захищену смугу шарів сокета. Використання цієї технології дозволяло зловмиснику перехоплювати і змінювати дані, що передавались між маршрутизатором і контролером.

На рис. 2.15 і 2.23 зображено стани системи до та після проведення атаки віддаленого управління режимом додавання. До атаки система працювала в нормальному режимі, де контролер не переходив у режим додавання без явного користувацького втручання. Однак після атаки, зловмисник зміг примусово перевести контролер у режим додавання, що дозволило йому підключити шкідливий пристрій до мережі.

Результати експерименту показали, що така атака дозволяє зловмиснику не лише змінювати налаштування контролера, а й перехоплювати ключі та іншу критичну інформацію, що значно послаблює безпеку мережі Z-Wave.

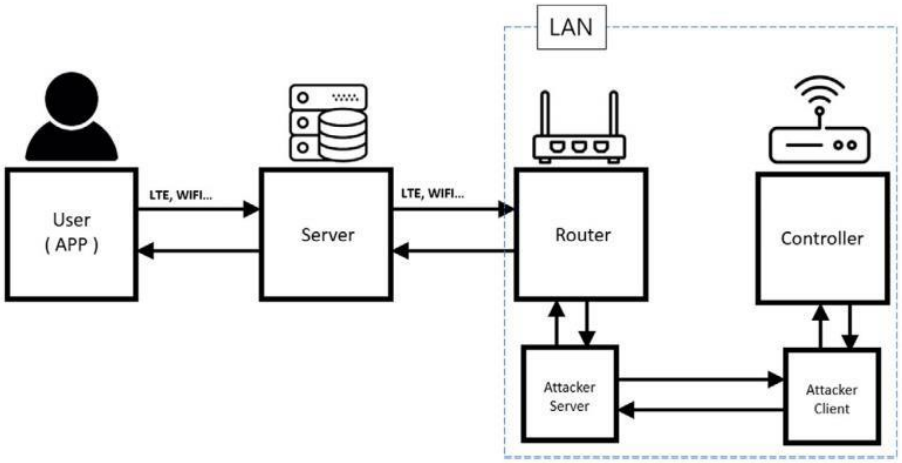


Рисунок 2.23 – Випробувальний стенд після віддаленого управління додатковим режимом

Після того, як було проведено атаки MITM між маршрутизатором і контролером, зловмисник зміг перехопити та модифікувати дані, що передавалися між ними. Для належної роботи Z-Wave пристрої повинні обмінюватися спільним мережевим ключем, що забезпечує їхню здатність до комунікації. Коли новий пристрій приєднується до мережі Z-Wave, проходить певний процес синхронізації, під час якого відбувається обмін мережевими ключами між пристроєм і контролером.

Зловмисник змусив контролер перейти в режим сполучення, надіславши йому команду, яка видавалася за запит від сервера. Завдяки цьому, пристрій зловмисника успішно з'єднався з контролером, а зловмисник отримав доступ до мережевого ключа, який забезпечував спільну роботу контролера і маршрутизатора. Рис. 2.21 відображає результат цієї атаки, в якому видно, що зловмисник отримав мережевий ключ, необхідний для подальшого зв'язку між пристроями.

За допомогою спеціально розробленого інструменту для віддаленого управління режимом додавання, зловмисник зміг змусити контролер передавати мережевий ключ, що дозволило йому підключити до мережі свій пристрій,

використовуючи ідентичний ключ Z-Wave. Це дало зловмиснику змогу віддалено управляти пристроями, що використовують стандарт S0.

Крім того, зловмисник здійснив атаку типу DoS на контролер, що дозволило йому приховати факт несанкціонованого контролю пристроїв від користувача, навіть коли пристрій був під повним контролем зловмисника. В експерименті, який був проведений десять разів, зловмисник зміг отримати повний контроль над пристроями, а середній час для цього становив 45,291 секунди. Рис. 2.25 наочно демонструє результати цього експерименту, підтверджуючи ефективність атак, що були здійснені.

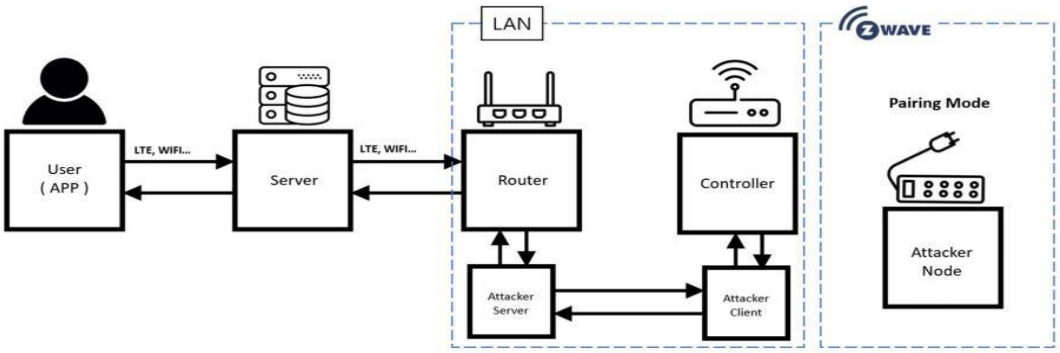
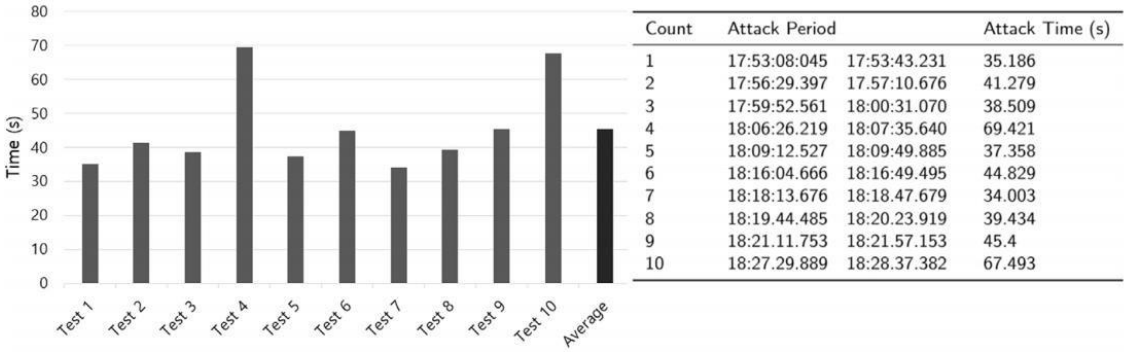


Рисунок 2.24 – Сполучення за допомогою вузла зловмисника



Count	Attack Period		Attack Time (s)
1	17:53:08:045	17:53:43.231	35.186
2	17:56:29.397	17.57:10.676	41.279
3	17:59:52.561	18:00:31.070	38.509
4	18:06:26.219	18:07:35.640	69.421
5	18:09:12.527	18:09:49.885	37.358
6	18:16:04.666	18:16:49.495	44.829
7	18:18:13.676	18:18.47.679	34.003
8	18:19.44.485	18:20.23.919	39.434
9	18:21.11.753	18:21.57.153	45.4
10	18:27.29.889	18:28.37.382	67.493

Рисунок 2.25 – Час атаки для отримання повного доступу

У цьому розділі було здійснено комплексний аналіз реальних пристроїв Z-Wave з використанням методів моделювання загроз, зокрема моделей DFD (Data Flow Diagram) та STRIDE. Було досліджено три основні вектори атак на протокол

Z-Wave: DoS (Denial of Service), FOTA (Firmware Over-the-Air) та атаки дистанційного керування режимом додавання. В експериментальних умовах, що імітують реальні сценарії, було поєднано кілька векторів атак. Це дозволило виявити, що їх комбінування може привести до серйозних наслідків, зокрема до критичних загроз для безпеки мешканців розумних будинків.

Попри підвищення рівня безпеки за допомогою переходу з протоколу S0 на S2, ще існує проблема з безпекою мережевого ключа. Хоча S2 використовує більш потужне шифрування, зловмисники можуть викрасти мережевий ключ, який контролер використовує для взаємодії з кількома вузлами. Важливим кроком для покращення безпеки є впровадження розподілу унікальних симетричних ключів для кожного вузла, що ускладнить зловмиснику доступ до кількох пристроїв одночасно.

Для додаткового посилення захисту, слід періодично змінювати попередньо спільний ключ випадковим чином для кожного пристрою. Це дозволить знизити ризик довгострокового контролю зловмисником навіть у випадку перехоплення ключа. Крім того, підключений пристрій повинен мати можливість змінювати цей ключ через механізм FOTA та використовувати PRNG (Псевдовипадковий генератор чисел) для забезпечення додаткового рівня захисту від можливих атак.

Зазначено, що протокол Z-Wave не використовує сегментацію кадрів, що є ще однією вразливістю. Відключення цієї функції на пристроях, де вона не є необхідною, може допомогти ефективно запобігти атакам типу DoS, які намагаються знищити або призупинити роботу мережі через перевантаження її компонентів.

3 РОЗРОБКА СИСТЕМИ АВТОМАТИЗАЦІЇ РОЗУМНОГО БУДИНКУ

3.1 Вибір середовища розробки та опис архітектури

Для створення інтерфейсу для інтеграції в систему розумного будинку Z-Wave було обрано бібліотеку OpenZWave. Це відкрита міжплатформна бібліотека, яка дозволяє додавати підтримку пристроїв Z-Wave до додатків без необхідності глибоких знань про протокол Z-Wave [20].

Z-Wave є бездротовим протоколом, що використовує технологію мереж для забезпечення зв'язку між пристроями на великих відстанях, навіть через бар'єри. Це дозволяє пристроям з низьким споживанням енергії зв'язуватися між собою, передаючи повідомлення з одного пристрою на інший. Важливо зауважити, що деякі пристрої Z-Wave, зокрема ті, що працюють на батареях, можуть не завжди бути активними і не здатні пересилати повідомлення, поки не отримають команду з центру управління.

Всі пристрої Z-Wave в мережі відомі як "вузли". Мережа може включати до 232 вузлів, але для більшої кількості пристроїв можуть знадобитися додаткові контролери. OpenZWave підтримує кілька контролерів, але сам по собі не може об’єднати їх в єдину мережу. Цю функціональність забезпечує додаток.

Вузли Z-Wave поділяються на контролери та підлеглі пристрої. Контролери можуть бути у вигляді пультів дистанційного керування або комп'ютерних інтерфейсів, а підлеглі пристрої, такі як датчики руху або вимикачі, виконують функції, пов'язані з контролем і спостереженням.

Уся комунікація між ПК і пристроями мережі Z-Wave здійснюється асинхронно. Наприклад, датчики руху можуть перебувати в режимі сну для економії енергії, що може спричинити затримку в обробці запитів або навіть відсутність відповіді. Це означає, що при запитах на зміну стану пристроїв відповіді можуть бути отримані не відразу, іноді з великими затримками або й зовсім не надійти. З цієї причини OpenZWave не завжди гарантує миттєву

відповідь на запити, а замість цього використовує систему повідомлень для асинхронної комунікації між додатком і пристроєм.

Крім того, Z-Wave мережі є динамічними — пристрої можуть бути додані або видалені в будь-який момент. Це робить необхідним обробку змін у мережі в реальному часі, що також підтримується через систему сповіщень.

Для розробки програми, що працює на основі OpenZWave, потрібно враховувати, що:

– Зв'язок між пристроями є асинхронним і не завжди гарантовано надійним. Неможливо покладатися на отримання відповіді на запит відразу.

– Мережа може змінюватися, тому програма повинна динамічно адаптуватися до додавання або видалення пристроїв, а інтерфейс користувача — оновлюватися відповідно до змін у мережі.

– У випадку використання кількох контролерів, програма має підтримувати управління декількома мережами одночасно.

З огляду на це, була вибрана інтеграційна платформа IOBroker, що надає багатофункціональну та гнучку панель управління. IOBroker є модульною платформою, що складається з численних компонентів, кожен з яких виконує певні функції. Централізоване управління здійснюється через контролер js-controller, який відповідає за зв'язок і обробку даних між модулями. Програма побудована на основі адаптерів, що дозволяють додавати нові функціональні можливості лише за необхідності.

Перевагами використання IOBroker є можливість легко налаштовувати та керувати пристроями, що використовують різні протоколи, та підтримка інтеграцій з іншими системами через зручний веб-інтерфейс.

При встановленні нового адаптера через адміністратора, файли адаптера спочатку завантажуються з Інтернету і зберігаються на сервері в локальній файловій системі. Якщо адаптер необхідно запустити, на цьому етапі генерується окремий екземпляр адаптера, що виконується як окремий процес. Кожен

екземпляр адаптера може бути індивідуально налаштований, зупинений або запущений незалежно від інших екземплярів або адміністратора. Це дає можливість управляти адаптерами гнучко, зокрема проводити їх налаштування, перезапуск або зупинку без впливу на інші частини системи. Кожен екземпляр працює в своєму окремому процесі, який взаємодіє з js-контролером ioBroker, обробляючи запити та обмінюючись даними в фоновому режимі.

У випадку використання системи Multihost, яка включає кілька серверів ioBroker, екземпляри адаптерів можуть бути розподілені між різними серверами. Це дозволяє ефективно розподіляти навантаження, а також додавати нові сервери для збільшення потужності або підключати додаткове обладнання безпосередньо до серверів. Наприклад, можна підключати порти введення/виведення або USB-пристрої безпосередньо до серверів, на яких працюють відповідні адаптери. Це підвищує ефективність роботи всієї системи і дозволяє адаптувати інфраструктуру під конкретні вимоги.

Зв'язок між адаптерами, js-контролерами, базами даних і веб-інтерфейсами в ioBroker забезпечується за допомогою декількох з'єднань TCP/IP. Це дозволяє забезпечити стабільний і безпечний обмін даними між компонентами системи. Обмін даними може здійснюватися як у вигляді простого тексту, так і в зашифрованому вигляді, залежно від обраних налаштувань безпеки. Таким чином, ioBroker забезпечує високий рівень конфіденційності і цілісності переданих даних, що є важливим аспектом при інтеграції з іншими системами та мережами.

ioBroker і його адаптери розроблені здебільшого на мові програмування JavaScript, що дозволяє легко інтегрувати систему з іншими веб-технологіями і платформами. Оскільки для виконання JavaScript потрібно середовище виконання, ioBroker використовує Node.js — потужну платформу, яка підтримує асинхронне виконання коду і дозволяє ефективно працювати з великою кількістю одночасних з'єднань. Node.js доступний на різних операційних системах, таких як Linux, Windows і macOS, що забезпечує високу гнучкість у використанні ioBroker в

різних середовищах. Крім того, для встановлення і керування компонентами ioBroker використовується менеджер пакетів npm, який дозволяє зручно інсталювати необхідні бібліотеки і адаптери без необхідності ручного завантаження файлів.

VIS (Visualization) — це візуалізаційне рішення для користувачів ioBroker, яке дає можливість створювати графічні інтерфейси для моніторингу та керування пристроями в системі. Завдяки VIS, користувач може розробити інтерфейс користувача, який ідеально відповідає вимогам конкретної інсталяції або персональних переваг. Інтерфейс може включати різні елементи, такі як графіки, кнопки, слайдери, індикатори тощо, що дозволяє легко відображати інформацію про стан системи [21].

Використовуючи редактор візуалізації, користувачі можуть безпосередньо створювати макети інтерфейсу, додаючи та налаштовуючи віджети. Кожен віджет може бути змінений залежно від вимог користувача — наприклад, налаштувати вигляд або функціональність за допомогою властивостей, які відображаються в панелі праворуч від редактора. Віджети можуть бути настроєні для роботи з різними типами даних, від простих текстових повідомлень до складних графічних відображень. Це дозволяє створювати високофункціональні та інтуїтивно зрозумілі інтерфейси для користувачів, які мають різні рівні досвіду роботи з технічними системами.

Таким чином, система VIS дає змогу користувачам ioBroker максимально адаптувати відображення даних і управління пристроями до своїх специфічних потреб, що є одним із основних переваг цієї платформи.

Рисунок 3.1 — VIS візуалізація в ioBroker

За допомогою редактора VIS ви можете створювати повністю індивідуалізовані візуалізації для вашого розумного будинку, які будуть відповідати саме вашим вимогам та уявленням. Візуальні елементи інтерфейсу можна налаштовувати на свій розсуд, а завдяки широкому вибору віджетів, які можна комбінувати, ви можете створювати інтерфейс, що максимально відповідає функціональним потребам. Більше того, доступні адаптери, що дозволяють розширювати можливості віджетів, додаючи нові функціональності та інтеграції з іншими пристроями чи сервісами.

Однак процес впровадження таких візуалізацій може бути досить складним, особливо коли йдеться про більш комплексні рішення в системах розумного будинку. Це зазвичай вимагає значного часу на налаштування та тестування всіх компонентів, а також розробку інтерфейсу, що відповідатиме специфічним вимогам користувача. Крім того, важливо відзначити, що візуалізація адаптована лише до конкретного кінцевого пристрою, на якому вона буде відображатися. Наприклад, інтерфейс, розроблений для великого екрану комп'ютера або планшета, не буде автоматично адаптуватися до мобільних пристроїв, через

відсутність адаптивного дизайну для різних типів кінцевих пристроїв. Це може стати обмеженням при використанні на різних платформах або пристроях, що вимагає додаткових зусиль для створення оптимізованих версій інтерфейсу.

2. Material UI

З використанням Material UI можна створювати динамічні візуалізації, де компоненти, такі як кімнати або функції, автоматично зчитуються з конфігурації ioBroker (списки та об'єкти). Це надає користувачеві інтерфейс для легкого налаштування управління будинком, без необхідності створювати складну структуру візуалізації у VIS.

За допомогою візуалізації інтерфейсу Material UI ви можете швидко та ефективно створити сучасний інтерфейс для управління розумним будинком. Основою для цієї візуалізації є списки та метадані об'єктів у ioBroker, що забезпечує зручність і гнучкість у побудові інтерфейсу.

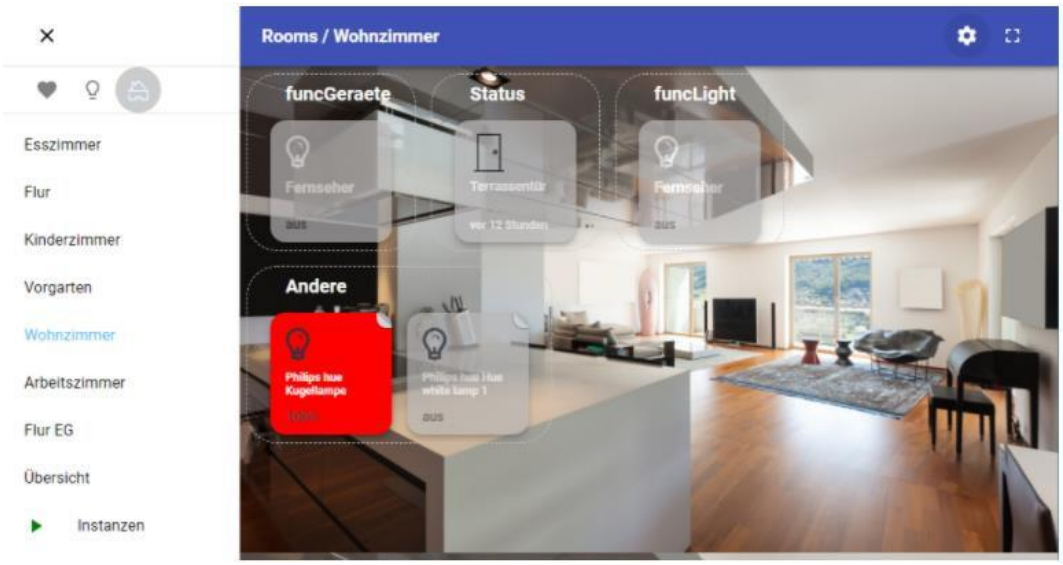


Рисунок 3.2 – Material UI візуалізація в ioBroker

У візуалізацію автоматично додаються пускачі та датчики на основі налаштувань приміщень та їх торгового призначення (функцій). Однак, ви також маєте можливість додавати власні віджети для ще більшої кастомізації інтерфейсу

відповідно до ваших потреб. На момент написання статті адаптер все ще проходив стадію тестування, але вже має досить великий функціонал і може бути використаний для створення власної візуалізації.

Найкращі результати досягаються, коли структура об'єкта добре організована, оскільки це є основою для створення зручної і ефективної візуалізації. Гарно структуровані дані та правильне їх використання у Material UI дозволяють створити інтерфейс, що буде не тільки функціональним, але й зручним для користувача.

3. iQontrol

iQontrol — це поєднання Apple Homekit і інтерфейсу Material UI. Візуалізацію можна легко налаштувати за допомогою готових шаблонів. Після цього різні пристрої, підключені до ioBroker, можуть бути додані до відповідних секцій інтерфейсу (наприклад, освітлення чи кондиціонери).

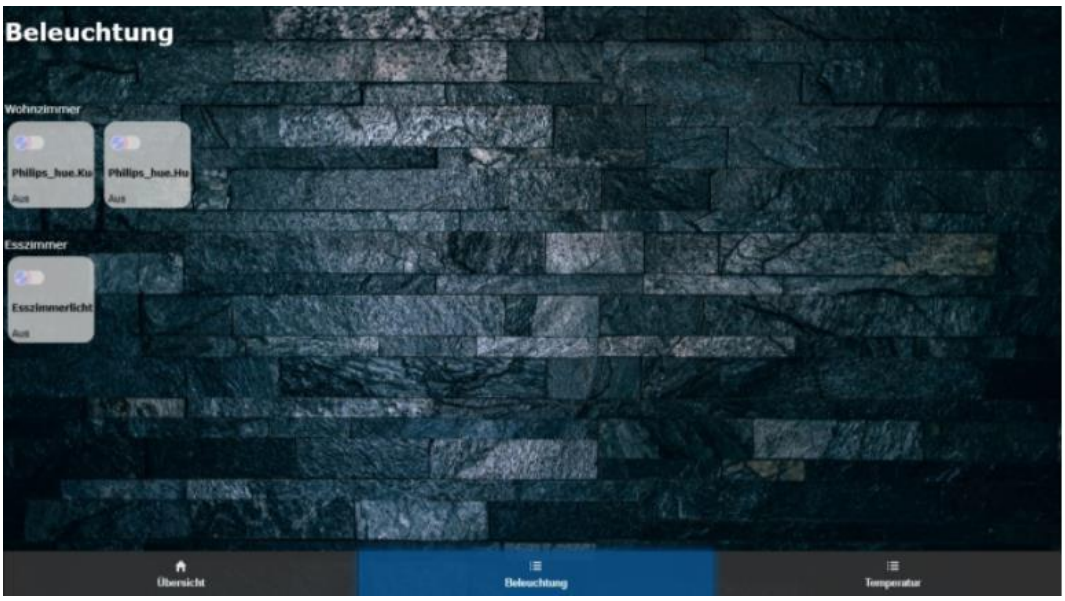


Рисунок 3.3 – iQontrol візуалізація в ioBroker

Після тестування адаптера я зміг побудувати частину своєї системи автоматизації будинку всього за годину. Особисто мені дуже сподобався підхід, який застосовує цей адаптер, і я бачу великий потенціал у його використанні.

Адаптер є дуже простим для розуміння і використання, що робить його ідеальним варіантом для початківців.

Зручний інтерфейс та простота налаштування дозволяють швидко створювати унікальні візуалізації для розумного будинку, навіть без глибоких знань у програмуванні. Цей адаптер поєднує функціональність з легкістю використання, що робить його дуже привабливим для тих, хто хоче швидко налаштувати свою систему автоматизації, не витрачаючи багато часу на складне програмування чи налаштування інтерфейсу.

4. HABPanel

HABPanel дозволяє швидко і легко налаштувати інтерфейси візуалізації для ioBroker. Спочатку розроблене для openHAB, воно було пізніше адаптоване для використання в ioBroker. На момент написання статті (серпень 2019 року) адаптер ще не був доступний для виробничого середовища.

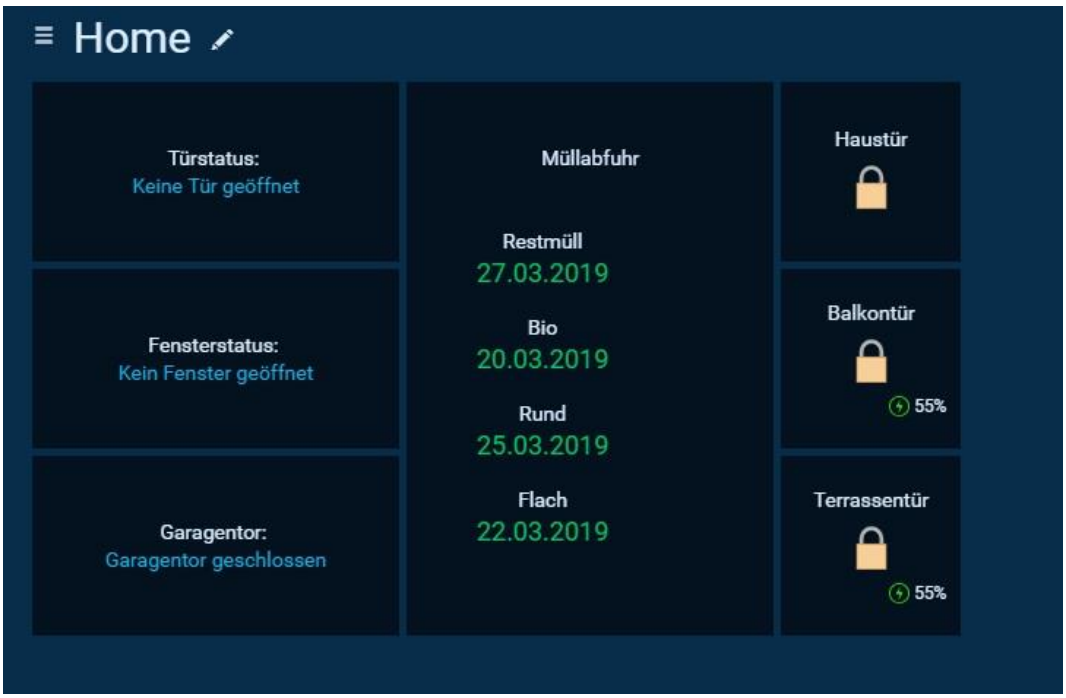


Рисунок 3.4 – HABPanel візуалізація в ioBroker

З HABPanel ви можете швидко реалізувати свою візуалізацію завдяки інтуїтивно зрозумілій структурі інтерфейсу. Адаптер пропонує велику кількість віджетів, які можна використовувати для створення користувацьких візуалізацій. Крім того, для розробки власних віджетів або імпортування віджетів з спільноти можна застосовувати знання HTML, CSS та AngularJS.

Незважаючи на багато переваг, одним із недоліків є відсутність адаптивного веб-дизайну, що може ускладнити використання інтерфейсу на різних пристроях. В цілому, HABPanel є потужним інструментом для налаштування візуалізацій, однак цей аспект обмежує його гнучкість для різних екранів і пристроїв.

5. Lovelace-UI

Lovelace-UI – це потужний інструмент для створення візуальних інтерфейсів в ioBroker, що спершу був розроблений для Home Assistant, але згодом став доступним і для платформи ioBroker. На момент написання статті (серпень 2019 року) адаптер ще не був повністю готовий до використання в реальному середовищі, однак його потенціал вже був очевидний [22].

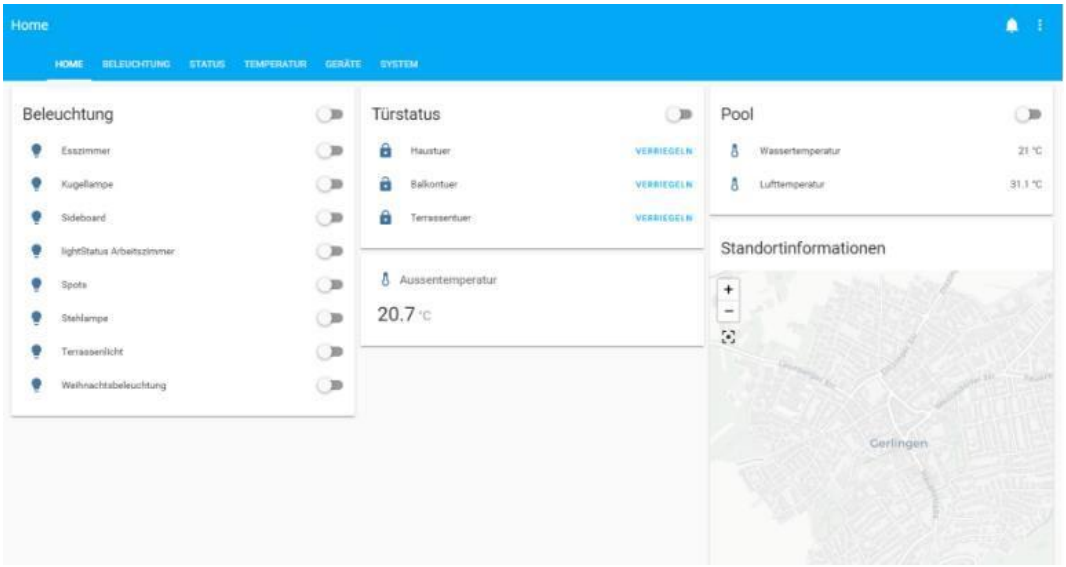


Рисунок 3.5 – Lovelace-UI візуалізація в ioBroker

Lovelace дозволяє створювати стильні, інтуїтивно зрозумілі інтерфейси за короткий час. Однак для коректного функціонування необхідно налаштувати певні аспекти, такі як активація датчиків та виконавчих механізмів в об'єктах. Крім того, важливо, щоб система була налаштована так, щоб приміщення та функціональні зони були чітко визначені і доступні для коректної роботи адаптера.

Що мене дійсно вразило в Lovelace-UI, це додаткові функції, які спрощують управління розумним будинком, зокрема таймери та система сповіщень. Ці можливості дозволяють не тільки спростити автоматизацію, але й зробити її більш інтерактивною та адаптованою до потреб користувача. Вони надають можливість гнучко налаштовувати систему, що особливо важливо для людей, які хочуть створити персоналізований досвід роботи зі своїм розумним будинком.

Незважаючи на те, що адаптер ще не був остаточно доведений до виробничого стану, я вже бачу в ньому великий потенціал для власної візуалізації. Я навіть вирішив перенести всі свої існуючі налаштування на цей новий адаптер, адже його гнучкість і зручність просто вражають. Для мене це справжній прорив у роботі з розумними будинками, оскільки він дозволяє створити більш зручний і інтуїтивно зрозумілий інтерфейс, а також забезпечує більше контролю та налаштувань.

6. Tileboard

Tileboard – це ще один адаптер для візуалізації, який був адаптований з платформи Home Assistant для використання з ioBroker. Станом на серпень 2019 року адаптер не був готовий для повноцінного використання в реальному середовищі, що означає, що його інтеграція в ioBroker була на ранній стадії розвитку.

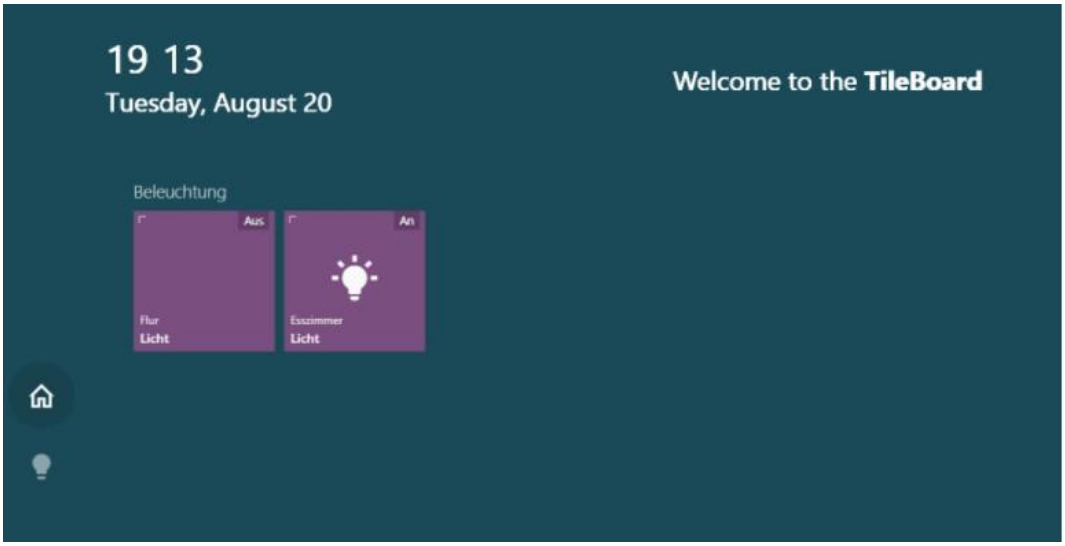


Рисунок 3.6 – Tileboard візуалізація в ioBroker

У Tileboard відсутній графічний інтерфейс для налаштування, тому весь процес конфігурації здійснюється через редагування JavaScript-файлу. Цей файл є основним інструментом для налаштування адаптера: в нього додаються окремі сторінки, групи, сутності та налаштовуються їх властивості. Для користувача це може бути як перевагою, так і недоліком, оскільки вимагається певний рівень знань в програмуванні.

Попри відсутність готового графічного інтерфейсу для налаштування, цей адаптер має значний потенціал для досвідчених користувачів, які хочуть мати повний контроль над своєю візуалізацією і готові самостійно налаштовувати різні компоненти системи. Гнучкість конфігурації через редагування JavaScript дозволяє створювати надзвичайно індивідуалізовані інтерфейси, але це також може стати бар’єром для новачків, оскільки без певних технічних навичок адаптер може бути важким для розуміння та використання.

У цілому, Tileboard є хорошим варіантом для користувачів, які мають досвід роботи з Home Assistant і хочуть переносити свою візуалізацію в ioBroker. Однак, поки адаптер не знаходиться в продуктивному середовищі, його застосування може бути обмеженим.

7. Node-RED візуалізація

Node-RED пропонує вбудовану візуалізацію, яка дозволяє створювати власні інформаційні панелі для управління різними компонентами системи автоматизації. Основна концепція візуалізації в Node-RED полягає в використанні вкладок та груп, які можна налаштовувати безпосередньо в середовищі Node-RED. Цей підхід дозволяє легко організувати різні елементи управління, об’єднуючи їх у єдину інформаційну панель.

У Node-RED візуалізація будується через визначення вкладок для різних категорій, таких як датчики, виконавчі механізми, освітлення тощо. Потім ці елементи збираються у вигляді вузлів (nodes) на інформаційній панелі. Вузли на панелі можуть включати в себе різні типи елементів управління, такі як кнопки, повзунки або індикатори стану, що дозволяють користувачу взаємодіяти з системою та переглядати її поточний стан.

Однією з основних переваг Node-RED є те, що вона надає користувачам гнучкість у створенні персоналізованих візуалізацій. Вони можуть обирати, які елементи виводити на панель, а також налаштовувати їхній вигляд і функціональність залежно від своїх потреб. Використовуючи підхід на основі потоків, користувач може створювати цілі інформаційні панелі для різних сценаріїв автоматизації.

Однак, варто зазначити, що для досягнення повної функціональності та налаштування візуалізації потрібен певний рівень знайомства з Node-RED та її поточною екосистемою. Для новачків цей процес може бути складним через необхідність створювати складні потоки даних і взаємодії між вузлами.

У загальному підсумку, Node-RED є потужним інструментом для створення кастомізованих інформаційних панелей, але вимагає часу на освоєння для досягнення найкращих результатів.

8. Jarvis

Jarvis є новим адаптером візуалізації для ioBroker, який використовує фреймворк Material UI для веб-розробки. Цей адаптер має на меті надати користувачам можливість створювати сучасні, адаптивні інформаційні панелі для управління пристроями та точками даних в системі автоматизації. Візуалізація Jarvis використовує гнучку структуру пристроїв і модулів, що дозволяє інтегрувати широкий спектр пристроїв і точок даних у єдину панель управління.

Візуалізація налаштовується через вкладки та стовпці, що дозволяє легко організовувати різні групи пристроїв і їх параметри. Користувач може створювати категорії для кожного пристрою чи функції, що допомагає краще організувати систему та забезпечити інтуїтивно зрозумілий інтерфейс для користувачів.

Однією з ключових особливостей Jarvis є можливість інтеграції пристроїв через віджети. Віджети виступають як елементи, які можна вставляти в інтерфейс візуалізації для представлення різних пристроїв, таких як лампи, термостати, датчики, або навіть групи пристроїв. Це дає змогу створювати зручні та персоналізовані панелі управління, що відповідають потребам користувача.

Jarvis також дозволяє налаштовувати вигляд та функціональність цих віджетів, щоб вони відповідали конкретним вимогам користувача. Завдяки гнучкості та інтеграційним можливостям адаптера, можна створити візуалізацію, яка є як простий, так і надзвичайно функціональний інтерфейс для управління будь-якими аспектами розумного дому.

Підсумовуючи, Jarvis є потужним та гнучким інструментом для створення сучасних візуалізацій в ioBroker, який дозволяє легко налаштовувати інтерфейс і інтегрувати різноманітні пристрої через зручні віджети.

Тип візуалізації	Складність	Власний дизайн	Власні віджети	Комплексність	Адаптивний дизайн	Спільнота
VIS	Високий	Так	Так	Високий	Ні	Дуже добре
Material UI	Середній	Ні	Ні	Середній	Так	Добре
iQuontrol	Середній	Ні	Ні	Середній	Так	-
HABPanel	Середній	Ні	Так	Середній	Ні	-
Lovelace-UI	Середній	Так	Так	Середній	Так	-
TileBoard	Середній	Так	Так	Середній	Так	-
Node-RED	Високий	Так	Так	Високий	Так	Добре
Jarvis	Середній	Ні	Ні	Середній	Так	Добре

Таблиця 3.1 – Порівняння середовищ візуалізації ioBroker.

Проаналізувавши всі варіанти візуалізацій для ioBroker, я визначив їхні ключові особливості та переваги. Кожен адаптер має свої сильні та слабкі сторони, і вибір залежить від конкретних вимог до адаптивності та гнучкості візуалізації.

1. VIS:

- Переваги: Потужний інструмент із максимальною свободою в дизайні та технічних можливостях.

- Недоліки: Неможливість адаптації до різних роздільних здатностей дисплеїв, що обмежує його застосування на різних пристроях.

2. HabPanel:

- Цей адаптер виявився менш зручним через проблеми з масштабуванням і обмежену кількість віджетів. Це стало поштовхом для переходу до інших варіантів.

3. Material-UI та Lovelace:

- Обидва адаптери підтримують адаптивний дизайн, що дозволяє створювати візуалізації, які коректно відображаються на різних пристроях.

- Lovelace потребує більше зусиль для налаштування точок даних, але дозволяє досягти більшої точності та гнучкості.

- Material-UI простіший у використанні, але має свої особливості в підходах до організації точок даних.

Отже, для складних візуалізацій, де важлива гнучкість і деталізація, я б рекомендував VIS. Для більш універсальних і адаптивних рішень на різних пристроях найкраще підходять Material-UI та Lovelace.

3.2 Архітектура та структура проекту

Adapter Admin є ключовим компонентом для управління всією інсталяцією ioBroker. Цей адаптер забезпечує доступ до системи через веб-інтерфейс, який можна відкрити за адресою: <IP-адреса сервера>:8081. Adapter Admin встановлюється автоматично під час інсталяції ioBroker, тому додаткової установки цього компонента не потрібно.

Основні функції веб-інтерфейсу Adapter Admin включають:

1. Загальносистемні налаштування – можливість налаштовувати основні параметри всієї системи.

2. Установка додаткових адаптерів та примірників – ви можете додавати нові адаптери і їх примірники для розширення можливостей ioBroker.

3. Конфігурація примірників – доступ до налаштувань та параметрів встановлених примірників для подальшої оптимізації роботи системи.

4. Огляд об'єктів – можливість перевіряти і контролювати різні об'єкти в системі, що підключені до ioBroker.

5. Огляд стану об'єктів – перевірка поточного стану всіх об'єктів у системі для моніторингу та керування ними [23].

6. Адміністрування користувачів та груп – налаштування прав доступу для користувачів та організація груп для ефективного управління доступом.

7. Доступ до лог-файлів – можливість переглядати логи системи для діагностики та виявлення потенційних проблем.

8. Адміністрування господарів – контроль за господарськими компонентами системи.

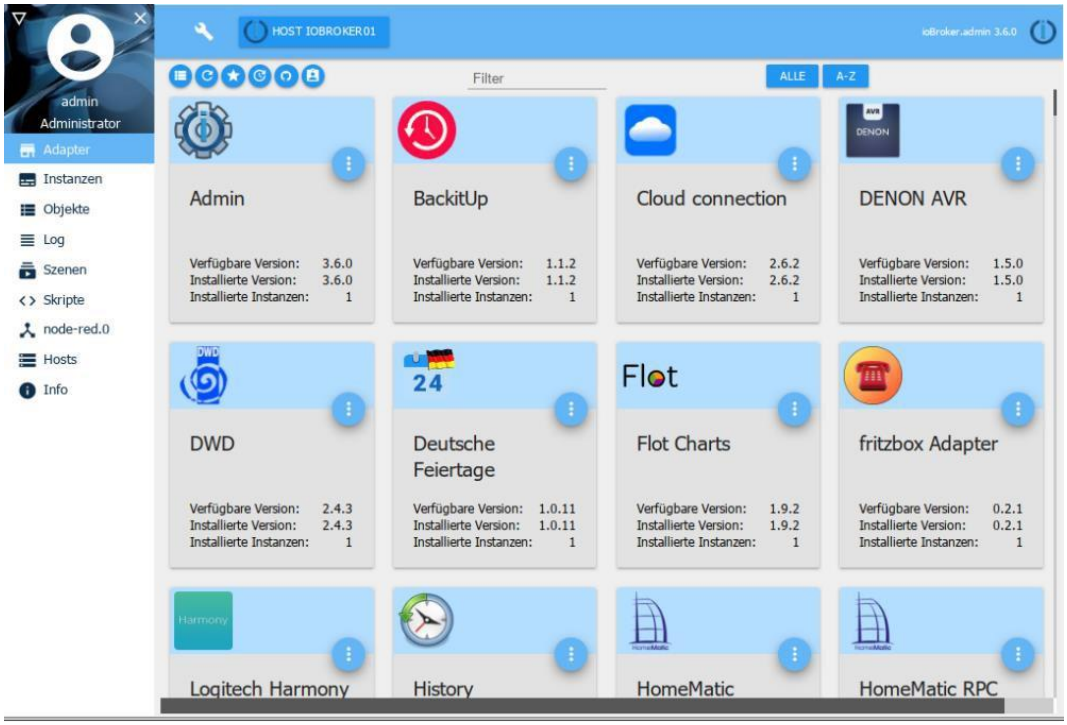


Рисунок 3.7 – Архітектура ioBroker

Веб-інтерфейс, наданий Adapter Admin, дозволяє зручно керувати всіма аспектами роботи ioBroker, включаючи налаштування, моніторинг, адміністрування та діагностику системи.

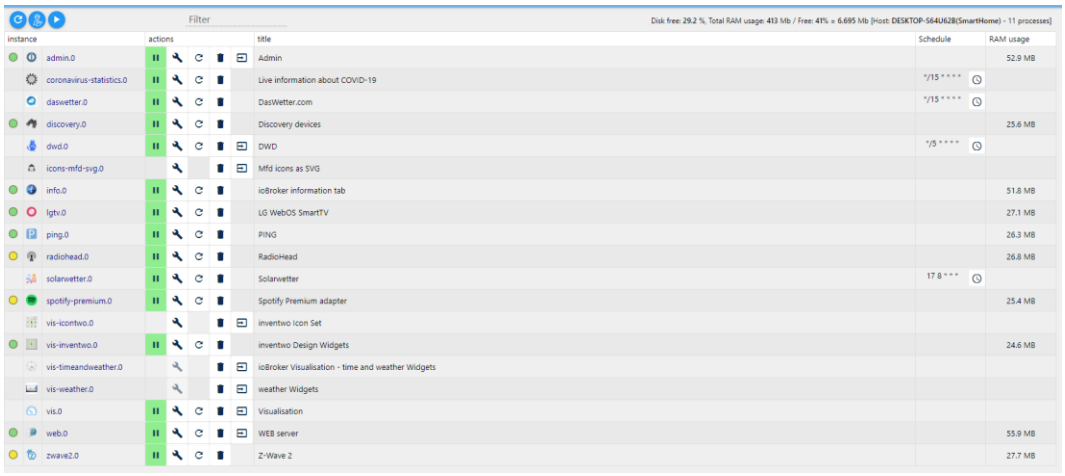


Рисунок 3.8 – Перелік адаптерів, використаних в ioBroker

Тепер розглянемо використання адаптерів для створення візуалізацій пристроїв і процесів в ioBroker. Для цього ми звернемо увагу на адаптер VIS Material Design.

Завдяки віджетам Material Design Widgets адаптера VIS, користувачі можуть створювати візуалізації, які дотримуються принципів матеріального дизайну. У наступних кроках ми розглянемо, як встановити цей адаптер і крок за кроком побудувати першу візуалізацію з використанням цього дизайну.

Структура навігаційного меню у візуалізації повинна виглядати наступним чином:

- Огляд – загальний огляд стану системи та її елементів.
- Освітлення – налаштування і контроль освітлення в системі.
- Статус – перевірка та моніторинг поточного стану різних пристроїв.
- Система – налаштування і моніторинг самої системи ioBroker.
- ioBroker – управління основними компонентами платформи ioBroker.
- Гомематичний – спеціальні налаштування для роботи з гомематичними пристроями.
- Відтінок – налаштування кольорів або відтінків візуалізації.

Першим кроком є створення нових уявлень для проекту VIS. Після цього відкривається індекс перегляду, в якому міститься віджет навігації Top Bar Navigation. Тут ми додаємо три елементи в меню навігації та розміщуємо підменю в третьому записі меню. Для підменю ми налаштовуємо кількість пунктів на три (по суті, це три елементи підменю).

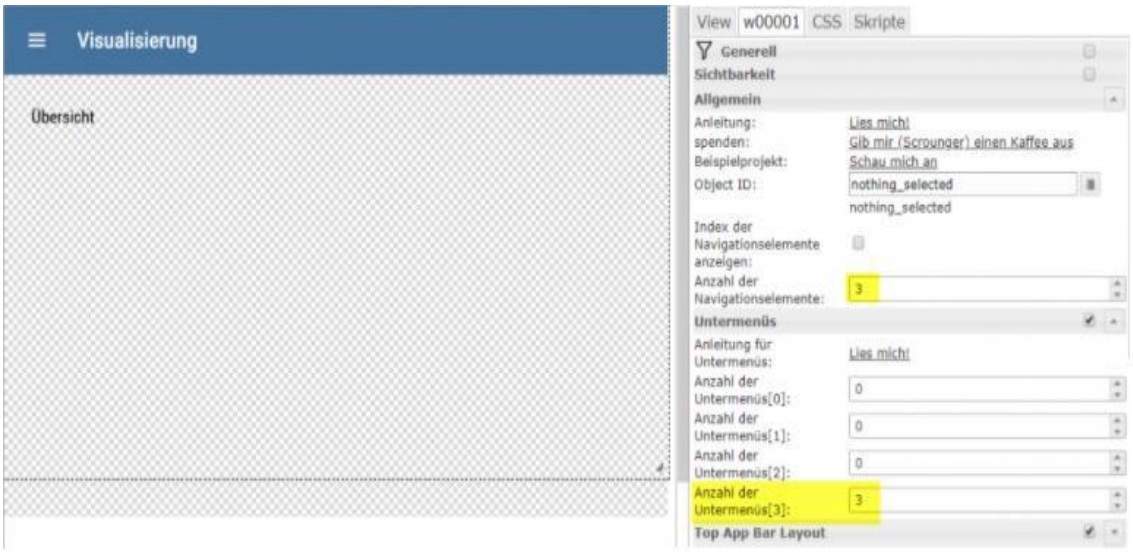


Рисунок 3.9 – Перегляд Material Design Widgets

Налаштування підменю відбувається через об'єкт JSON, в якому основний елемент позначено властивістю `itemText`. У списку підпунктів перераховуються окремі елементи підменю, кожен з яких визначається в квадратних дужках і розділяється комами (наприклад, `["A", "B", "C"]`).

Після завершення налаштування віджета Top Bar Navigation, ми переходимо до налаштування віджета View in View в індексному перегляді. Властивості цього віджета дозволяють змінити кількість переглядів до шести, що в загальному дає сім переглядів (від 0 до 6). У нових полях для подань зберігаються раніше створені перегляди. Для основного входу в систему використовувався ioBroker View, що забезпечує зручність для користувачів.

3.3 Опис та розширений розбір роботи

Було розроблено додаток VIS, який включає в себе управління пристроями та синхронізацію з системою розумного будинку Z-Wave. У даному розділі ми розглянемо детально візуалізацію управління термостатами, що є однією з ключових функцій цієї системи.

Особливості візуалізації:

1. Універсальність – одна сторінка може бути використана для управління будь-якою кількістю кімнат та профілів, що значно спрощує користування.

2. Логічний поділ на картки – для забезпечення адаптивного дизайну використовуються різні картки, що організовують візуалізацію та дозволяють легше працювати з інформацією.

3. Перекриваючі віджети – додано більше віджетів, що можуть накладатися один на одного, що дозволяє компактніше організувати елементи управління.

4. Налаштування кольорів – через діалогове вікно можна змінювати кольори елементів інтерфейсу (шрифт, фон тощо), що дає можливість персоналізувати інтерфейс.

5. Адаптивність до змін конфігурації адаптера – система автоматично адаптується при зміні конфігурації адаптера, наприклад, при зміні кількості профілів, режиму зниження температури, типу профілю та періодів.

6. Огляд стану вікон – додана картка для перегляду стану вікон у кожній кімнаті, що є корисним для моніторингу енергозбереження.

7. Чуйний дизайн – адаптивний дизайн забезпечує зручне використання на різних пристроях, зокрема на мобільних.

8. Складані елементи – для економії місця використано складані елементи, які згортаються при необхідності.

9. Легкість установки – навіть новачки можуть легко встановити цей додаток, оскільки це цілісний проект, який можна використовувати як основу для власних проектів.

10. Приклад для розширення функцій – додано зразок сторінки статусу кімнати, який можна використовувати як основу для подальших розширень основних функцій.

Опис інтерфейсу:

- Шаблон простору для карт: шаблон для макету карток, який дозволяє відображати основні значення для кожної кімнати. Після налаштування віджетів користувач може експортувати їх, створюючи нові сторінки для кожної кімнати, а потім налаштовувати ідентифікатори об’єктів віджетів.

- Інтеграція карток на сторінку: створені картки інтегруються на центральну сторінку через віджет "Перегляд у віджеті", що дозволяє визначити порядок відображення карток. Для цього необхідно редагувати властивості "Viewname" і змінювати клас CSS для визначення порядку карток.

Приклад реалізації:

1. Створення вигляду для кімнати, наприклад, вітальня, шляхом експорту віджетів і вставки їх у новий вигляд.

2. Налаштування карток з різними профілями, наприклад, для профілю "Усі дні разом", "Пн-Пт / Сб-Нд", або "кожен день окремо".

3. Після інтеграції карток на сторінку, коригуються віджети для відображення в правильному порядку.

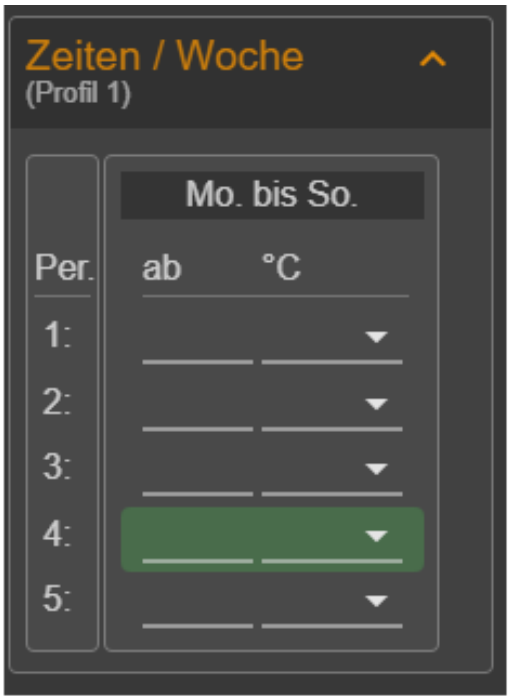


Рисунок 3.10 – Для профілю "Усі дні разом"

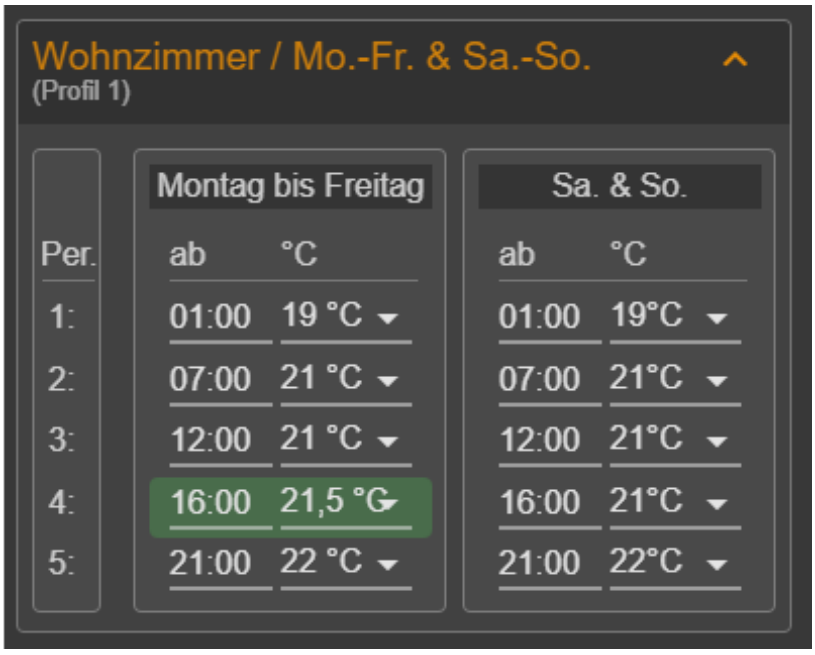


Рисунок 3.11 – З типом профілю "Пн-Пт / Сб-Нд"

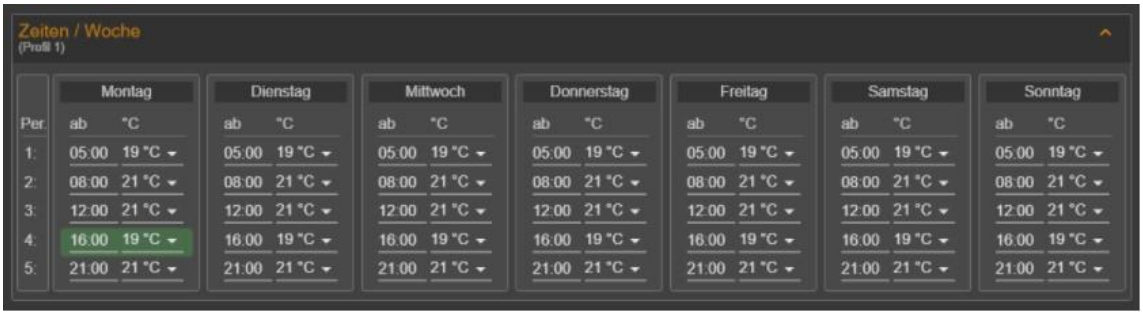


Рисунок 3.12 – З типом профілю "кожен день відокремлено"

В результаті виконаних налаштувань отримано інтерфейс, який дозволяє ефективно управляти термостатами у розумному будинку через веб-інтерфейс, з можливістю моніторингу та налаштування температури в різних кімнатах. Рисунок 3.13 показує приклад вигляду цього додатку, в якому чітко представлені всі елементи управління.

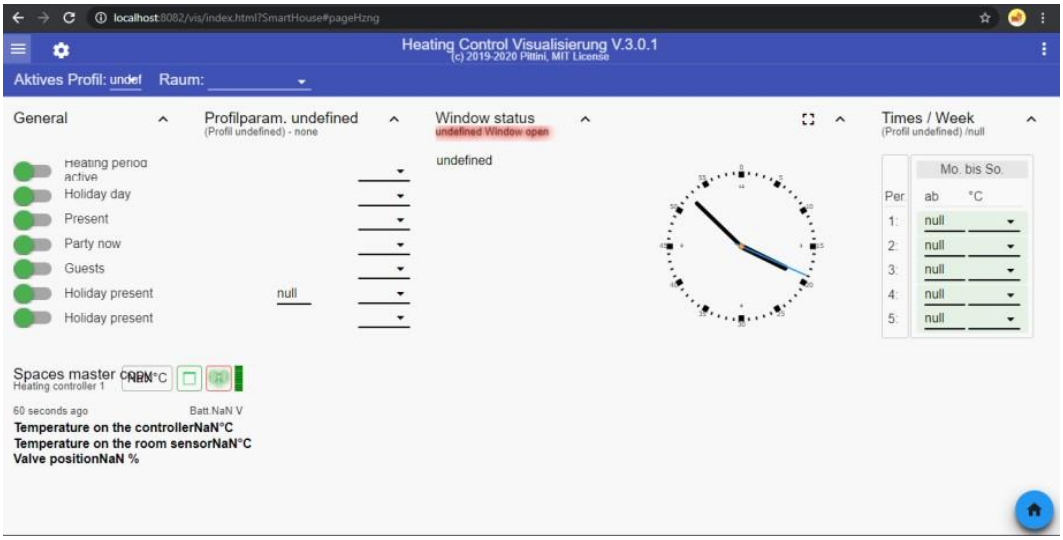


Рисунок 3.13 – Вигляд додатку управління термостатами

Використані адаптери:

1. Material Design Widgets – цей адаптер використовується для створення візуальних віджетів, що відповідають принципам матеріального дизайну.

2. LG WebOs SmartTV – адаптер для інтеграції з телевізорами LG на платформі WebOS [24].

3. Bars Widgets, Canvas-Gauges style Widgets, MFD icons as PNG/SVG – використовуються для створення різноманітних віджетів та інтерфейсних елементів, таких як індикатори та іконки [25].

4. Time and Weather Widgets, Admin, BackItUp, Discovery devices – додаткові адаптери для управління часом, погодними умовами та для резервного копіювання.

Усі ці адаптери підтримують необхідні функції для управління пристроями розумного будинку та надають гнучкість в налаштуваннях, що робить систему потужною та зручною для користувачівх [26].

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Автоматизована система керування розумним будинком» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;

2) розраховано витрати на здійснення науково-технічної розробки;

3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Автоматизована система керування розумним будинком» є оцінювання

науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [27].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					

8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	3	4
2. Ринкові переваги (наявність аналогів)	2	3	2
3. Ринкові переваги (ціна продукту)	3	4	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	3	4	4
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	35	39	37
Середньоарифметична сума балів $СБ_c$	37,0		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [27].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Автоматизована система керування розумним будинком» становить 37,0 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [27]:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i \,, \tag{4.1}$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу *i-го* технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має виконуватись умова $\sum_{i=1}^{k} \alpha_i = 1$;

β_i – відносне значення *i-го* технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{нi}}{I_{ai}} \,, \tag{4.2}$$

де $I_{нi}$ та $I_{на}$ – чисельні значення конкретного *i*-го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ні}}; \tag{4.3}$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Кількість базових підсистем контролю	од.	4	8	2	0,25
Кількість допоміжних підсистем контролю	од.	1	3	3	0,15
Рівень завадозахищеності системи	%	87	96	1,1	0,25
Споживана потужність, не більше	Вт	0,2	0,15	1,33	0,15
Рівень надійності системи	%	90	92	1,02	0,2

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення складе:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 2 \cdot 0{,}25 + 3 \cdot 0{,}15 + 1{,}1 \cdot 0{,}25 + 1{,}33 \cdot 0{,}15 + 1{,}02 \cdot 0{,}2 = 1{,}63.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,63 рази [28].

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Автоматизована система керування розумним будинком», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [27]:

$$З_о = \sum_{i=1}^{k} \frac{М_{ni} \cdot t_i}{Т_р}, \quad (4.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

$М_{ni}$ – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

$Т_р$ – середнє число робочих днів в місяці, $Т_р$=21 дні.

$$З_о = 16400{,}00 \cdot 21 / 21 = 16400{,}00 \; грн.$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий	Оплата за робочий	Число днів роботи	Витрати на заробітну

	оклад, грн	день, грн		плату, грн
Проектний менеджер	16400,00	780,95	21	16400,00
Інженер-дослідник	15550,00	740,48	10	7404,76
Інженер-розробник автоматизованих систем управління	15200,00	723,81	11	7961,90

Продовження таблиці 4.5.

Лаборант	8000,00	380,95	21	8000,00
Всього				39766,67

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Автоматизована система керування розумним будинком» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i \,, \tag{4.5}$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}} \,, \tag{4.6}$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $Мм$=8000,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2) [27];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 8000{,}00 \cdot 1{,}35 \cdot 1{,}15 / (21 \cdot 8) = 73{,}93$ *грн.*
$З_{р1} = 73{,}93 \cdot 6{,}50 = 480{,}54$ *грн.*

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Підготовка робочого місця розробника автоматизованих систем управління	6,50	3	1,35	73,93	480,54
Підготовка програмного забезпечення імітаційного моделювання	6,00	4	1,50	82,14	492,86
Формування бази даних моделей впливу	9,10	3	1,35	73,93	672,75
Проведення експерименту	6,25	4	1,50	82,14	513,39
Фіксація результатів експерименту	2,00	2	1,10	60,24	120,48
Всього					2280,01

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \qquad (4.7)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 10%.

$З_{дод} = (39766{,}67 + 2280{,}01) \cdot 10 / 100\% = 4204{,}67$ *грн.*

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (4.8)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (39766,67 + 2280,01 + 4204,67) · 22 / 100% = 10175,30 грн.

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Автоматизована система керування розумним будинком».

Витрати на матеріали (*М*), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj}\ , \quad (4.9)$$

де $Н_j$ – норма витрат матеріалу *j*-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу *j*-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 ... 1,15*);

$В_j$ – маса відходів *j*-го найменування, кг;

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

$М_1$ = 3,0 · 188,00 · 1,01 - 0 · 0 = 569,64 грн.

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір офісний (А4, 500, 80г/м)	188,00	3,0	0	0	569,64
Папір для заміток	75,00	4,0	0	0	303,00
Органайзер офісний	132,00	2,0	0	0	266,64
Начиння канцелярське	185,00	3,0	0	0	560,55
Картридж для принтера	980,00	2,00	0	0	1979,60
Блок памяті (зовнішній) 256 ГБ	2400,00	1,0	0	0	2424,00
FLASH-пам'ять	169,00	1,0	0	0	170,69
Всього					6274,12

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Автоматизована система керування розумним будинком», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j \qquad (4.10)$$

де $Н_j$ – кількість комплектуючих *j*-го виду, шт.;

$Ц_j$ – покупна ціна комплектуючих *j*-го виду, грн;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ *= 1,1 ... 1,15*).

$К_в = 1 \cdot 1200{,}00 \cdot 1{,}01 = 1212{,}00$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.8 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Обладнання керуючої частини автоматизованої системи (імітаційна модель)	1	1200,00	1212,00
Підсистема керування	1	650,00	656,50

освітленням (імітаційна модель)			
Підсистема керування клімат-контролем будівлі (імітаційна модель)	1	980,00	989,80
Охоронно-протипожежна підсистема (імітаційна модель)	1	820,00	828,20
Інші допоміжні підсистеми (імітаційна модель)	1	1050,00	1060,50
Всього			4747,00

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (4.11)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.i}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань устаткування.

$B_{спец} = 26116,00 \cdot 1 \cdot 1,01 = 26377,16$ *грн.*

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Планшет для керування розумним будинком Livolo (VL-XT001)	1	26116,00	26377,16
Всього			26377,16

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.12)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань програмних засобів.

$B_{прг} = 9420,00 \cdot 1 \cdot 1,01 = 9514,20$ грн.

Отримані результати зведемо до таблиці:

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Середовище моделювання Proteus (ISIS Proteus)	1	9420,00	9514,20
MathLab 12 PRO	1	6800,00	6868,00
Всього			16382,20

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.13)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл}$ = (44850,00 · 1) / (3 · 12) = 1245,83 грн.

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Програмно-аналітичний комплекс на основі ПК	44850,00	3	1	1245,83
Графічно-обчислювальний комплекс обробки даних	33999,00	3	1	944,42
Програмні пакети Microsoft Windows, Office 2016	9340,00	3	1	259,44
Імітаційне обладнання (аналогово-цифрове)	8700,00	4	1	181,25
Місце розробника спеціалізоване	8255,00	5	1	137,58
Офісна	9650,00	5	1	160,83

оргтехніка				
Приміщення дослідницької лабораторії	380000,00	30	1	1055,56
Принтер	6520,00	5	1	108,67
Всього				4093,58

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (4.14)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо Цe = 10,98 грн;

$K_{вni}$ – коефіцієнт, що враховує використання потужності, $K_{вni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$B_e = 0,35 \cdot 160,0 \cdot 10,98 \cdot 0,95 / 0,97 = 614,88$ грн.

Проведені розрахунки зведемо до таблиці.

Таблиця 4.12 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Програмно-аналітичний комплекс на основі ПК	0,35	160,0	614,88
Графічно-обчислювальний комплекс обробки даних	0,25	160,0	439,20
Імітаційне обладнання (аналогово-цифрове)	0,12	50,0	65,88
Місце розробника спеціалізоване	0,10	160,0	175,68
Офісна оргтехніка	0,40	1,2	5,36
Принтер	0,26	3,3	9,28

Планшет для керування розумним будинком Livolo (VL-XT001)	0,02	100,0	21,96
Всього			1332,24

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Автоматизована система керування розумним будинком» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» відсутні.

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» відсутні.

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.15)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$I_в$ = (39766,67 + 2280,01) · 50 / 100% = 21023,34 грн.

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.16)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 100%.

$B_{нзв}$ = (39766,67 + 2280,01) · 100 / 100% = 42046,68 грн.

Витрати на проведення науково-дослідної роботи на тему «Автоматизована система керування розумним будинком» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + B_{спец} + B_{прг} + A_{обл} + B_е + B_{св} + B_{сп} + I_в + B_{нзв}. \quad (4.18)$$

$B_{заг}$= 39766,67 +2280,01 +4204,67 +10175,30 +6274,12 +4747,00 +26377,16 + 16382,20 + 4093,58 +1332,24 +0,00 +0,00 +21023,34 +42046,68 = 178702,96 грн.

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.17)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,9.

ЗВ = 178702,96 / 0,9 = 198558,84 грн.

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Автоматизована система керування розумним будинком» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

1) Розробка чи суттєве вдосконалення машини (механізму, приладу, пристрою) для використання кінцевими споживачами.

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

ΔN – збільшення кількості споживачів пристрою, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	50	90	95	80

N – кількість споживачів які використовували аналогічний пристрій у році до впровадження результатів нової науково-технічної розробки, приймемо 800 осіб;

$Ц_б$ – вартість пристрою у році до впровадження результатів розробки, приймемо 65000,00 грн;

$\pm\Delta Ц_o$ – зміна вартості пристрою від впровадження результатів науково-технічної розробки, приймемо 5852,50 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [27]:

$$\Delta П_i = (\pm\Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.18)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2025 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо $\rho = 38\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2025 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1 = (5852,50 \cdot 800,00 + 70852,50 \cdot 50) \cdot 0,83 \cdot 0,38 \cdot (1\text{-}0,18/100\%) = 2127118,31$ грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2 = (5852,50 \cdot 800,00 + 70852,50 \cdot 140) \cdot 0,83 \cdot 0,38 \cdot (1\text{-}0,18/100\%) = 3776317,95$ грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3 = (5852,50 \cdot 800,00 + 70852,50 \cdot 235) \cdot 0,83 \cdot 0,38 \cdot (1\text{-}0,18/100\%) = 5517139,78$ грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4 = (5852,50 \cdot 800,00 + 70852,50 \cdot 315) \cdot 0,83 \cdot 0,38 \cdot (1\text{-}0,18/100\%) = 6983095,01$ грн.

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

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.19)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=0,25;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = 2127118,31/(1+0,25)^1+3776317,95/(1+0,25)^2+5517139,78/(1+0,25)^3+$$
$$+6983095,01/(1+0,25)^4=1701694,65+2416843,49+2824775,57+2860275,72=$$
$$=9803589,42 \text{ грн.}$$

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \quad (4.20)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 198558,84 грн.

$PV = k_{інв} \cdot ЗВ$ = 2 · 198558,84 = 397117,69 грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \tag{4.21}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 9803589,42 грн;

PV – теперішня вартість початкових інвестицій, 397117,69 грн.

$E_{абс} = ПП - PV$ = 9803589,42 - 397117,69 = 9406471,73 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{4.22}$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 9406471,73 грн;

PV – теперішня вартість початкових інвестицій, 397117,69 грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1$ = (1+9406471,73/397117,69)$^{1/4}$= 1,23.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f\,, \tag{4.23}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2025 році в Україні d =0,1;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,3.

$\tau_{мін}$ = 0,1+0,3 = 0,4 < 1,23 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Автоматизована система керування розумним будинком» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}\,, \tag{4.24}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 1,23 = 0,81 р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Автоматизована система керування розумним будинком» становить 37,0 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,63 рази.

Також термін окупності становить 0,81 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Автоматизована система керування розумним будинком».

ВИСНОВКИ

В процесі виконання даної роботи було вирішено важливе завдання, що стосувалося покращення зручності використання додатків для управління системами розумного будинку. Потрібно було знайти спосіб інтеграції існуючих пристроїв, що працюють на протоколі Z-Wave, з іншими пристроями, які не підтримують стандартні протоколи розумних будинків. Завдяки цьому було забезпечено зручне доповнення вже наявної системи розумного будинку новими пристроями, які раніше не могли взаємодіяти з основною системою через обмеження протоколу.

У розглянутій роботі особлива увага була приділена протоколу Z-Wave, який використовувався в уже існуючій системі управління розумним будинком. Цей протокол був протестований на надійність в умовах зовнішніх загроз, таких як DDoS-атаки та спроби злому, і підтвердив свою стабільність та безпеку. За результатами тестування стало зрозуміло, що Z-Wave є надійним і ефективним рішенням для інтеграції різноманітних пристроїв у систему розумного будинку. Оскільки протокол забезпечує безпечне підключення та управління пристроями, було вирішено знайти платформу автоматизації, яка підтримує цей протокол.

Після порівняння кількох систем автоматизації, таких як zVirtualScenes, Ago Control, ioBroker та Domoticz, було визначено, що ioBroker є найкращим вибором для цієї роботи. Domoticz було відкинуто через обмеження щодо безкоштовного використання для некомерційних цілей, що ускладнювало його застосування в даному контексті. Зрештою, платформу ioBroker було вибрано через її гнучкість, зручність у налаштуванні, а також наявність великої спільноти розробників, які активно обмінюються досвідом і інформацією через форуми. Це дозволило забезпечити високу якість роботи з платформою та прискорити процес розробки необхідних функцій.

Для реалізації системи автоматизації та візуалізації було використано адаптери OpenZWave та zwave-js для інтеграції з пристроями на протоколі Z-Wave, а також адаптер "VIS візуалізація" для створення інтерфейсу користувача. Платформа ioBroker дозволила створити чотири основні сторінки додатку, які були завантажені на сервер Google і виконуються в реальному часі. Ці сторінки включають головне меню, статус по коронавірусу, а також відображення термостатів, що дозволяє користувачам легко відслідковувати та керувати параметрами температури в системі розумного будинку.

Налаштовані сторінки мають адаптивний дизайн, що дозволяє користуватися інтерфейсом з різних пристроїв, таких як смартфони, планшети та комп'ютери. Це дозволяє досягти максимального комфорту та зручності для користувачів, незалежно від того, яким пристроєм вони користуються. У результаті було створено інтуїтивно зрозумілий і функціональний інтерфейс, який відповідає вимогам сучасних систем управління розумним будинком. Всі налаштування та вигляд створеного інтерфейсу доступні в додатках, що дає змогу легко вносити зміни і коригування за потреби.

Загалом, проведена робота підтвердила, що використання платформи ioBroker разом із протоколом Z-Wave дозволяє створити зручну та ефективну систему управління розумним будинком, яка може інтегрувати пристрої різних типів і марок. Цей підхід забезпечує не тільки високу функціональність, а й зручність для користувача, що є важливим аспектом для успішного використання таких систем в повсякденному житті.

ПЕРЕЛІК ПОСИЛАНЬ

1. Крижанівський І., Гальчинський М., "Інтелектуальні системи управління: теорія та практика", Львів: Видавництво Львівської політехніки, 2022, 340 с.

2. Соловей О., "Інтернет речей і кіберфізичні системи", Київ: Наукова думка, 2020, 288 с.

3. Мартиненко А., "Автоматизація будівель: сучасні рішення", Одеса: ТЕС, 2021, 245 с.

4. Данилюк П., "Технології керування системами розумного будинку", Харків: ХНУРЕ, 2023, 180 с.

5. Карпенко О., "Інтелектуальні мережі для розумних будинків", Київ: Інтерпрес, 2022, 310 с.

6. Герасименко І., "Застосування штучного інтелекту в системах розумного будинку", Дніпро: ДНУ, 2021, 250 с.

7. Петров О., "Розумні будинки: теорія і практика", Львів: Видавництво Львівської політехніки, 2020, 298 с.

8. Смірнов В., "Автоматизація житлових систем: сучасні підходи", Київ: КНТЕУ, 2023, 275 с.

9. Ковальчук О., "Системи моніторингу і управління в розумних будинках", Харків: ХНУРЕ, 2022, 260 с.

10. Бондаренко В., "Архітектура автоматизованих систем управління", Київ: Вид-во КНЕУ, 2021, 350 с.

11. Василенко В., "Інтелектуальні автоматизовані системи", Харків: НТУ "ХПІ", 2023, 320 с.

12. Павлюк А., "Технології автоматизованого управління будівлями", Вінниця: ВНТУ, 2020, 290 с.

13. Чумак М., "Автоматизація будівельних систем на базі IoT", Львів: ЛНУ, 2023, 270 с.

14. Соколов О., "Керування електроспоживанням в розумних будинках", Київ: Техніка, 2021, 210 с.

15. Білан В., "Технології автоматизації розумних будинків", Одеса: ОНПУ, 2022, 225 с.

16. Малюк О., "Автоматизовані системи контролю і управління", Дніпро: ДНУ, 2023, 180 с.

17. A. Smith, "Smart Home Automation: A Complete Guide", 2nd ed., New York: TechWorld, 2021, 420 pp. –

URL:https://www.researchgate.net/publication/369434604_Smart_Home_System_A_Comprehensive_Review

18. J. Brown, "IoT in Smart Homes: Current and Future Perspectives", London: Academic Press, 2023, 310 pp.-

URL:https://www.researchgate.net/publication/26788467_Smart_homes_-_Current_features_and_future_perspectives

26. Vue.js. [Електронний ресурс]. // Released under the MIT License. Copyright © 2014-2024 Evan You – URL: https://vuejs.org/

27. Angular. [Електронний ресурс] // official documentation. Super-powered by Google ©2010-2024. URL: https://angular.io/

28. Web frameworks. [Електронний ресурс]. – geeksforgeeks.org, 15 Oct, 2024 . URL: https://www.geeksforgeeks.org/top-10-frameworks-for-web-applications/

29. Curie D.H., et al. Analysis on web frameworks. Journal of Physics: Conference Series. IOP Publishing, 2019. p. 012114.

30. .NET Framework documentation. [Електронний ресурс] // Bill Wagner and other 12 contributions, .03.30.2023 – URL: https://learn.microsoft.com/en-us/dotnet/framework/

31. What is PostgreSQL. [Електронний ресурс] // Neon, January 18, 2024 – URL: https://www.postgresql.org/docs/current/

32. Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch, and others. [Електронний ресурс] // AltexSoft, 16 May, 2023 – URL: https://www.altexsoft.com/blog/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/

33. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 174 «Автоматизація, комп’ютерно-інтегровані технології та робототехніка» (кафедра комп'ютерних систем управління) [Електронний ресурс] / уклад. В. М. Дубовой. – Вінниця :ВНТУ, 2023 – (PDF, 45 с.). URL: https://iq.vntu.edu.ua/repository/getfile.php/6077.pdf

27. Козловський В., Лесько О., Кавецький В.. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. :– Вінниця : ВНТУ, 2021. – 42 с.

28. Кавецький В., Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

ДОДАТКИ

Додаток А

(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: «Автоматизована система керування розумним будинком »
Тип роботи: магістерська кваліфікаційна робота
(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше (зазначити))
Підрозділ кафедра комп'ютерних систем управління
(кафедра, факультет (інститут), навчальна група)
Керівник Ковтун В. В., професор, зав. кафедри КСУ
(прізвище, ініціали, посада)

Показники звіту подібності

Turnitin	
Оригінальність	89%
Загальна схожість	11%

Аналіз звіту подібності (відмітити потрібне)

√ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Віталій БАЖАН _____
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

_____

Особа, відповідальна за перевірку _____ Володимир ДУБОВОЙ
(підпис) (прізвище, ініціали)

Експерт _____ _____
(за потреби) (підпис) (прізвище, ініціали, посада)

Додаток Б

(обов'язковий)

ВНТУ

ЗАТВЕРДЖУЮ
Зав. кафедри КСУ

В'ячеслав КОВТУН

"14" жовтня 2024 року

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістреської кваліфікаційної роботи

Автоматизована система керування розумним будинком
08.33.МКР.01.00.000 ТЗ

Студент групи 2АКІТР-23м
Віталій БАЖАН
Підпис *(Ім'я та ПРІЗВИЩЕ)*

Керівник: Д.т.н., проф. КСУ
В'ячеслав КОВТУН
Підпис *(Ім'я та ПРІЗВИЩЕ)*

Вінниця 2025

1. Назва та галузь застосування

1.1.Назва – Автоматизована система керування розумним будинком.

1.2.Галузь застосування – комп’ютерно-інтегровані системи управління.

2.Підстава для проведення розробки.

2.1.Тема магістреської кваліфікаційнохї роботи затверджена наказом по ВНТУ №310 від 17.09.2024 р.

3.Мета та призначення розробки.

3.1. Метою магістерської кваліфікаційної роботи розробка та впровадження автоматизованої системи керування розумним будинком, яка забезпечує інтеграцію різних пристроїв та підсистем для підвищення енергоефективності, безпеки та комфорту мешканців.

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

- .Данилюк, "Технології керування системами розумного будинку", Харків: ХНУРЕ, 2023, 180 с.

О. Карпенко, "Інтелектуальні мережі для розумних будинків", Київ: Інтерпрес, 2022, 310 с.

- І. Герасименко, "Застосування штучного інтелекту в системах розумного будинку", Дніпро: ДНУ, 2021, 250 с.
- О. Петров, "Розумні будинки: теорія і практика", Львів: Видавництво Львівської політехніки, 2020, 298 с.
- В. Смірнов, "Автоматизація житлових систем: сучасні підходи", Київ: КНТЕУ, 2023, 275 с.5.Вимоги до розробки.

5.1. Перелік головних функцій:

- збір інформації.
- керування процесами.
- керування базою даних.
- формування звітів.

5.2. Основні технічні вимоги до розробки

5.2.1. Вимоги до програмної платформи:

- Window 10;
- Arduino IDE;

5.2.2. Умови експлуатації системи:

- робота на мобільних додатках;
- можливість цілодобового функціонування системи;
- дані оновлюються і є актуальними.

6. Стадії та етапи розробки.

6.1. Поянювальна записка:

1. Аналіз методів, принципів, підходів і засобів реалізації задачі ав- томатизації процесами в об’єкті управління відповідно до теми магістреської кваліфікаційнохї роботи. Постановка задач дослідження 09.12.2024р.
2. Удосконалення технології прийняття рішень при автоматизації об’єкту 23.12.2024р.
3. Визначення техніних характеристик системи 01.01.2025р
4. Розробка апаратної частини системи 21.01.2025р
5. Розробка програмного забезпечення системи 26.01.2025р.

6.2. Графічні матеріали:

1. Розробка UML-діаграм системи 23.12.2024 р.
2. Розробка моделі бази даних системи 01.01.2025р.
3. Тестування програмного забезпечення 01.01.2025р.

7. Порядок контролю й приймання.

7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «26» 01.2026 р.

7.2. Атестація проекту здійснюється на попередньому захисті. Попередній захист магістреської кваліфікаційної роботи провести до «1» лютого 2025р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістреської кваліфікаційної роботи провести «18» 02.2025р.

Додаток В

(вибірковий)

Лістинг коду

```
import UIKit
struct City {
var x: Int
var y: Int
var isVisisted: Bool
}
func twoOptSwap(route: [Int], i: Int, j: Int) -> [Int] {
var newRoute = route
// Reverse the subpath between indices i and j
var k    =        i
var l    =        j
while   k        < l {
newRoute.swapAt(k, l) k += 1
l -= 1
}
return newRoute
}
func twoOpt(route: [Int], distances: [[Double]]) -> [Int] {
var bestRoute = route
var improvement = true
while improvement { improvement = false var bestDistance =
calculateTotalDistance(route: bestRoute, distances: distances)
for i in 1..<route.count - 2 {
for j in (i + 1)..<route.count {
let newRoute = twoOptSwap(route: bestRoute, i: i, j: j)

let newDistance = calculateTotalDistance(route: newRoute, distances: distances)
```

```
if newDistance < bestDistance { bestRoute = newRoute improvement = true bestDistance = newDistance
}
}
}
}
return bestRoute
}
func calculateTotalDistance(route: [Int], distances: [[Double]]) -> Double {
var totalDistance: Double = 0.0
for i in 0..<route.count - 1 {
totalDistance += distances[route[i]][route[i
+ 1]]
}
totalDistance += distances[route.last!][route.first!]
return totalDistance
}
let cities = 5
let distances: [[Double]] = [
[0,      2,      9,      10, 1],
[2,      0,      6,      3, 2],
[9,      6,      0,      7, 6],
[10, 3, 7, 0, 5],
[1, 2, 6, 5, 0]
]
var initialRoute = Array(0..<cities) print("Initial Route: \(initialRoute)")
let optimizedRoute = twoOpt(route: initialRoute,
distances: distances)
print("Optimized Route: \(optimizedRoute)")

let totalDistance = calculateTotalDistance(route: optimizedRoute, distances: distances) print("Total Distance: \(totalDistance)")
```

```
func nearestNeighborTSP(distances: [[Double]]) -> [Int] {
let numberOfCities = distances.count
var visited = Array(repeating: false, count: numberOfCities)
var tour = [0]
visited[0] = true
for _ in 1..<numberOfCities {
var nearestCity = -1
var minDistance = Double.infinity
for city in 0..<numberOfCities {
if !visited[city] {
if distances[tour.last!][city] <

minDistance {

minDistance =

distances[tour.last!][city]
nearestCity = city
}
}
}
if nearestCity != -1 { tour.append(nearestCity) visited[nearestCity] = true
}
}
tour.append(tour[0])
return tour
}

func angleBetweenLines(line1: (m: Double, c: Double), line2: (m: Double, c: Double)) -> Double {
let       m1      = line1.m
let       m2      = line2.m
let theta = atan(abs((m2 - m1) / (1 + m1 * m2)))
```

```
let degrees = theta * (180.0 / Double.pi)
return degrees
}
struct Point { let x: Double let y: Double
}
func isConvexPolygon(points: [Point]) -> Bool {
let count = points.count
if count < 3 {
return false
}
var sign: Double = 0
for i in 0..<count {
let p1 = points[i]
let p2 = points[(i + 1) % count]
let p3 = points[(i + 2) % count]
let crossProduct = (p2.x - p1.x) * (p3.y - p2.y) - (p2.y - p1.y) * (p3.x - p2.x)
if crossProduct == 0 {
continue
}
if sign == 0 {
sign = crossProduct > 0 ? 1 : -1
} else if crossProduct > 0 {
return false
}
}

return true
}
import SwiftCV
let image = imread("city1.png")
var grayImage = Mat()
cvtColor(image, &grayImage, code: .COLOR_BGR2GRAY)
```

```
var binaryImage = Mat()
threshold(grayImage, &binaryImage, 128, 255, type:
.THRESH_BINARY)
var contours = [CvPointSeq]()
var hierarchy = Mat()
findContours(binaryImage, &contours, &hierarchy, mode:
.RETR_EXTERNAL, method: .CHAIN_APPROX_SIMPLE)
var cityCoordinates = [Point]()
for contour in contours {
let perimeter = arcLength(contour, closed: true) var approx = Mat()
approxPolyDP(contour, &approx, 0.04 * perimeter, closed: true)
if let points = approx.toArray(type: Point.self)
{
cityCoordinates.append(contentsOf: points)
}
}
for (index, point) in cityCoordinates.enumerated() { print("Місто \(index + 1): X = \(point.x), Y =
\(point.y)")
}

func loadImageFromDesktop() {
let openPanel = NSOpenPanel() openPanel.allowedFileTypes = ["jpg", "jpeg",
"png", "gif", "bmp"] openPanel.canChooseFiles = true openPanel.canChooseDirectories = false
openPanel.allowsMultipleSelection = false
openPanel.begin { (result) in if result == .OK {
if let url = openPanel.url {
do {

url) data) {

let data = try Data(contentsOf:
if let image = NSImage(data:
```

```
}
} catch {
print("Помилка при читанні

зображення: \(error)")
}
}
}
}
}
func printTSPRoute(route: [Int], cities: [String]) {
guard route.count == cities.count else { print("Помилка: кількість міст у шляху не
співпадає з заданою кількістю міст.")
return
}
var totalDistance = 0.0
for i in 0..<route.count - 1 {
let cityIndex = route[i]
let nextCityIndex = route[i + 1]
let cityName = cities[cityIndex]

let nextCityName = cities[nextCityIndex]
let distance = distances[cityIndex][nextCityIndex]
totalDistance += distance
print("Крок \(i + 1): Відвідати \(cityName),
Відстань до \(nextCityName): \(distance)")
}
let startCity = cities[route[0]] let endCity = cities[route.last!] let lastDistance =
distances[route.last!][route[0]] totalDistance += lastDistance print("Шаг \(route.count): Повернутись
у
початкове місто \(startCity), відстань до \(endCity):
\(lastDistance)")
```

```
print("Фінальна довжина шляху: \(totalDistance)")
}
func measureExecutionTime(block: () -> Void) {
let startTime = Date() block()
let endTime = Date()
let executionTime = endTime.timeIntervalSince(startTime)
print("Час виконання: \(executionTime) секунд")
}
// Структура для представлення точки в 2D-просторі
struct Point { var x: Double var y: Double
}

// Структура для представлення прямої в загальному вигляді Ax + By + C = 0
struct Line {
var      A:      Double
var
var      B:
C:       Double
Double
}
// Функція для перевірки, чи точка лежить на прямій
func pointOnLine(point: Point, line: Line) -> Bool {
return line.A * point.x + line.B * point.y + line.C == 0
}
// Структура для представлення прямокутника
struct Rectangle { var left: Double var right: Double var top: Double
var bottom: Double
}
// Функція для перевірки, чи точка знаходиться всередині, на межі або поза межами прямокутника func pointInRectangle(point: Point, rectangle: Rectangle) -> String {
if point.x >= rectangle.left && point.x <= rectangle.right &&
point.y >= rectangle.top && point.y <= rectangle.bottom {
```

```
return "Точка знаходиться всередині
прямокутника"
} else if point.x == rectangle.left || point.x ==
rectangle.right ||
point.y == rectangle.top || point.y == rectangle.bottom {
return "Точка лежить на межі прямокутника"
} else {
return "Точка знаходиться поза межами прямокутника"
}
}
// Приклад використання функцій
let point = Point(x: 2.0, y: 3.0)
let line = Line(A: 1.0, B: -1.0, C: -1.0)

let rectangle = Rectangle(left: 1.0, right: 5.0, top: 2.0, bottom: 4.0)
let isPointOnLine = pointOnLine(point: point, line: line)
let pointStatusInRectangle = pointInRectangle(point:
point, rectangle: rectangle)
print("Перевірка прямої: \(isPointOnLine ? "Точка лежить на прямій" : "Точка не лежить на прямій")") print("Перевірка прямокутника:
\(pointStatusInRectangle)")
import Foundation
func measureExecutionTime(block: () -> Void) {
let startTime = Date()
struct City {
var x: Double
var y: Double
}
func calculateDistance(city1: City, city2: City)
-> Double {
        let dx   = city1.x        - city2.x
        let dy   = city1.y        - city2.y
```

```
        return  sqrt(dx *       dx + dy * dy)
}
func calculateTotalDistance(path: [Int], cities: [City]) -> Double {
var totalDistance = 0.0
for i in 0..<path.count - 1 {
let fromCity = cities[path[i]]
let toCity = cities[path[i + 1]] totalDistance += calculateDistance(city1:
fromCity, city2: toCity)
}
totalDistance += calculateDistance(city1: cities[path.last!], city2: cities[path.first!])
return totalDistance

}
func generateRandomCities(n: Int) -> [City] {
var cities = [City]()
for _ in 0..<n {
let x = Double.random(in: 0.0...100.0) let y = Double.random(in: 0.0...100.0) cities.append(City(x: x, y:
y))
}
return cities
}
func permute(_ array: inout [Int], _ n: Int, cities: [City], bestPath: inout [Int], bestDistance: inout
Double) {
if n == 0 {
let distance = calculateTotalDistance(path: array, cities: cities)
if distance < bestDistance {
bestDistance = distance bestPath = array
}
} else {
for i in 0..<n { array.swapAt(i, n - 1)
permute(&array, n - 1, cities: cities, bestPath: &bestPath, bestDistance:
&bestDistance)
```

```
array.swapAt(i, n - 1)
}
}
}
func tspBruteForce(n: Int) -> (path: [Int], distance: Double) {
let cities = generateRandomCities(n: n)
let numberOfCities = cities.count var path = Array(0..<numberOfCities) var bestPath = path
var bestDistance = calculateTotalDistance(path: path, cities: cities)

permute(&path, numberOfCities, cities: cities, bestPath: &bestPath, bestDistance:
&bestDistance)
return (path: bestPath, distance: bestDistance)
}
let numberOfCities = 10
let result = tspBruteForce(n: numberOfCities) print("Випадково згенеровані міста:")
for (index, city) in generateRandomCities(n:
numberOfCities).enumerated() {
print("Місто \(index + 1): (\(city.x),
\(city.y))")
}
print("Найкращий маршрут: \(result.path)") print("Мінімальна відстань: \(result.distance)")
let endTime = Date()
let executionTime = endTime.timeIntervalSince(startTime)
print("Час виконання: \(executionTime) секунд")
}
import UIKit
import CryptoKit
import CommonCrypto
class ViewController: UIViewController, UIImagePickerControllerDelegate,
UINavigationControllerDelegate {
@IBOutlet var loadImageButton: UIButton!
@IBOutlet var enterDataButton: UIButton!
```

```
var imagePicker = UIImagePickerController()
override func viewDidLoad() { super.viewDidLoad() loadImageButton.clipsToBounds = true loadImageButton.layer.cornerRadius = 16 enterDataButton.clipsToBounds = true

enterDataButton.layer.cornerRadius = 16 loadImageButton.addTarget(self, action:
#selector(loadImage), for: .touchUpInside) enterDataButton.addTarget(self, action:
#selector(enterData), for: .touchUpInside)
imagePicker.delegate = self imagePicker.sourceType = .savedPhotosAlbum imagePicker.allowsEditing = false
}
@objc func loadImage() {
if UIImagePickerController.isSourceTypeAvailable(.savedP hotosAlbum){
present(imagePicker, animated: true, completion: nil)
}
}
@objc func enterData() {
let vc = self.storyboard?.instantiateViewController(withIdenti fier: "CoordinatesVC") as? CoordinatesVC
self.present(vc ?? UIViewController(), animated: true)
}
func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any]) {
picker.dismiss(animated: true, completion:

nil)

guard let image = info[.originalImage] as?

UIImage else {
fatalError("Expected a dictionary containing an image, but was provided the following:
\(info)")
}
```

```
let vc = self.storyboard?.instantiateViewController(withIdenti fier: "CalculationVC") as? CalculationViewController
vc?.chosenImage = image
//4b31167b5a50fd2b93ad9e53ead691c6c9c298e1829ac0f3fbb 216784ecf906b
self.present(vc ?? UIViewController(),
animated: true)
}
}
var imageHashes = [
"4b31167b5a50fd2b93ad9e53ead691c6c9c298e1829ac0f3fbb2 16784ecf906b": UIImage(named: "map1"),
"36fc2d356fe24990fbd48b8147065496976b07567a752fcf39c4
53cce6fc5840": UIImage(named: "map2"),
"07b92e01d91eaa71ab394843f66432ca32f14bff97b9c4ebf6b7 a8b7eaf1bca3": UIImage(named: "map3")
]
extension UIImage {
public func sha256() -> String{
if let imageData = cgImage?.dataProvider?.data as? Data {
return hexStringFromData(input: digest(input: imageData as NSData))
}
return ""
}
private func digest(input : NSData) -> NSData {
let digestLength = Int(CC_SHA256_DIGEST_LENGTH)
var hash = [UInt8](repeating: 0, count:
digestLength)


&hash)
```

```
CC_SHA256(input.bytes, UInt32(input.length),
return NSData(bytes: hash, length:

digestLength)
}
private func hexStringFromData(input: NSData) -> String {
var bytes = [UInt8](repeating: 0, count:
input.length)
input.getBytes(&bytes, length: input.length)
var hexString = ""
for byte in bytes {
hexString += String(format:"%02x",
UInt8(byte))
}
return hexString
}
}
import UIKit
class ResultViewController: UIViewController {
@IBOutlet var imageRes: UIImageView!
@IBOutlet var exitButton: UIButton!
var image = UIImage()
override func viewDidLoad() { super.viewDidLoad() exitButton.clipsToBounds = true
exitButton.layer.cornerRadius = 16 imageRes.image = image
}
@IBAction func exit(_ sender: Any) {
self.view.window!.rootViewController?.dismiss(animate d: false, completion: nil)
}
}

import UIKit
class CalculationViewController: UIViewController, UITextFieldDelegate {
```

```
@IBOutlet var bgView: UIView!
@IBOutlet var loader: UIActivityIndicatorView!
@IBOutlet var textField: UITextField!
@IBOutlet
@IBOutlet var
var lable: UILabel! tableImageView: UIImageView!
@IBOutlet var calculateButton: UIButton!
@IBOutlet var image: UIImageView!
var chosenImage: UIImage?
override func viewDidLoad() { super.viewDidLoad() calculateButton.clipsToBounds = true calculateButton.layer.cornerRadius = 16 loader.isHidden = true
bgView.isHidden = true if chosenImage == nil {
image.isHidden = true
textField.delegate = self
//tableImageView.image = UIImage(named:
"таблица")
} else {
textField.isHidden = true lable.isHidden = true image.image = chosenImage
}
}
func textFieldShouldReturn(_ textField: UITextField) -> Bool {
textField.resignFirstResponder() tableImageView.image = UIImage(named:
"таблиця")
return true
}
@IBAction func calculate(_ sender: Any) {

loader.isHidden = false bgView.isHidden = false loader.startAnimating()
DispatchQueue.main.asyncAfter(deadline:
.now() + 1, execute: {
let vc = self.storyboard?.instantiateViewController(withIdenti fier: "ResultVC") as?
ResultViewController
```

```
vc?.image = (imageHashes[self.image.image?.sha256() ?? ""] ?? UIImage())!
self.present(vc ?? UIViewController(),
animated: true)
self.loader.stopAnimating()
})
}
}
import UIKit
class CoordinatesVC: UIViewController, UITextFieldDelegate {
@IBOutlet var yLabel: UILabel!
@IBOutlet
@IBOutlet var
var xLabel: xStack: UILabel! UIStackView!
@IBOutlet var yStack:UIStackView!
@IBOutlet var textField: UITextField!
override func viewDidLoad() {
super.viewDidLoad()
[yStack, xStack, xLabel, yLabel].forEach {
$0?.isHidden = true }
textField.delegate = self
}
func textFieldShouldReturn(_ textField: UITextField) -> Bool {
textField.resignFirstResponder()
[yStack, xStack, xLabel, yLabel].forEach {
$0?.isHidden = false }
return true

}
@IBAction func calculate(_ sender: Any) { DispatchQueue.main.asyncAfter(deadline:
.now() + 1, execute: {
let vc = self.storyboard?.instantiateViewController(withIdenti fier: "ResultVC") as?
ResultViewController
```

Додаток Г
(обов'язковий)

Ілюстративна частина

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ РОЗУМНИМ БУДИНКОМ

Перелік ілюстративних матеріалів:

Слайд 1 – Початковий слайд

Слайд 2 — Мета та завдання роботи

Слайд 3 — Огляд наявних рішень

Слайд 4 — Механізм роботи системи

Слайд 5 — Розробка системи розумного будинку

Слайд 6 — Розробка інструментів розмуного будинку

Слайд 7 — Вибір технологій програмування

Слайд 8 — Опис архітктури

Слайд 9 — Вибір середовища розробки

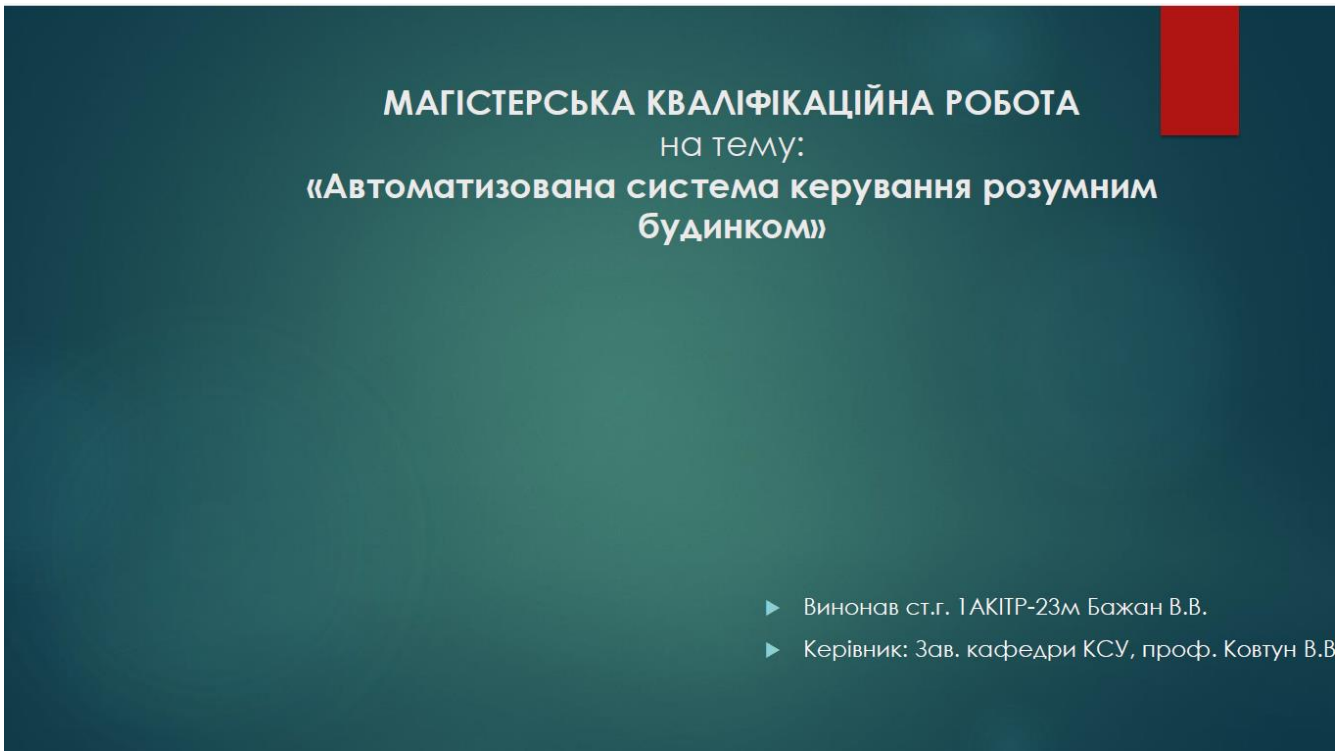
Слайд 10 — Тестування та результати

Слайд 11 — Економічна частина

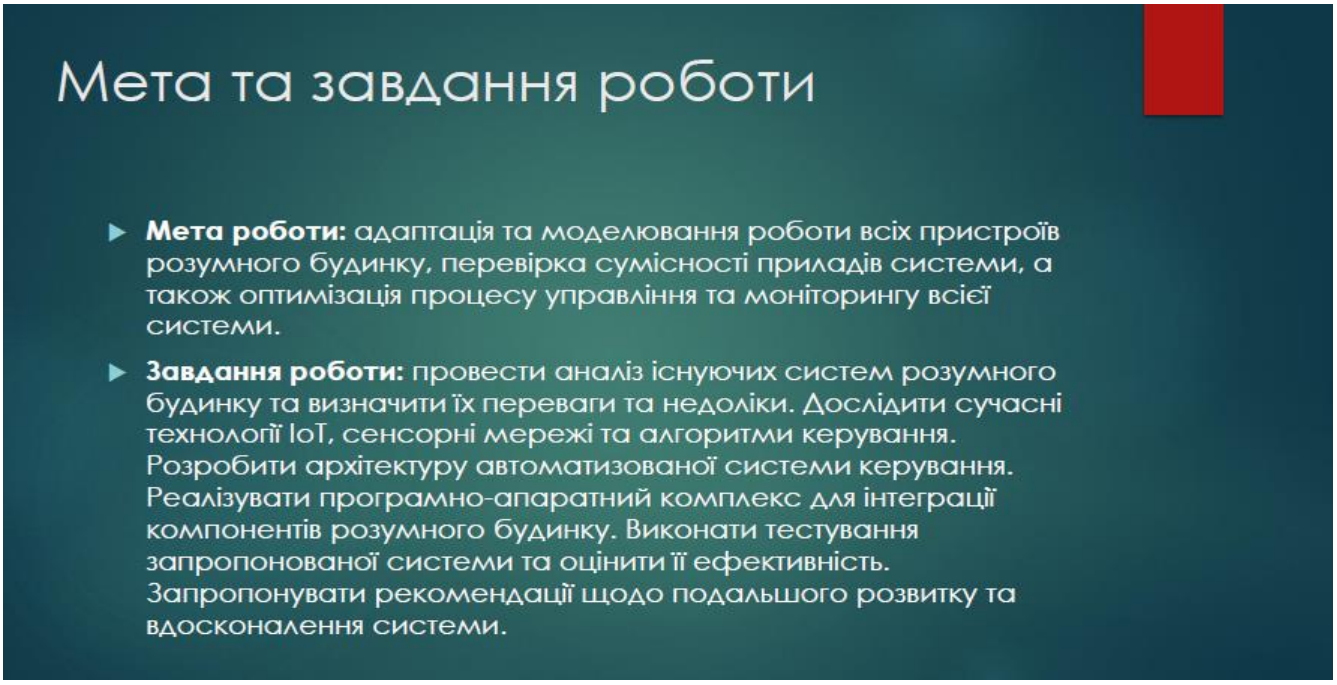
Слайд 12 — Висновки

Студент Бажан В.В.

Керівник роботи Ковтун В.В.

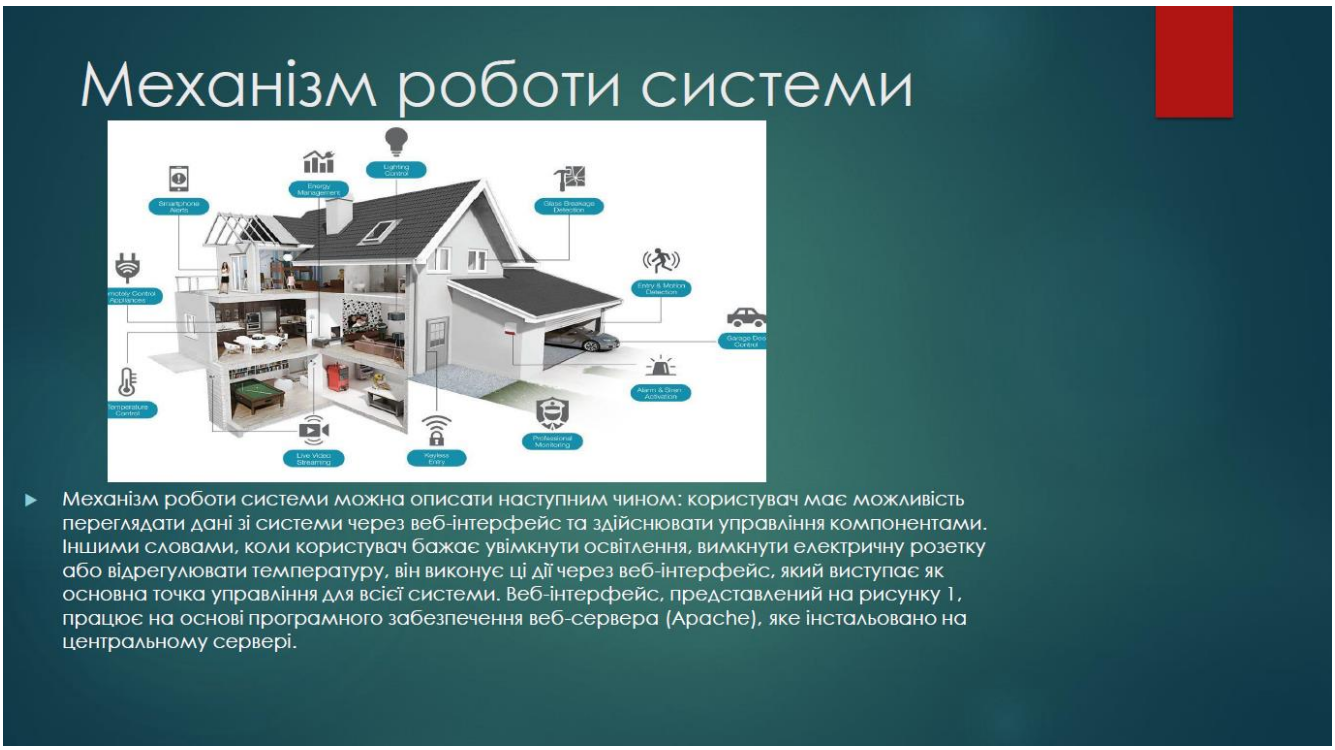


Слайд 1 – Початковий слайд

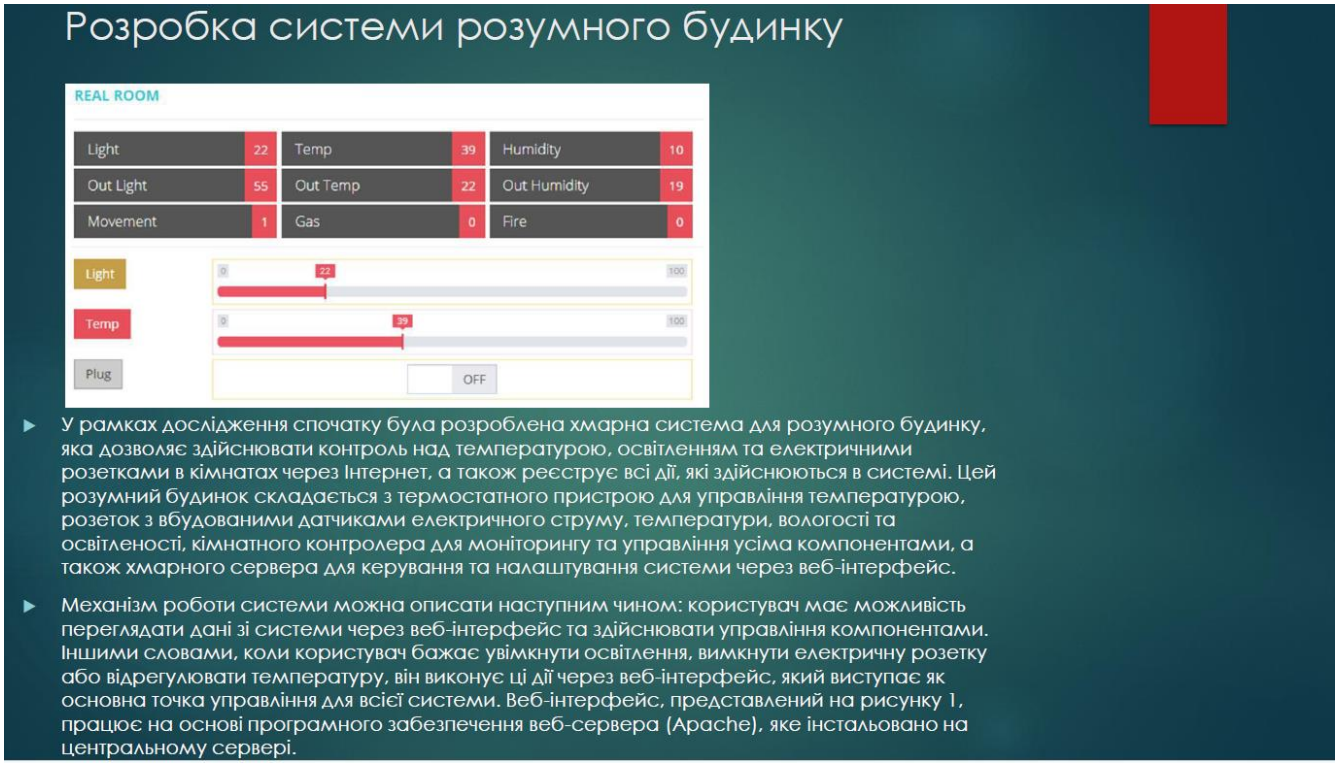


Слайд 2 — Мета та завдання роботи

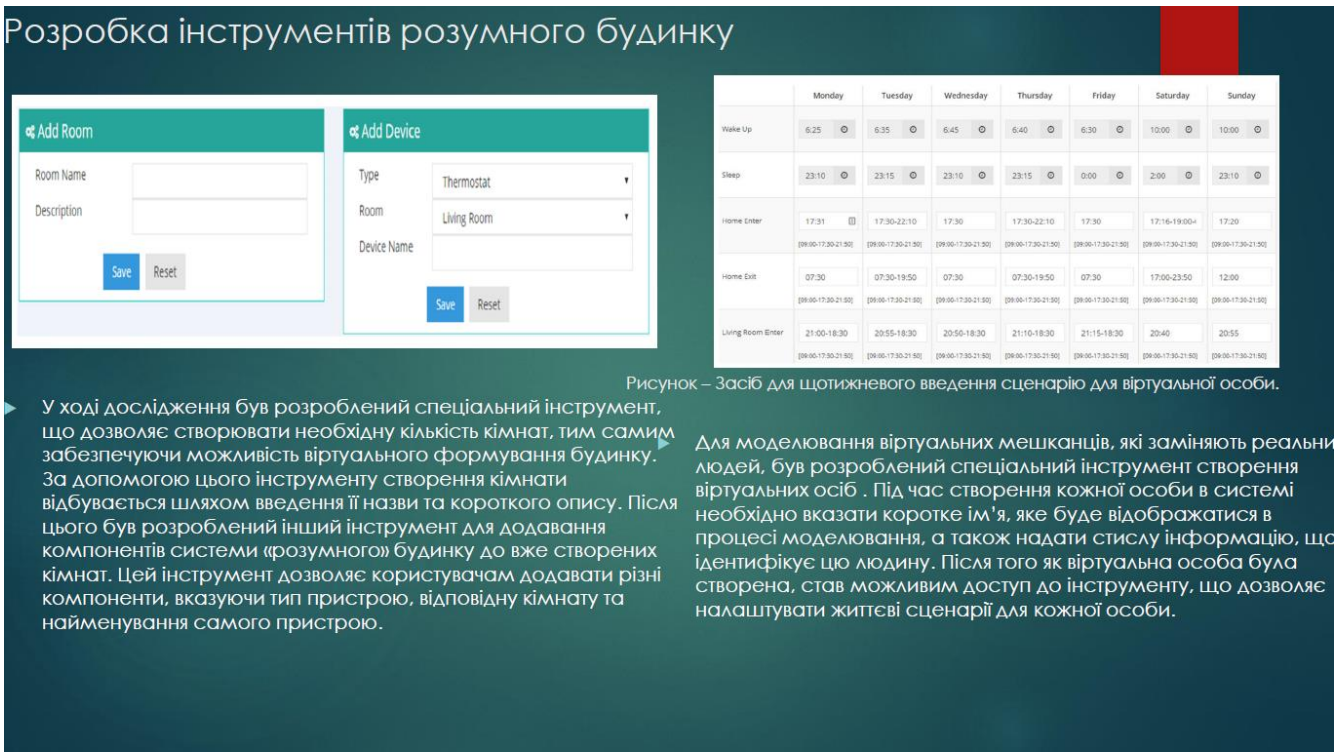
Слайд 3 — Огляд наявних рішень



Слайд 4 — Механізм роботи системи

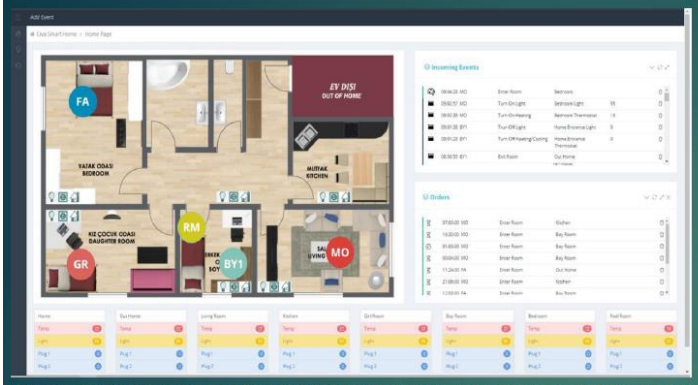


Слайд 5 — Розробка системи розумного будинку



Слайд 6 — Розробка інструментів розмуного будинку

Вибір технологій програмування



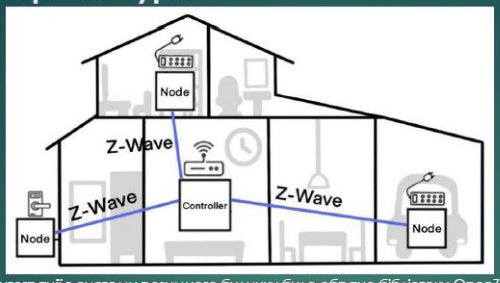
Після створення моделі розумного будинку, його компонентів, мешканців та сценаріїв, було розроблено спеціалізоване програмне забезпечення, що дозволяє здійснювати моделювання на основі цих даних. Для реалізації використовувалася мова програмування JavaScript, яка працює в середовищі Node.js. Завдяки застосуванню PM2 (інструмент для керування процесами в Node.js, що забезпечує розширену функціональність), моделювання може бути автоматично перезапущене у разі збоїв або перезавантаження системи, що гарантує безперервну роботу.

Веб-інтерфейс був реалізований за допомогою технологій CSS, JavaScript та HTML, що дозволяє здійснювати моніторинг моделювання та втручатися в процес за необхідності. Цей інтерфейс надає можливість в реальному часі та з висоти пташиного польоту відображати розумний будинок, його компоненти та індивідуумів

- На Рис. зображено інтерфейс веб-моделювання, де кола що символізують віртуальних людей, відображаються з висоти пташиного польоту над розумним будинком. Ці кола автоматично змінюють своє місцезнаходження в залежності від переміщення людей між кімнатами. Положення значків розміщується випадковим чином, щоб уникнути їх перекриття. Значки ламп, температури та розеток в кімнатах позначають наявні компоненти розумного будинку в конкретному приміщенні.
- Праворуч від інтерфейсу розташовані дві таблиці. Верхня таблиця містить інформацію про майбутні транзакції, що відображають події, які відбуватимуться за заздалегідь визначеними сценаріями. Нижня таблиця показує розпорядження, які надаються менеджером поза сценарієм. Додатково, внизу екрану містяться маленькі таблиці, які демонструють дані про температуру, рівень освітлення та стан розеток у всіх кімнатах розумного будинку.
- Меню розташоване в верхній та лівій частинах екрана, зокрема ліворуч. Меню з лівого боку включає блоки управління, де можна безпосередньо керувати компонентами розумного будинку в кожній кімнаті. Через це меню можна швидко переглянути та змінити налаштування температури, освітлення або розеток у вибраній кімнаті.

Слайд 7 — Вибір технологій програмування

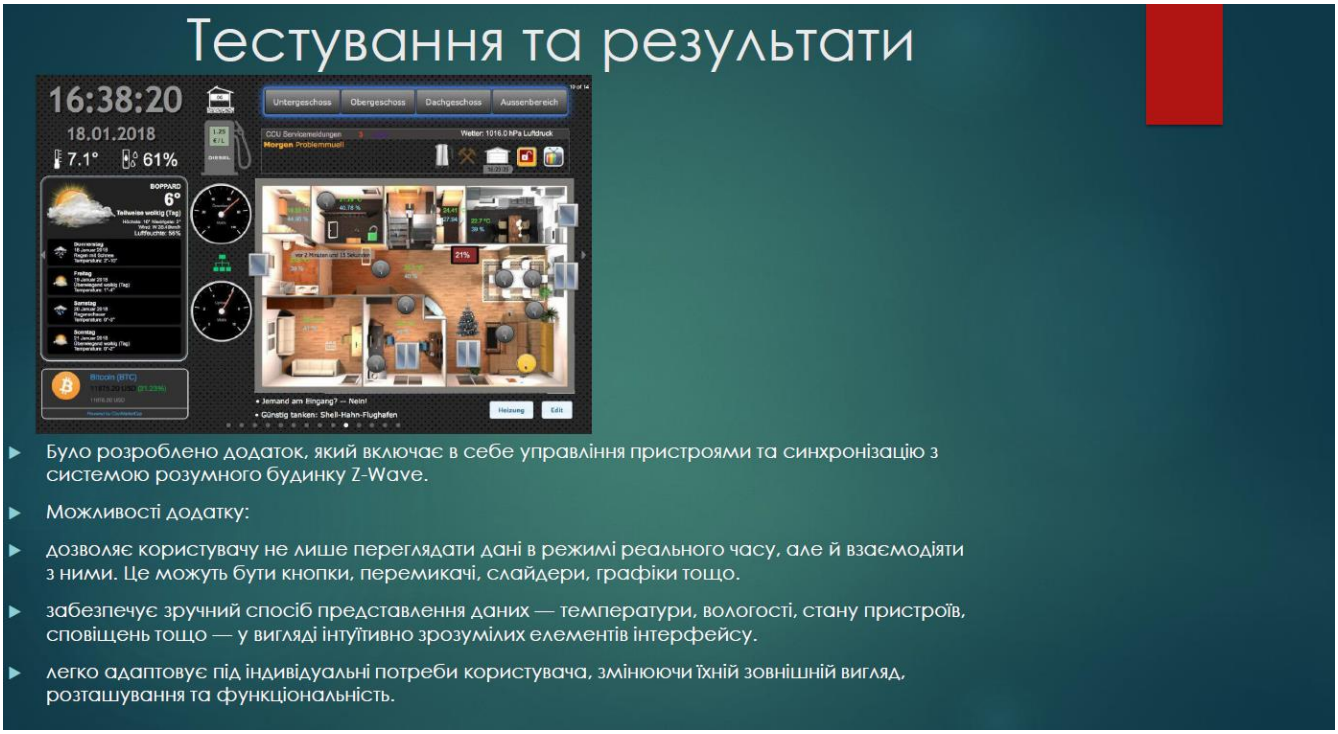
Опис архітектури



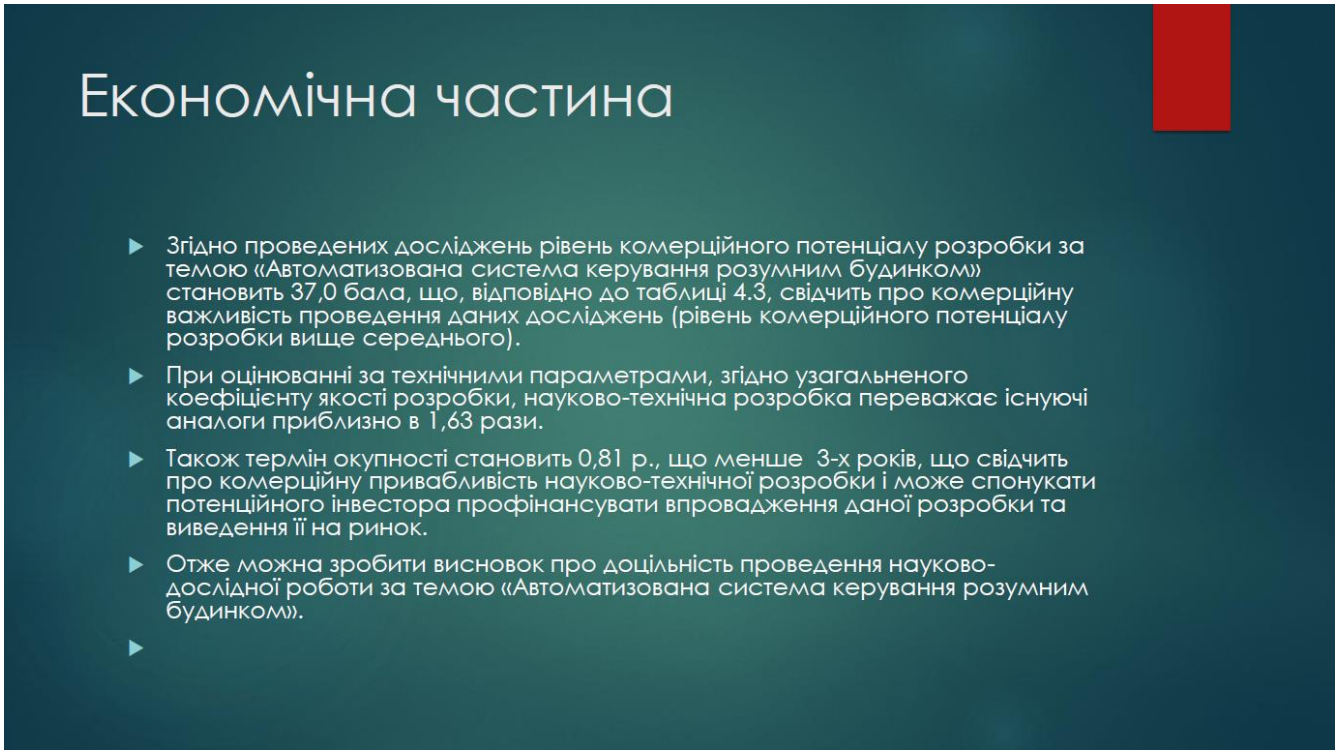
- Для створення інтерфейсу для інтеграції в систему розумного будинку було обрано бібліотеку OpenZWave. Це відкрита міжплатформна бібліотека, яка дозволяє додавати підтримку пристроїв Z-Wave до додатків без необхідності глибоких знань.
- Переваги:
- **1. Низьке енергоспоживання**
- Z-Wave працює на низьких частотах і споживає мало енергії, що робить його ідеальним для батарейних пристроїв, таких як датчики руху, дверні замки та термостати.
- **◆ 2. Висока сумісність**
- Стандарт Z-Wave гарантує, що всі сертифіковані пристрої (незалежно від виробника) будуть сумісними між собою.
- **◆ 3. Меш-мережа (Mesh Network)**
- Кожен пристрій (крім батарейних) може працювати як ретранслятор сигналу, що збільшує дальність і надійність зв'язку.
- **◆ 4. Менша ймовірність перешкод**
- Z-Wave працює на частоті 868,42 МГц у Європі (908,42 МГц у США), що уникає перевантаженості 2,4 ГГц, яка використовується Wi-Fi та Bluetooth.
- **◆ 5. Висока безпека**
- Протокол використовує 128-бітове шифрування AES, що забезпечує захищений зв'язок між пристроями.
- **◆ 6. Великий радіус дії**
- Кожен пристрій може передавати сигнал до 100 метрів у відкритому просторі, а за рахунок меш-мережі покриття може значно збільшуватися.

Слайд 8 — Опис архітктури

Слайд 9 — Вибір середовища розробки



Слайд 10 — Тестування та результати



Слайд 11 — Економічна частина

Слайд 12 — Висновки