Хмельницький національний університет
Факультет інформаційних технологіїй
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32
Назва теми

КвРКІ 210103.21.01.14 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»
Назва

Виконав: студент IV курсу, група КІ2–21–1 _____ Павло БІЛЯК
Підпис — Ініціали, прізвище

Керівник _____ Дмитро МЕДЗАТИЙ
Підпис, дата — Ініціали, прізвище

Нормоконтролер _____ Тетяна КИСІЛЬ
Підпис, дата — Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної інженерії та інформаційних систем _____ Ольга ПАВЛОВА
Підпис — Ініціали, прізвище

«9» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

" 10 " 01 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Павлу БІЛЯКУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32

Керівник проекту (роботи) Дмитро МЕДЗАТИЙ, к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. №23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз програмно-апаратних засобів керування доступом до кіберфізичної системи "розумний будинок"

Проектування програмно-апаратного засобу керування доступом до кіберфізичної системи "розумний будинок" із функцією сповіщення на основі esp32

Тестування та реалізація програмно-апаратного засобу керування доступом до кіберфізичної системи "розумний будинок" із функцією сповіщення на основі esp32

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Структура прогармно-апаратного засобу

Схема електрична принципова

Сценарії NODE-RED для клієнта керування

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийн
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КІІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІІС		

7. Дата видачі завдання «_10_»_____01_____2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітк
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2025	виконан
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконан
3	Робота над розділом 1 – Огляд відомих систем керування доступом до кіберфізичної системи "розумний будинок"	01.03.2025	виконан
4	Робота над розділом 2 – Проєктування програмно-апаратного засобу до кіберфізичної системи "розумний будинок"	01.04.2025	виконан
5	Робота над розділом 3 – Тестування та реалізація програмно-апаратного засобу до кіберфізичної системи "розумний будинок"	30.04.2025	виконан
6	Оформлення пояснювальної записки згідно вимог	20.05.2025	виконан
7	Попередній захист ВКР	26.05.202	виконан
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент Підпис Павло БІЛЯК
Ініціали, прізвище

Керівник роботи Підпис Дмитро МЕДЗАТИЙ
Ініціали, прізвище

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ. 210103.21.01.14 ПЗ	Пояснювальна записка	60		
			Графічні матеріали			
2		КвРКІ. 210103.21.01.14 Е8	Структура програмно-апаратного засобу	1		
3		КвРКІ. 210103.21.01.14 Е2	Схема електрична принципова	1		
4		КвРКІ. 210103.21.01.14 Е8	Сценарії NODE-RED для клієнта керування	1		

Зм	Арк	№ докум	Підпис	Дата	КвРКІ. 210103.21.01.14 ПЗ		
Розробив		Біляк П.О.		09.09.25	Відомість проєкту	Літера	Аркуш 1 / Аркушів 1
Перевір.		Медзатий Д.М.		9.09.25		У	
Н. контр.		Кисіль Т.М.		9.09.25	ХНУ, КІ2-21-1		
Затв.		Павлова О.О.		9.09.25			

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32».

Автор роботи: *Павло БІЛЯК*

Керівник роботи: *Медзатий Дмитро Миколайович.*

Пояснювальна записка: *60 с.*, 38 *рис.*, 1 *табл.*, 3 *дод.*, 45 *джерел.*

Графічна частина: 3 креслення.

КЕРУВАННЯ ДОСТУПОМ, КІБЕРФІЗИЧНА СИСТЕМА, МІКРОКОНТРОЛЕР, NODE-RED.

Мета кваліфікаційної роботи: розробка програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32.

Сучасний світ стрімко розвивається в напрямку інтеграції інформаційних технологій із фізичними процесами, що створює нові можливості для підвищення комфорту, безпеки та ефективності житлових і промислових середовищ. Кіберфізичні системи, зокрема концепція "Розумного будинку", стають невід'ємною частиною сучасного побуту, забезпечуючи автоматизоване керування освітленням, кліматом, безпекою, а також доступом до приміщень. Проте разом із впровадженням таких систем виникає гостра потреба у розробці надійних засобів контролю доступу, що здатні не лише забезпечити безпеку, а й гнучко адаптуватися до різних сценаріїв використання та змінних умов експлуатації.

Використання мікроконтролера ESP32 в якості апаратної основи для реалізації локального контролю доступу відкриває широкі можливості завдяки його високій обчислювальній потужності, вбудованим засобам бездротового зв'язку та підтримці різноманітних інтерфейсів введення-виведення. Це дозволяє створити автономний пристрій, здатний здійснювати автентифікацію користувачів через введення паролю, самостійно управляти замками та вести облік спроб доступу без необхідності постійного зовнішнього втручання.

09.06.2025

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

КФС – Кіберфізична система

ПЗ – Програмне забезпечення

ПЗП – Постійний запам’ятовуючий пристрій

API – Application Programming Interface

GPIO – General Purpose Input/Output

IoT – Internet of Things

MQTT – Message Queuing Telemetry Transport

UI – User Interface

TCP/IP – Transmission Control Protocol/Internet Protocol

PWM – Pulse Width Modulation

ВСТУП

Сучасний світ стрімко розвивається в напрямку інтеграції інформаційних технологій із фізичними процесами, що створює нові можливості для підвищення комфорту, безпеки та ефективності житлових і промислових середовищ. Кіберфізичні системи, зокрема концепція "Розумного будинку", стають невід’ємною частиною сучасного побуту, забезпечуючи автоматизоване керування освітленням, кліматом, безпекою, а також доступом до приміщень. Проте разом із впровадженням таких систем виникає гостра потреба у розробці надійних засобів контролю доступу, що здатні не лише забезпечити безпеку, а й гнучко адаптуватися до різних сценаріїв використання та змінних умов експлуатації.

Використання мікроконтролера ESP32 в якості апаратної основи для реалізації локального контролю доступу відкриває широкі можливості завдяки його високій обчислювальній потужності, вбудованим засобам бездротового зв’язку та підтримці різноманітних інтерфейсів введення-виведення. Це дозволяє створити автономний пристрій, здатний здійснювати автентифікацію користувачів через введення паролю, самостійно управляти замками та вести облік спроб доступу без необхідності постійного зовнішнього втручання. Водночас інтеграція ESP32 із хмарним MQTT-брокером HiveMQ забезпечує надійний, масштабований і гнучкий канал обміну повідомленнями, що дозволяє реалізувати віддалене керування та моніторинг системи в режимі реального часу.

Клієнт керування, побудований на базі середовища Node-RED, доповнює цю архітектуру, надаючи користувачам зручний інтерфейс для візуального управління, ведення журналу подій та оперативного реагування на потенційні загрози. Node-RED, завдяки своїй модульній і візуальній структурі, дозволяє легко інтегрувати різноманітні протоколи та сервіси, що робить можливим швидке налаштування логіки роботи системи без потреби глибоких знань програмування. Це дає змогу адміністраторам і користувачам не лише контролювати доступ у режимі реального часу, а й автоматизувати обробку подій, формувати сповіщення,

виконувати аналіз журналів і приймати оперативні рішення на основі отриманих даних. Поєднання локального апаратного контролю з хмарною інфраструктурою та можливостями віддаленого моніторингу створює комплексне рішення, яке відповідає сучасним вимогам до кіберфізичних систем і сприяє підвищенню безпеки, зручності та адаптивності "Розумного будинку". Враховуючи динамічний розвиток технологій і зростаючі вимоги користувачів, впровадження таких програмно-апаратних засобів керування доступом є важливим кроком на шляху до створення безпечних, інтелектуальних і гнучких житлових середовищ майбутнього.

Метою роботи є розробка програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32.

Об’єктом дослідження є процеси керування доступом до кіберфізичних систем, віддаленого керування та сповіщення користувачів про події безпеки в режимі реального часу.

Предметом дослідження є програмно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32.

1 АНАЛІЗ ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ КЕРУВАННЯ ДОСТУПОМ ДО КІБЕРФІЗИЧНОЇ СИСТЕМИ "РОЗУМНИЙ БУДИНОК"

1.1 Огляд кіберфізичних систем та їх ролі у "Розумному будинку"

Кіберфізичні системи (КФС) є основою сучасних інтелектуальних технологій, що поєднують у собі фізичні процеси з цифровими обчисленнями і комунікаціями. Апаратні складові концепції «Кіберфізична система» становлять фундамент, на якому базується взаємодія між фізичним світом та цифровими обчисленнями (рис. 1.1). По-перше, це різноманітні датчики, які виконують роль первинних джерел інформації, збираючи дані про навколишнє середовище чи внутрішні параметри системи. Ці датчики можуть вимірювати температуру, вологість, рух, освітленість, тиск та інші фізичні величини, перетворюючи їх у електричні сигнали, придатні для подальшої обробки.

Другим важливим елементом є виконавчі механізми, які отримують команди від обчислювальної частини системи і безпосередньо впливають на фізичне середовище. Вони можуть бути представлені електромоторами, електромагнітними замками, клапанами, реле чи іншими пристроями, що виконують дії згідно з керівними сигналами.

Для забезпечення обробки та керування усіма цими пристроями в апаратну складову входять контролери або мікроконтролери, які виступають центральним мозком системи. Вони приймають сигнали від датчиків, аналізують інформацію, приймають рішення та генерують команди для виконавчих механізмів. Часто до них інтегруються модулі бездротового зв’язку, такі як Wi-Fi, Bluetooth або Zigbee, що дозволяє забезпечити зв’язок з іншими пристроями або хмарними сервісами.

Ще одними складовими є HMI (Human-Machine Interface), які забезпечують зручний спосіб управління та моніторингу системи. HMI можуть бути представлені сенсорними панелями, дисплеями з кнопками, мобільними

додатками або веб-інтерфейсами, що дозволяють користувачам отримувати інформацію, вводити команди та налаштовувати параметри системи в реальному часі.

Крім того, важливу роль у структурі кіберфізичної системи відіграють сервери та обчислювальні платформи, які виконують функції зберігання, обробки і аналізу великих обсягів даних, а також управління логікою системи на більш високому рівні. Сервери можуть бути розташовані локально в межах будинку або працювати в хмарних інфраструктурах, забезпечуючи масштабованість, резервування та віддалений доступ до системи. Вони дозволяють реалізовувати складні алгоритми обробки даних, зберігати історію подій, виконувати аналітику та інтегруватися з іншими системами, такими як служби безпеки чи енергоменеджменту.

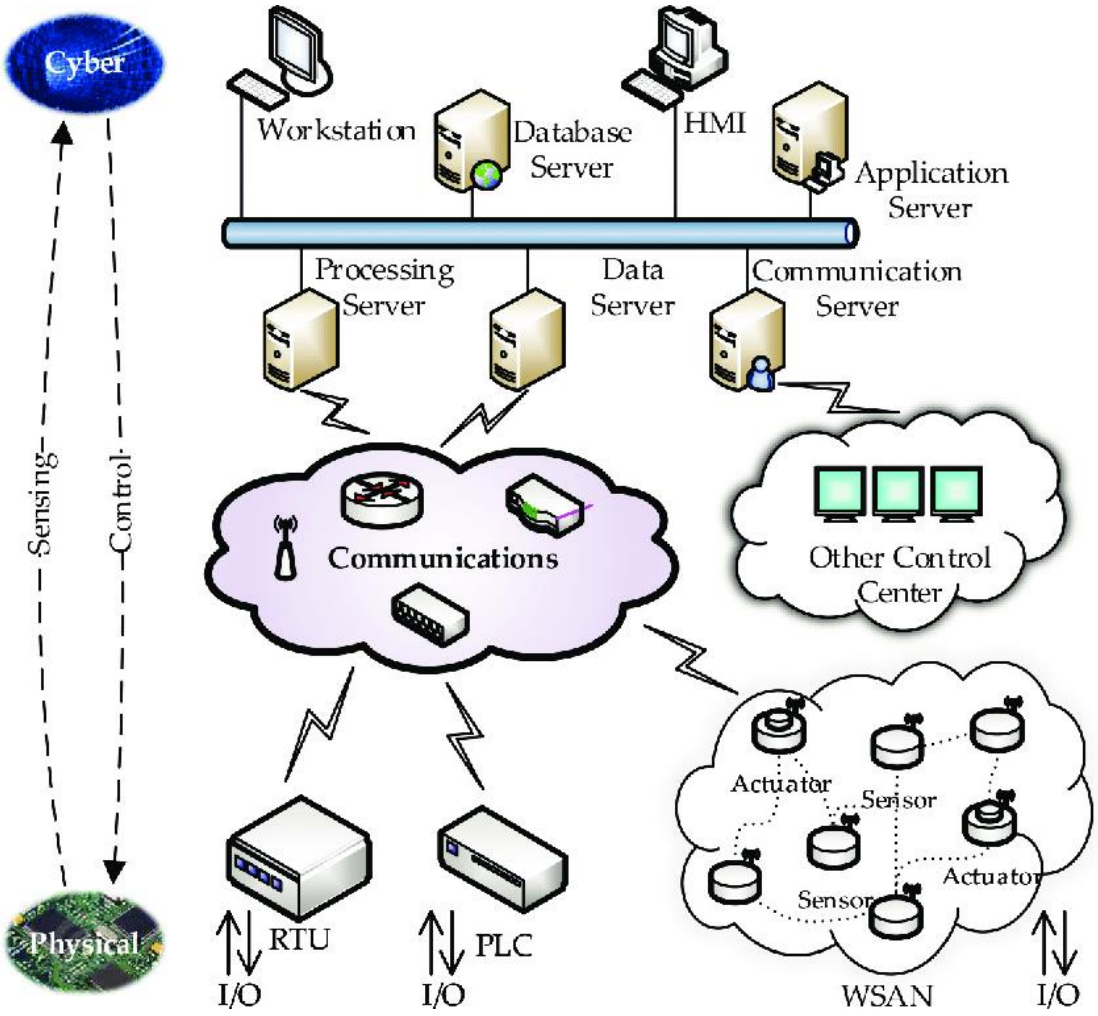


Рисунок 1.1 – Апаратні складові концепції «Кіберфізична система» [1]

У контексті "Розумного будинку" кіберфізичні системи забезпечують інтеграцію різноманітних пристроїв та сенсорів, що відповідають за моніторинг, контроль і автоматизацію житлового середовища [14-21]. Кіберфізичні системи збирають дані з фізичних джерел, наприклад, датчиків температури, руху, освітлення, а також керують виконавчими механізмами — дверними замками, освітленням, опаленням і системами безпеки. Ці системи працюють у тісній взаємодії між собою та з користувачем, забезпечуючи комфорт, ефективність і безпеку житла.

Основна роль КФС у "Розумному будинку" полягає у створенні адаптивного і гнучкого середовища, яке реагує на зміни в навколишньому середовищі і потреби користувача в режимі реального часу. Вони дають змогу не лише автоматизувати рутинні процеси, але й впроваджувати інтелектуальні рішення, які аналізують поведінку мешканців, прогнозують можливі ризики і оптимізують енергоспоживання. Завдяки цьому підвищується рівень безпеки, економії ресурсів і загальний комфорт проживання [22-25].

Кіберфізичні системи в розумних будинках також забезпечують взаємодію з зовнішніми сервісами, зокрема хмарними платформами, що дає змогу здійснювати віддалене керування і моніторинг через інтернет. Це відкриває широкі можливості для модернізації інфраструктури будинку, забезпечення своєчасного реагування на надзвичайні ситуації та інтеграції з іншими інтелектуальними системами, наприклад, охоронними або енергоменеджментом. В результаті, КФС формують основу для побудови комплексних, надійних і зручних у використанні "Розумних будинків", які відповідають сучасним технологічним вимогам і очікуванням користувачів.

1.2 Особливості керування доступом у кіберфізичних системах

Керування доступом у кіберфізичних системах, зокрема в умовах «розумного будинку», є однією з ключових функцій, що забезпечує фізичну

безпеку мешканців та захист цифрових даних. Такі системи поєднують сенсорну, обчислювальну та мережеву інфраструктуру, яка функціонує в реальному часі. Особливість керування доступом у цьому контексті полягає в необхідності взаємодії між фізичними пристроями (дверні замки, датчики руху, камери) та цифровими елементами (сервери, додатки, алгоритми аутентифікації), що робить процес значно складнішим порівняно зі звичайними охоронними системами.

Основою побудови такої системи є реалізація двох взаємопов'язаних процесів: автентифікації та авторизації. Автентифікація – це процес перевірки особи, яка намагається отримати доступ до системи. У розумному будинку це може бути реалізовано за допомогою введення PIN-коду на клавіатурі, сканування відбитків пальців, розпізнавання обличчя, використання RFID-технологій або мобільного додатку. Автентифікація відповідає на запитання «Хто ти?». Після підтвердження особи виконується авторизація – визначення рівня доступу цієї особи. Система перевіряє, чи дозволено користувачеві, наприклад, відчинити вхідні двері, увімкнути сигналізацію чи змінити налаштування пристроїв у будинку. Це етап, який відповідає на запитання «Що тобі дозволено робити?».

Проте впровадження ефективної системи контролю доступу в кіберфізичному середовищі супроводжується низкою викликів і технічних вимог. По-перше, розумний будинок є розподіленою системою, де численні пристрої спілкуються між собою через мережу – найчастіше бездротову. Це створює потенційні точки вразливості до атак типу «людина посередині» (MITM), перехоплення даних або фальсифікації команд. Тому критично важливо впроваджувати надійні механізми шифрування та захисту комунікацій, як наприклад HTTPS, WPA2 або протоколи з асиметричним шифруванням.

По-друге, система повинна бути стійкою до збоїв і атак. Наприклад, якщо інтернет-зв'язок перервано, локальна логіка керування має залишатись працездатною і безпечною. У разі DDoS-атаки або спроби фізичного злому система повинна вміти реагувати миттєво — блокувати доступ, активувати

сигналізацію, повідомити користувача через мобільний додаток або електронну пошту.

Ще одним викликом є конфіденційність та збереження персональних даних, які обробляються в розумному будинку: історія входів/виходів, персональні налаштування, доступні профілі користувачів. Така інформація має бути надійно захищена від стороннього втручання та несанкціонованого збору. Система також повинна враховувати багатокористувацький режим, у якому різні члени родини мають різні рівні доступу. Наприклад, батьки можуть змінювати налаштування, а діти – лише керувати певними пристроями в межах дозволеного часу.

Таким чином, особливості керування доступом у кіберфізичних системах вимагають глибокого розуміння як апаратного рівня (сенсори, мікроконтролери, виконавчі пристрої), так і програмного (логіка керування, захищені протоколи, розмежування прав). Реалізація таких рішень повинна бути не лише функціональною, а й безпечною, масштабованою та зручною для користувачів.

1.3 Сучасні рішення та платформи для керування доступом у розумних будинках

У контексті проєктування програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32 важливо проаналізувати сучасні рішення та платформи, які використовуються у сфері розумного доступу, а також зрозуміти їх функціональні можливості, щоб обґрунтувати вибір конкретної архітектури системи, зокрема на основі ESP32.

На ринку існує широкий спектр комерційних рішень, які пропонують готові системи контролю доступу для розумних будинків. Серед найпоширеніших можна виокремити такі рішення як Nuki Smart Lock, August Smart Lock, Yale Linus, Ring Alarm, Bosch Smart Home, Somfy та інші. Ці пристрої здебільшого працюють за моделлю «ключ через смартфон» і мають мобільний

застосунок, хмарне зберігання історії подій, сповіщення через push або email, можливість інтеграції з голосовими помічниками типу Google Assistant, Alexa або Apple HomeKit.

Зокрема одним із таких є розумний дверний замок Nuki Smart Lock Pro 4, що є сучасним комерційним рішенням для організації безпечного та зручного доступу до приміщень у межах розумного будинку (рис. 1.2). Цей пристрій монтується з внутрішнього боку стандартного механічного замка без потреби в його заміні, що робить установку швидкою й нескладною. Управління замком здійснюється за допомогою смартфона через Bluetooth або Wi-Fi. Вбудований Wi-Fi-модуль у версії Pro 4 дозволяє керувати замком дистанційно, без додаткового шлюзу, що значно розширює функціональні можливості. Наприклад, користувач може відчинити двері, перебуваючи в іншому місті, або надати тимчасовий доступ іншим особам.

Замок підтримує інтеграцію з популярними платформами для розумного дому, такими як Apple HomeKit, Google Assistant, Amazon Alexa, а також із сервісами IFTTT. Крім того, у Nuki Smart Lock Pro 4 передбачено можливість створення детальних журналів подій, де фіксуються всі дії: хто й коли заходив. Це підвищує рівень безпеки й дозволяє здійснювати моніторинг доступу в реальному часі. До того ж, користувачі можуть встановлювати автоматичне блокування або розблокування замка залежно від геолокації пристрою або розкладу.

У конструкції передбачено акумуляторну батарею, що забезпечує кілька місяців автономної роботи, та зарядку через USB-C. Пристрій має елегантний сучасний дизайн і пропонується в кількох варіантах кольору, що дозволяє гармонійно інтегрувати його в інтер’єр. Загалом, Nuki Smart Lock Pro 4 є прикладом високотехнологічного, надійного й водночас зручного рішення для забезпечення контролю доступу в системах «розумного дому», орієнтованого на кінцевого користувача без технічної підготовки.

Рисунок 1.2 – Розумний дверний замок Nuki Smart Lock Pro 4 [2]

Ще одним рішення є програмно-апаратний пристрій August Smart Lock (рис. 1.3). Порівнюючи August Smart Lock із попереднім розглянутим рішенням Nuki Smart Lock Pro 4, можна відзначити як схожість у загальній концепції, так і низку функціональних і технічних відмінностей, які впливають на вибір того чи іншого пристрою для інтеграції у систему «розумного будинку».

Обидва замки призначені для внутрішнього монтажу поверх наявного механічного циліндра, що дозволяє зберегти ключову частину замка без необхідності його заміни. Вони підтримують безключовий доступ, віддалене керування, надання цифрових ключів іншим користувачам та ведення журналу подій. Обидва також мають функцію автоматичного блокування та можуть інтегруватися з провідними платформами розумного дому — Apple HomeKit, Google Assistant, Amazon Alexa.

Втім, Nuki Smart Lock Pro 4 має вбудований Wi-Fi-модуль, що дозволяє йому працювати без додаткового шлюзу (Bridge), тоді як August Smart Lock, залежно від версії, часто потребує окремого Wi-Fi Bridge для доступу через Інтернет. Це впливає як на зручність встановлення, так і на кінцеву вартість системи.

У плані дизайну August має більш компактну й мінімалістичну форму, що може бути важливо для користувачів, які прагнуть естетичної інтеграції в інтер’єр. З іншого боку, Nuki пропонує розширені можливості кастомізації доступу, а також краще адаптований до європейських стандартів дверей, тоді як

August орієнтований переважно на ринок США, де використовуються інші типи замків.

Щодо енергоспоживання, Nuki використовує вбудований акумулятор, що заряджається через USB-C, тоді як August працює на змінних батарейках, які потрібно замінювати вручну. Це створює різницю в обслуговуванні: у Nuki – заряджання, у August – регулярна заміна елементів живлення.

Рисунок 1.3 – August Smart Lock [3]

Іншим рішенням є електронний контролер YALE Linus (рис. 1.4), що працює за схожим принципом до пристроїв Nuki Smart Lock Pro 4 та August Smart Lock. Він встановлюється з внутрішньої сторони дверей поверх існуючого циліндрового замка, зберігаючи можливість відкривати двері звичайним ключем із зовнішнього боку. Такий підхід дозволяє легко інтегрувати пристрій у вже наявну інфраструктуру без кардинальних змін.

Подібно до Nuki, YALE Linus розроблений із урахуванням європейських стандартів циліндрів, що робить його сумісним із більшістю замків, використовуваних у країнах Європи, включаючи Україну. Це дає перевагу перед August, який краще підходить до американських типів замків. YALE Linus підтримує відкриття через мобільний додаток завдяки Bluetooth-з'єднанню, а для повного віддаленого керування потрібен окремий модуль Yale Connect Wi-Fi Bridge, аналогічно до August.

У функціональному плані YALE Linus забезпечує основні можливості, типові для розумних замків: автоматичне блокування/розблокування залежно від геолокації, створення та надсилання тимчасових або постійних цифрових ключів, перегляд журналу подій, а також інтеграцію з голосовими асистентами – Google Assistant, Amazon Alexa та Apple HomeKit. У поєднанні з додатком Yale Access замок надає повноцінний контроль і моніторинг дій, що відбуваються з дверима.

На відміну від Nuki Smart Lock Pro 4, у YALE Linus живлення здійснюється не через вбудований акумулятор, а від змінних батарейок, як і в August. Це накладає певні вимоги до періодичної заміни елементів живлення, але й дозволяє швидко відновити роботу пристрою у випадку розрядження.

У плані дизайну YALE Linus вирізняється стриманим і елегантним виглядом, а версія в чорному кольорі особливо популярна серед користувачів, які прагнуть поєднати функціональність і стильний вигляд інтер'єру. Як і Nuki, він може бути додатково укомплектований аксесуарами – зокрема, розумною клавіатурою для безключового доступу без смартфона, що збільшує гнучкість використання.

Рисунок 1.4 – Електронний контролер YALE LINUS чорний до циліндру [4]

Проте можна відмітити загальні недоліки для всіх таких типових пристроїв – вони, як правило, є закритими пропрієтарними рішеннями з обмеженим доступом до внутрішніх алгоритмів, високою вартістю, залежністю від

стабільного інтернет-з’єднання та центрального хмарного сервера. Деякі з них можуть не підтримувати кастомізовану логіку чи розширення на апаратному рівні, що критично в умовах побудови гнучких кіберфізичних систем. Ще одним недоліком є ціна таких рішень.

На противагу цьому існує велика кількість відкритих рішень на основі мікроконтролерів, зокрема Arduino, ESP8266, ESP32, Raspberry Pi. Одним із найбільш популярних підходів є створення індивідуальних систем із використанням ESP32. Цей мікроконтролер є потужним, недорогим, має вбудований Wi-Fi та Bluetooth, підтримує численні бібліотеки, а також забезпечує гнучкість у побудові логіки автентифікації, авторизації, зв’язку з хмарою або локальними серверами.

До найвідоміших платформ з відкритим кодом, які підтримують подібні реалізації, належать:

– Home Assistant – масштабована платформа з інтеграцією понад тисячі пристроїв, що дозволяє створювати автоматизації, сповіщення, сценарії доступу;

– OpenHAB – ще одне потужне середовище для автоматизації будинку з відкритим кодом, що підтримує інтеграцію з системами контролю доступу, сенсорами, камерами;

– ESPHome – спеціалізований фреймворк для мікроконтролерів ESP, що дозволяє швидко будувати власні прошивки для пристроїв із YAML-конфігурацією;

– Blynk IoT — хмарна платформа з візуальним інтерфейсом для створення інтерфейсів керування на мобільному пристрої для пристроїв на ESP32.

У порівнянні функціональних можливостей можна виділити кілька критеріїв: гнучкість, вартість, складність розгортання, можливість локального керування, залежність від хмари, розширюваність. Комерційні рішення зазвичай пропонують простоту встановлення та стабільну роботу з високим рівнем користувацького сервісу, але вони малопридатні до глибокої кастомізації. Натомість рішення на ESP32, як у розробленому пристрої, забезпечують повний контроль над логікою доступу, гнучке налаштування інтерфейсів (наприклад,

через клавіатуру 4×4, локальний LCD-дисплей, бузер), можливість вбудованого сповіщення на смартфон або через телеграм-бота без використання дорогих платформ.

Таким чином, у межах даного проєкту використання ESP32 як ядра програмно-апаратного засобу дає змогу реалізувати індивідуальну систему контролю доступу з функцією сповіщення, яка за своїми можливостями не поступається комерційним аналогам, але при цьому залишається повністю відкритою, розширюваною і адаптованою під конкретні потреби розумного будинку.

1.4 Протоколи та технології комунікації у системах керування доступом

У системах «Розумного будинку», які включають програмно-апаратні засоби керування доступом, що базуються на мікроконтролерах ESP32, особливу роль відіграє вибір протоколу комунікації. Такий протокол повинен бути не лише легким і енергоефективним, але й достатньо функціональним для забезпечення надійного обміну даними між пристроями в локальній мережі та хмарними сервісами. Серед найпоширеніших протоколів, які застосовуються у системах IoT, можна виділити MQTT (Message Queuing Telemetry Transport), HTTP (Hypertext Transfer Protocol), CoAP (Constrained Application Protocol), а також WebSocket, AMQP і LoRaWAN (для спеціалізованих сценаріїв).

Один із найпопулярніших протоколів у сфері IoT є протокол MQTT. Він був розроблений для роботи в умовах обмежених ресурсів і нестабільного з’єднання, що робить його ідеальним для пристроїв на зразок ESP32. MQTT працює за моделлю "публікація–підписка" (publish/subscribe) з використанням брокера – центрального вузла, через який всі клієнти обмінюються повідомленнями. Завдяки цьому, пристрій може, наприклад, публікувати повідомлення про подію (наприклад, "двері відкрито", "неправильний пароль") на певну тему, а інші системи – як-от інтерфейс Node-RED або хмарна аналітика

– можуть підписатися на цю тему й отримувати ці повідомлення в реальному часі.

Можна відзначити наступні переваги даного протоколу:

– дуже малий розмір пакету (мінімум 2 байти заголовка);

– низьке енергоспоживання;

– підтримка QoS (якість обслуговування);

– можливість роботи через TLS/SSL;

– асинхронна комунікація (без очікування відповіді).

Це дозволяє зручно реалізувати функції сповіщення, моніторингу та керування доступом, зокрема в "розумному" замку, що керується через ESP32.

Основна концепція MQTT базується на архітектурі «видавець-підписник» (publish-subscribe), де всі повідомлення проходять через центральний вузол – MQTT-брокер (рис. 1.5). Клієнти (наприклад, ESP32, смартфон, хмарний сервер чи вебінтерфейс) можуть або публікувати повідомлення на певну тему (topic), або підписуватися на тему, щоб отримувати повідомлення, які в майбутньому надсилатимуться іншими клієнтами. Такий підхід забезпечує гнучкість та ізоляцію між відправником і отримувачем даних, що дуже важливо для побудови модульних і розширюваних IoT-систем.

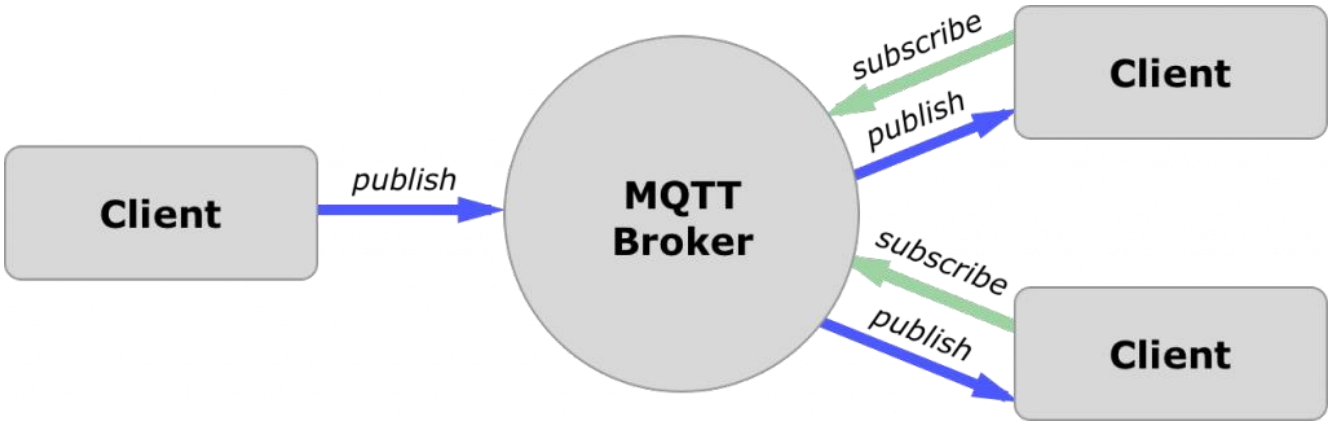


Рисунок 1.5 – Організації взаємодії через протокол MQTT

У системі керування доступом MQTT може використовуватись для вирішення таких завдань:

– передачі команд відкриття або блокування дверей;

– надсилання сповіщень про події (наприклад, вхід дозволено, пароль неправильний, спроба злому);

– ведення журналу доступу у форматі JSON, що передається на хмарну платформу чи сервер;

– організації зв’язку з мобільним застосунком або браузером у реальному часі.

Також MQTT підтримує три рівні якості доставки повідомлень:

– at most once (не більше одного разу) – без підтвердження доставки;

– at least once (принаймні один раз) – з підтвердженням, але з можливим дублюванням;

– exactly once (рівно один раз) – найнадійніший варіант, але ресурсоємніший.

Це дозволяє адаптувати протокол до потреб конкретної системи, балансуючи між швидкістю і надійністю.

Ще одним можливим протоколом, що може бути використаний у системах доступупу є протокол HTTP. Цей протокол є класичним протокол інтернету, який базується на моделі запит–відповідь. Він часто використовується у веб-додатках і API, однак у контексті пристроїв з обмеженими ресурсами, таких як ESP32, має обмеження. По-перше, HTTP-повідомлення мають відносно великий розмір, що створює навантаження на мережу та енергоспоживання. По-друге, він не є оптимальним для подій, які мають бути надіслані з пристрою без запиту – наприклад, тривожні сигнали.

Проте разом із тим протокол HTTP залишається корисним для конфігурації пристрою через REST API, віддаленого доступу через веб-браузер або інтеграції із зовнішніми сервісами.

Іншим протоколом є CoAP, що був розроблений спеціально для IoT. Він заснований на UDP і має модель, подібну до HTTP, але з набагато меншою надбудовою. CoAP підтримує методи GET, POST, PUT, DELETE і має вбудовану підтримку сповіщень через "observe"-механізм, що робить його придатним для

сенсорних мереж. Однією з особливостей CoAP є можливість роботи без постійного з’єднання, що знижує енергоспоживання.

У порівнянні з MQTT, CoAP краще працює у вузькоспеціалізованих сенсорних мережах, але гірше масштабується для інтеграції з хмарними брокерами. На практиці CoAP часто використовується в індустріальних або обмежених сценаріях, де немає потреби у централізованому брокері.

Ще одним рішенням є WebSocket (рис. 1.6). WebSocket – це протокол двостороннього зв’язку поверх TCP, який забезпечує постійне з’єднання між клієнтом і сервером. На відміну від HTTP, який закриває з’єднання після кожного запиту, WebSocket дає змогу обом сторонам обмінюватися повідомленнями у реальному часі. Ця особливість є досить зручною для інтерфейсів на кшталт панелі керування в браузері або в Node-RED, однак менш оптимальне для мікроконтролерів через потребу постійно підтримувати з’єднання.

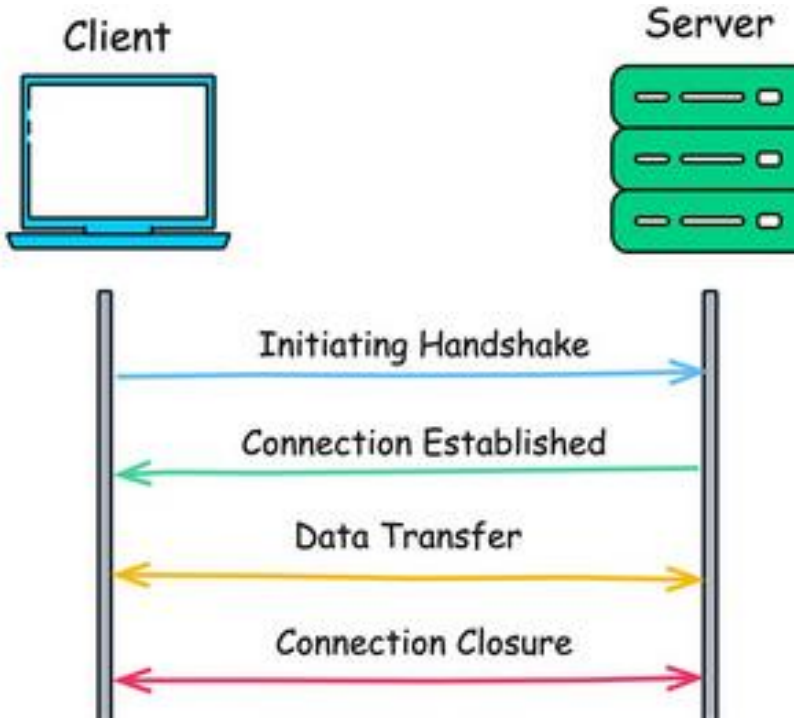


Рисунок 1.6 – Організації взаємодії через протокол WebSocket

Також досить часто використовують ще один протокол – AMQP (рис. 1.7). Advanced Message Queuing Protocol – це складніший, повнофункціональний

протокол обміну повідомленнями, який використовується переважно в корпоративних системах. Для ESP32 він є надто "важким" і рідко використовується безпосередньо на рівні пристрою. Проте AMQP іноді використовується у хмарній частині IoT-систем, де здійснюється подальша маршрутизація повідомлень від брокерів MQTT.

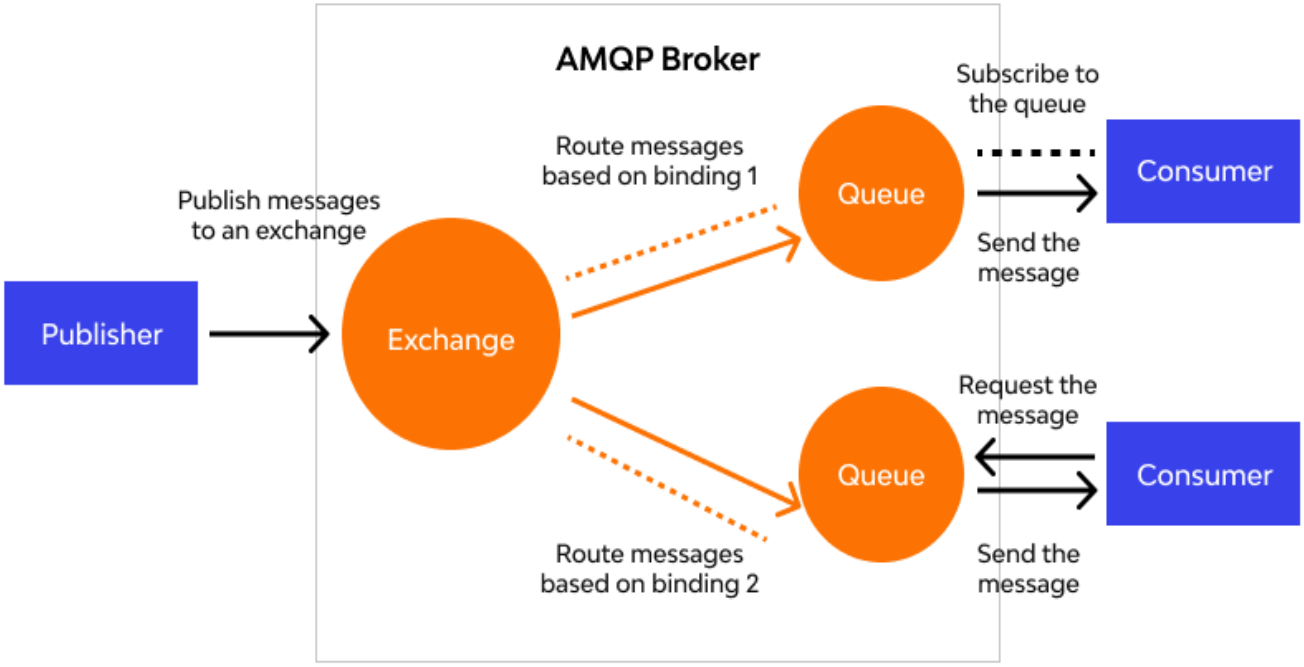


Рисунок 1.7 – Організації взаємодії через протокол AMQP

1.5 Висновки. Постановка задачі

Однією з ключових причин розробки власного програмно-апаратного засобу керування доступом до кіберфізичної системи «Розумний будинок» із функцією сповіщення на основі ESP32 є потреба у створенні персоналізованого, економічно вигідного та технічно гнучкого рішення. Комерційні продукти часто є дорогими, обмеженими у функціоналі або складними для інтеграції в індивідуальні проєкти. Тому використання мікроконтролера ESP32 дозволить реалізувати систему з підтримкою бездротового зв'язку, локальної автентифікації (через клавіатуру) та миттєвим сповіщенням користувача про події доступу, що є критично важливим для сучасних систем безпеки у розумних будинках.

Щоб розв’язати поставлене завдання, необхідно виконати наступні ключові етапи:

1) дослідити сучасні програмно-апаратні рішення, що реалізують доступ до системи Розумного будинку; проаналізувати відомі апаратні та хмарні платформи; виконати аналіз традиційних та сучасних технологій віддаленого контроолю, визначити їх переваги та недоліки;

2) виокремити вимоги та спроєктувати структуру програмно-апартаного засобу керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32; виконати аналіз та вибір компонентів, протоколів та механізмів комунікаццї;

3) виконати проєктування схеми електричної для пропонованого пристрою;

4) здійснити тестування спроєктованого пристрою;

5) оцінити ефективність роботи спроєктованого пристрою;

6) оцінити вартість компонентів програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32.

2 ПРОЕКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ КЕРУВАННЯ ДОСТУПОМ ДО КІБЕРФІЗИЧНОЇ СИСТЕМИ "РОЗУМНИЙ БУДИНОК" ІЗ ФУНКЦІЄЮ СПОВІЩЕННЯ НА ОСНОВІ ESP32

2.1 Визначення вимог до програмно-апаратного засобу

Під час проєктування програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32 було проаналізовано та виявлено низку функціональних та нефункціональних вимог, які необхідно було врахувати для забезпечення належної роботи системи, її надійності, безпеки, гнучкості та масштабованості.

Передусім, однією з основних вимог було забезпечення механізму віддаленого керування доступом, що дозволяє розблоковувати двері за командою з іншого пристрою через мережу. В якості протоколу комунікації між складовими проєктованого пристрою планується використати протокол MQTT. Цей протокол було обрано через його легкість, ефективність і оптимальність для передачі повідомлень у мережах з обмеженими ресурсами, таких як ESP32. Для цього система має бути здатна підключатися до інтернету за допомогою вбудованого Wi-Fi модуля ESP32, встановлювати захищене з'єднання з MQTT-брокером HiveMQ через протокол TLS/SSL та обробляти вхідні MQTT-повідомлення. Це вимагатиме реалізації захищеної автентифікації MQTT-клієнта із зазначенням імені користувача, пароля та використанням порту 8883, який є стандартним для шифрованих з’єднань.

Наступною критичною вимогою стало локальне керування доступом за допомогою клавіатури. Система повинна була ідентифікувати користувача на основі введеного пароля, а також відображати інструкції, статуси й повідомлення на LCD-дисплеї, зручному для сприйняття користувачем. Важливою умовою була наявність функції зміни пароля з самої клавіатури, з підтвердженням нового пароля, що підвищує безпеку та адаптивність системи під різні сценарії використання.

Також система повинна була вести журнал подій доступу: фіксувати кожну спробу входу, незалежно від того, була вона успішною чи ні, й публікувати відповідну інформацію на MQTT-тему у вигляді JSON-структури. Це дозволяє адміністраторам або власникам будинку в реальному часі переглядати історію взаємодії з системою, здійснювати моніторинг та аудит дій користувачів.

Особливу увагу також слід приділити обробці некоректних спроб входу. З метою захисту від підбору пароля, система повинна була реагувати на надмірну кількість хибних спроб (понад три) відповідним повідомленням MQTT на окрему тему, що може бути використано для миттєвого сповіщення через інші сервіси або активації захисних сценаріїв, таких як блокування доступу або ввімкнення сигналізації.

2.2 Проєктування структури програмно-апаратного засобу

Після визначення вимог було спроєктовано структуру програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32, схему якої подано на рис. 2.1.

Проєктована система Система складається з трьох основних компонентів:

1. клієнт пристрою (ESP32);
2. хмарний MQTT-брокер HiveMQ;
3. клієнт керування (Node-RED).

Клієнт пристрою у проєктованій системі реалізований на базі мікроконтролера ESP32, який виконує основні функції локального керування доступом до "Розумного будинку". Основне його призначення – забезпечення автентифікації користувача шляхом введення паролю через клавіатуру.

У випадку правильного введення паролю ESP32 ініціює фізичне розблокування дверей, тоді як при помилковому введенні фіксує спробу доступу та публікує відповідне повідомлення у хмарний MQTT-брокер для подальшого оброблення.

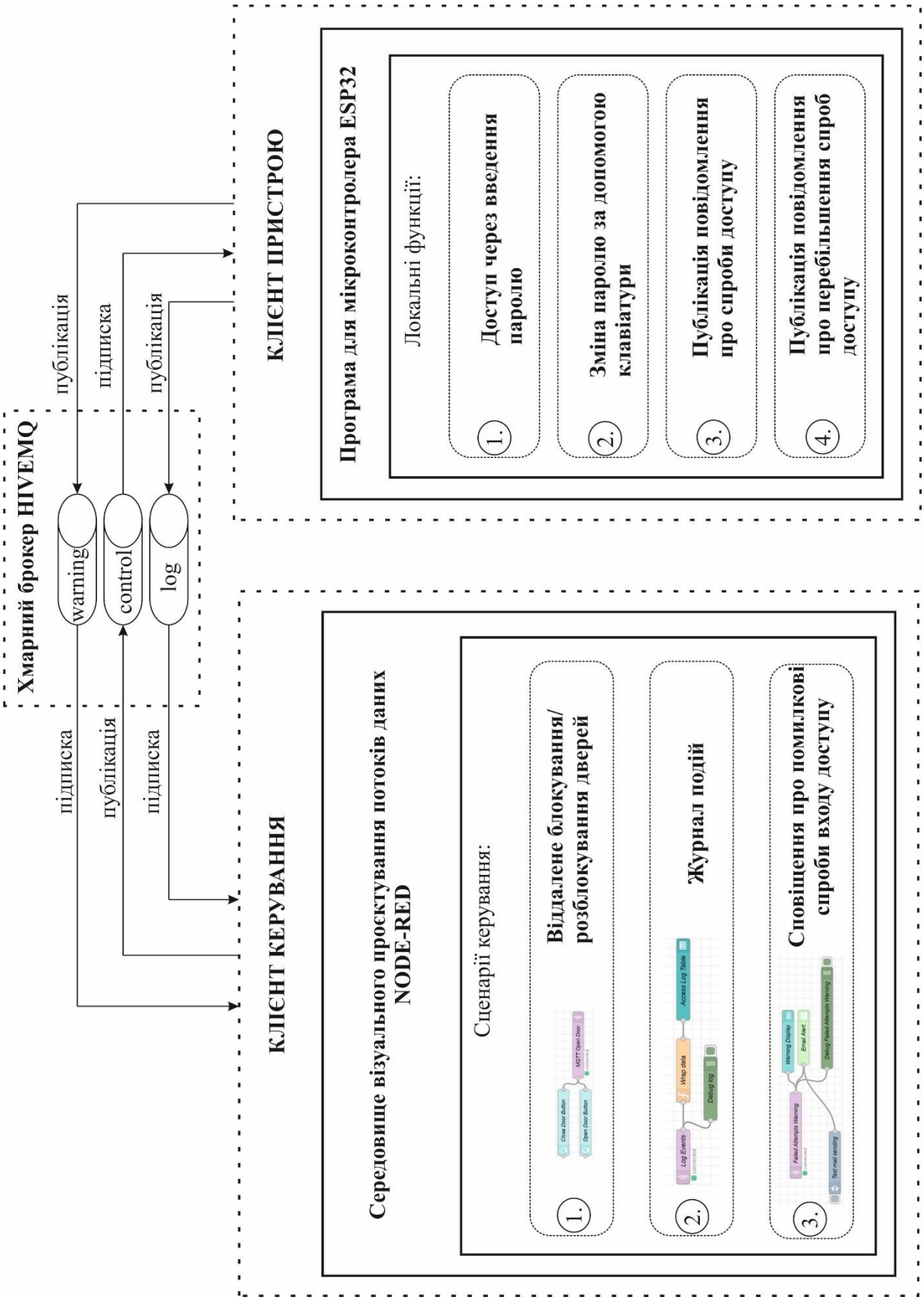


Рисунок 2.1 – Структруа програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32

Крім цього, клієнт пристрою підтримує зміну паролю без потреби зовнішнього керування – користувач може ввести новий пароль безпосередньо через клавіатуру, що забезпечує автономність пристрою у разі зміни умов безпеки. Для підвищення рівня контролю ESP32 також відстежує кількість невдалих спроб доступу, і у випадку перевищення встановленого порогу автоматично генерує попередження, яке передається через MQTT у вигляді повідомлення у відповідну тему.

Таким чином, клієнт пристрою виконує роль автономного захисного елемента, здатного локально здійснювати авторизацію, вести облік спроб входу та взаємодіяти з мережею для передачі інформації про події та інциденти, забезпечуючи інтеграцію з іншими елементами системи безпеки.

Іншою складовою є клієнт керування. Клієнт керування, реалізований у середовищі Node-RED, відповідає за централізоване управління доступом до кіберфізичної системи "Розумний будинок" та оброблення інформації, що надходить від пристрою на базі ESP32 через хмарний MQTT-брокер HiveMQ. Node-RED виконує роль візуального інтерфейсу, за допомогою якого адміністратор або користувач може віддалено контролювати стан доступу – зокрема, блокувати або розблоковувати двері, надсилаючи відповідні MQTT-команди.

Крім цього, клієнт керування забезпечує фіксацію усіх подій, пов’язаних зі спробами доступу, у вигляді журналу подій. Це дозволяє відстежувати активність у системі, аналізувати потенційні загрози чи спроби несанкціонованого доступу. У випадку перевищення кількості помилкових введень паролю або підозрілої активності, Node-RED приймає повідомлення з ESP32 і ініціює формування відповідного сповіщення, що може бути передане користувачеві у реальному часі через зручні канали, наприклад, push-повідомлення, електронну пошту або месенджер.

Таким чином, клієнт керування у вигляді Node-RED не лише виконує функції інтерфейсу керування та моніторингу, а й виступає ключовою ланкою, яка поєднує фізичну частину системи з хмарною інфраструктурою та забезпечує автоматизовану реакцію на події у режимі реального часу.

Хмарний MQTT-брокер HiveMQ у цій системі виступає посередником між клієнтом пристрою (ESP32) та клієнтом керування (Node-RED), забезпечуючи надійний обмін повідомленнями за протоколом MQTT. Його основна роль полягає в організації передачі даних між фізичним рівнем системи – де виконується взаємодія з користувачем через пристрій – і логічним рівнем, де відбувається обробка інформації, ухвалення рішень та реагування на події.

Брокер HiveMQ обробляє публікацію та підписку на повідомлення в межах трьох основних топіків: control, log і warning. Через топік control передаються команди на пристрій, наприклад, для віддаленого відкриття чи закриття дверей. Топік log використовується для передачі даних про події, які відбуваються на стороні пристрою, зокрема про успішні та невдалі спроби введення паролю. Нарешті, топік warning слугує для пересилання повідомлень про надзвичайні ситуації, зокрема при перевищенні ліміту невдалих спроб доступу.

Таким чином, HiveMQ є ключовим елементом, що забезпечує масштабовану, асинхронну й безперервну комунікацію між усіма компонентами системи. Завдяки використанню хмарного брокера можливо реалізувати не лише локальний, а й віддалений контроль доступу, зокрема з будь-якої точки, де є підключення до Інтернету. Це суттєво розширює функціональні можливості системи та підвищує її адаптивність до майбутніх вимог користувачів.

2.3 Схема елекрична клієнта пристрою

Була спроєктовано схему елекричну для клієнта пристрою. Схема подана на рис. 2.2. Дана електрична схема зображує клієнтський пристрій для програмно-апаратного засобу керування доступом до кіберфізичної системи «Розумний будинок», що побудований на основі мікроконтролера ESP32. У схемі реалізовано інтерфейс введення з клавіатури, виведення інформації на LCD-дисплей, живлення від 5В із стабілізацією напруги до 3.3В, звукове сповіщення через бузер, а також взаємодія з периферією.

Основу схеми становить мікроконтролер ESP32, позначений як U3. Саме він виконує обробку введених з клавіатури даних, взаємодіє з дисплеєм і керує іншими елементами. ESP32 живиться напругою 3.3В, яку забезпечує стабілізатор напруги LM1117 (позначений як U1). Вхідна напруга 5В подається через джерело живлення, а на виході стабілізатора маємо 3.3В, що подається на відповідні виводи ESP32 (3V3 або VCC). Конденсатори C1 (10 µF) і C2 (47 µF) підключені на вході та виході стабілізатора для фільтрації пульсацій та забезпечення стабільної роботи регулятора напруги.

До ESP32 підключено 4x4 матричну клавіатуру (U2), яка підключена через 8 цифрових входів/виходів (GPIO). У схемі видно, що 4 лінії рядків клавіатури підключені до виводів IO12–IO15, а 4 лінії стовпців – до IO16–IO19. Така реалізація дозволяє сканувати натиснуту клавішу за допомогою програмного опитування – ESP32 по черзі активує рядки й читає сигнали з колонок, виявляючи замкнуту комбінацію.

Результат введення з клавіатури відображається на LCD-дисплеї типу 1602 (позначено як U4), що має 16 символів у 2 рядки. Цей дисплей працює за протоколом паралельного інтерфейсу й підключений до ESP32 через 6 виводів (RS, E, D4–D7). У схемі видно, що виводи дисплея з'єднані з ESP32 через GPIO IO21–IO26. Крім того, дисплей має живлення 5В і заземлення, а також регулювання контрасту через V0.

Ще один важливий елемент схеми це бузер BUZZER1, який підключено до одного з цифрових виводів ESP32 (через вивід IO23), і дозволяє генерувати звукові сигнали, наприклад, для підтвердження правильного введення пароля або попередження про помилку. Бузер підключено між GPIO та землею через відповідну обмотку (зазвичай використовується активний бузер, який спрацьовує при подачі логічної «1» на вхід).

Таким чином, уся схема розроблена так, щоб забезпечити повноцінну взаємодію користувача з мікроконтролером через клавіатуру та LCD, а також забезпечити звукове сповіщення через бузер. Стабілізація напруги необхідна для живлення ESP32, який працює при 3.3В, хоча джерело живлення є 5-вольтовим. Усі

підключення відповідають логіці керування, де ESP32 є центральним вузлом обробки, а інші компоненти виконують роль периферії для взаємодії з користувачем і подання зворотного зв’язку.

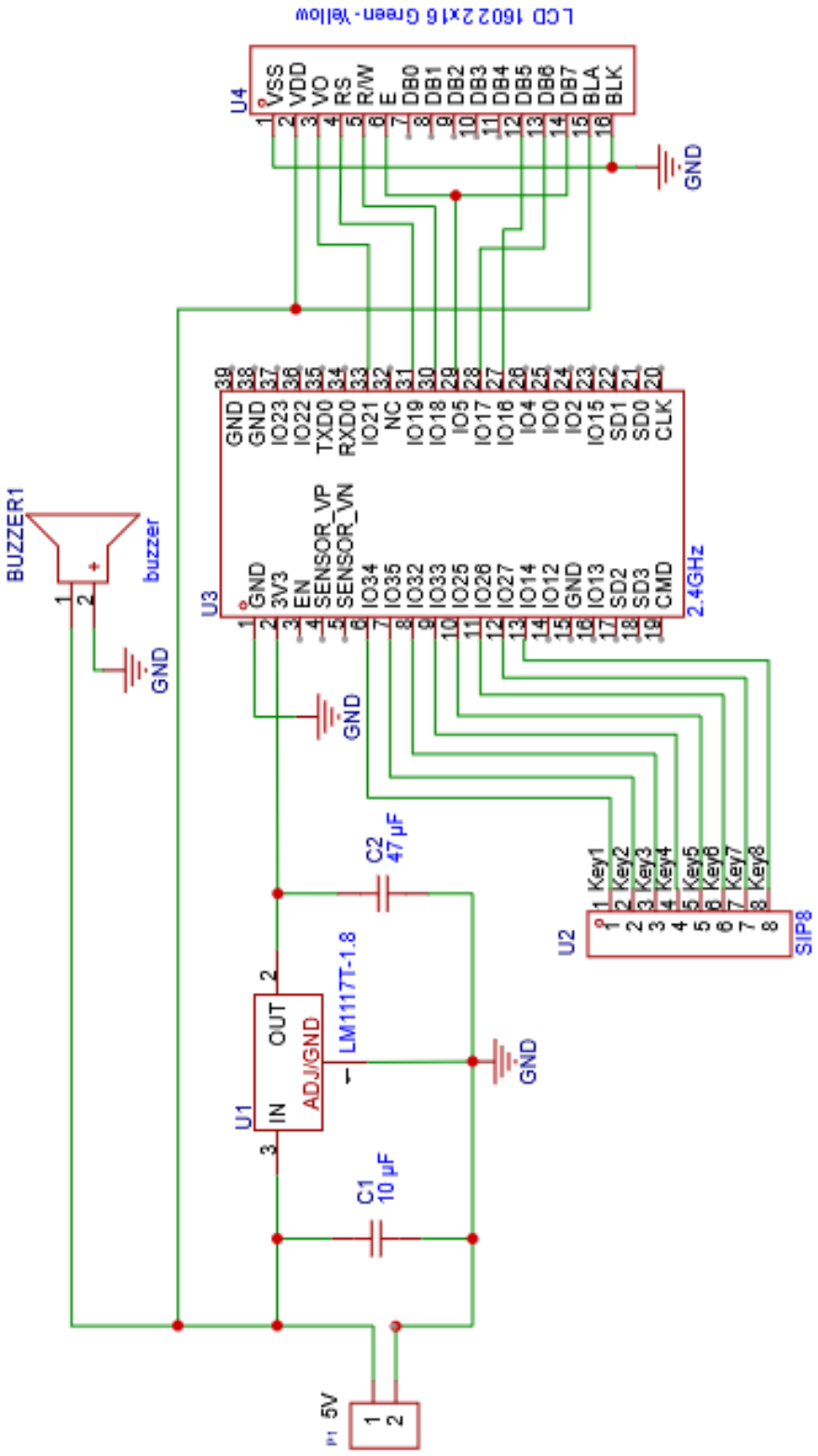


Рисунок 2.2 – Схема елекрична клієнта пристрою

2.4 Вибір апаратних компонентів для клієнта пристрою

Відповідно до спроєктовано структури програмно-апаратного засобу керування доступом до кіберфізичної системи «Розумний будинок» із функцією сповіщення на основі ESP32 і схеми електричної для клієнта пристрою наступний перелік електронних компонентів і модулів:

– мікроконтролер;

– матрична клавіатуру;

– lcd дисплей;

– бузер;

– стабілізатор напруги;

– додаткові компоненти (електролітичні конденсатори, резистори, тощо).

У процесі розробки клієнта пристрою для системи контролю доступу було обрано мікроконтролер ESP32 (рис. 2.3), оскільки він поєднує високу обчислювальну потужність, підтримку бездротових комунікацій і багатий набір периферійних інтерфейсів. Цей мікроконтролер є логічним вибором для IoT-систем, які потребують одночасної обробки даних з кількох джерел, взаємодії з хмарними сервісами та управління виконавчими пристроями.

Рисунок 2.3 – Мікроконтролер ESP32

Однією з ключових переваг ESP32 є наявність вбудованого модуля Wi-Fi, що дозволяє встановлювати з’єднання з бездротовими мережами без додаткового обладнання. Це особливо важливо для даної системи, адже мікроконтролер повинен підключатися до MQTT-брокера через інтернет, використовуючи захищене з'єднання на основі SSL/TLS. ESP32 має апаратну підтримку криптографічних алгоритмів, що дозволяє ефективно обробляти зашифровані з'єднання без суттєвого навантаження на процесор.

Мікроконтролер оснащений двоядерним процесором Tensilica Xtensa, що працює на частоті до 240 МГц. Це дає можливість реалізовувати паралельну обробку кількох задач — наприклад, зчитування натискань клавіш, оновлення дисплея, обробки зворотного зв’язку з MQTT-брокера та керування сервоприводом – без втрати швидкодії. До того ж, ESP32 підтримує FreeRTOS, що відкриває можливість масштабування проєкту у майбутньому з використанням багатозадачності.

З погляду периферії, ESP32 має велику кількість GPIO-виводів, що дозволяє під’єднати одночасно кілька пристроїв: матричну клавіатуру, I2C LCD дисплей, сервомотор, а також залишити резерв для можливого підключення інших сенсорів чи модулів. Підтримка інтерфейсу I2C значно спрощує підключення дисплея, дозволяючи заощадити виводи мікроконтролера.

Ще одним важливим фактором є енергоефективність ESP32 – у режимі очікування він може споживати мінімум енергії, що є актуальним у випадку побудови автономних або мобільних систем з живленням від акумулятора.

Ще одним компонентом для даного пристрою є матрична клавіатура. Для даного пристрою було обрано матричну клавіатуру 4×4, що обґрунтовується як оптимальне рішення для введення даних користувачем, зокрема PIN-коду або пароля доступу (рис. 2.4).

Ця клавіатура складається з 16 клавіш, організованих у вигляді матриці 4 рядки × 4 стовпці, що дозволяє значно зекономити кількість GPIO-пінів ESP32. На схемі клавіатура (позначена як U2, SP8) підключена безпосередньо до восьми цифрових входів/виходів мікроконтролера ESP32. Саме таке підключення є

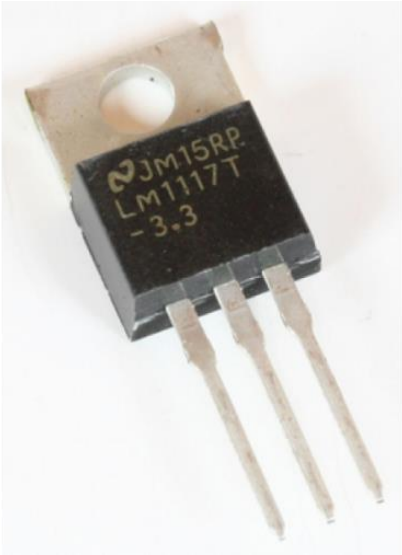
стандартним у випадках, коли використовується прямий скан клавіатури без додаткових контролерів або мультиплексорів.

Рисунок 2.4 – Матрична клавіатура 4x4

З технічної точки зору, вибір матричної клавіатури обумовлений її простотою реалізації, низькою вартістю, компактністю та надійністю. Вона дозволяє реалізувати інтерфейс користувача без потреби у сенсорному екрані чи інших складних елементах. При цьому, як видно зі схеми, кожна з восьми ліній клавіатури підключена до окремого GPIO-піну ESP32, що забезпечує можливість програмного сканування: мікроконтролер по черзі активує кожен рядок (подає логічний LOW), а на стовпцях перевіряє наявність натискання (логічний рівень змінюється на LOW, якщо кнопка натиснута).

У системі керування доступом саме така клавіатура дає змогу вводити PIN-коди або спеціальні команди. Це підвищує рівень безпеки та дозволяє ідентифікувати користувача локально, без потреби в зовнішньому зв’язку. Крім того, проста реалізація коду сканування клавіш для ESP32 забезпечує високу швидкість зчитування та чутливість до натискань, що критично важливо в системах, де користувач очікує миттєвого зворотного зв’язку.

У контексті вибору апаратних засобів для клієнтського пристрою системи керування доступом до кіберфізичної системи «Розумний будинок», використання стабілізатора напруги LM1117-3.3 є ключовим елементом живлення, що забезпечує надійну та стабільну роботу мікроконтролера ESP32 та інших компонентів, чутливих до перенапруги (рис. 2.5).



Рисунко 2.5 – Стабілізатор напруги LM1117-3.3

На електричній схемі видно, що живлення подається від джерела 5 В. Проте мікроконтролер ESP32 та підключені до нього компоненти, зокрема дисплей, клавіатура й інші периферійні пристрої, працюють при напрузі 3.3 В, тому виникає необхідність у пониженні напруги з 5 В до 3.3 В. Саме цю функцію виконує лінійний стабілізатор LM1117-3.3, розташований у схемі під позначенням U1.

LM1117-3.3 – це фіксований стабілізатор напруги, що на виході завжди підтримує стабільні 3.3 В (на відміну від регульованої версії, яка потребує додаткових резисторів). У схемі також використано два конденсатори: C1 (10 µF) на вході та C2 (47 µF) на виході, які згідно з технічними рекомендаціями необхідні для забезпечення стійкості стабілізатора та пригнічення високочастотних шумів. Ці електролітичні конденсатори фільтрують напругу, запобігаючи стрибкам та осциляціям, особливо при миттєвій зміні навантаження (наприклад, коли ESP32 активує Wi-Fi або виводить зображення на дисплей).

Таким чином, стабілізатор LM1117-3.3 гарантує, що навіть при незначних коливаннях на вході (наприклад, 5.1 В або 4.9 В), на виході зберігатиметься стабільна напруга 3.3 В. Це критично важливо для ESP32, оскільки перевищення допустимого рівня напруги може вивести його з ладу. Крім того, такий стабілізатор легко інтегрується в проєкт: він має стандартне трививідне підключення (вхід, маса, вихід), займає мінімум місця на платі та не потребує складного налаштування.

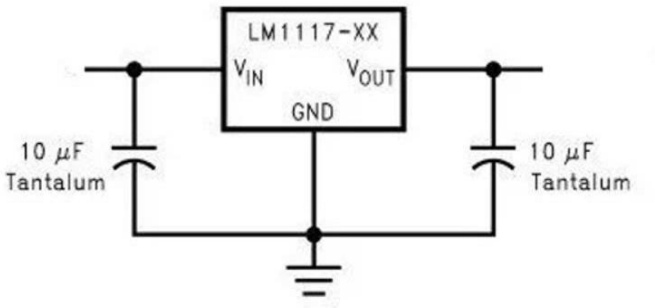


Рисунок 2.6 – Схема включення стабілізатора LM1117-3.3

У складі клієнтського пристрою програмно-апаратного засобу керування доступом до кіберфізичної системи «Розумний будинок» із функцією сповіщення важливу роль виконує LCD-дисплей типу 1602, який слугує засобом візуального виведення інформації користувачеві (рис. 2.7). На електричній схемі він позначений як U4 (LCD1602), і, на відміну від варіантів з I2C-модулем, реалізований через паралельне підключення, що потребує використання кількох GPIO-пінів ESP32.

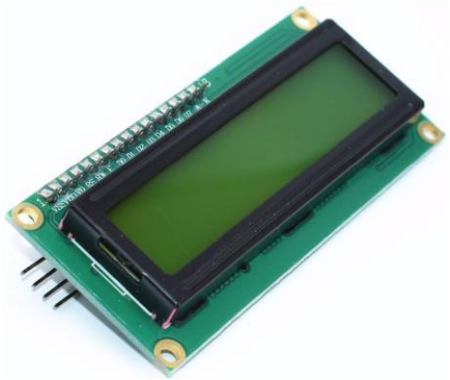


Рисунок 2.1 – LCD 1602 дислпей

Зокрема, дисплей з’єднаний із мікроконтролером через 7 сигнальних ліній: чотири лінії передачі даних (D4–D7), а також керувальні сигнали RS (Register Select), RW (Read/Write) і E (Enable). Це типове 4-бітне паралельне з'єднання, яке дозволяє передавати інформацію порціями по 4 біти, зменшуючи кількість необхідних ліній даних у порівнянні з повним 8-бітним режимом, але все ж таки вимагає значно більше пінів, ніж I2C-варіант.

Живлення для дисплея подається від стабілізованого джерела 5 В, при цьому важливо, що шини даних і керування з’єднані безпосередньо з ESP32, який працює на рівні логіки 3.3 В. Така конфігурація є допустимою, оскільки більшість LCD1602 підтримують роботу з TTL-рівнями, однак у випадку потреби додаткової стабільності можуть бути використані підтягувачі або буфери рівнів.

Паралельне підключення потребує трохи більше зусиль під час програмування, адже потрібно правильно ініціалізувати порти ESP32, однак з технічного боку воно забезпечує більшу швидкість оновлення дисплея, що може бути корисно в реальному часі – наприклад, при введенні пароля або індикації статусу доступу. Також це дозволяє не використовувати додаткові модулі I2C, спрощуючи схему на рівні апаратного проєктування.

Функціонально дисплей відображає повідомлення, пов’язані з роботою системи:

– введення PIN-коду;

– підтвердження авторизації;

– помилки доступу або інструкції для користувача.

Таким чином, він відіграє ключову роль у забезпеченні локального зворотного зв’язку.

У складі клієнтського пристрою програмно-апаратного засобу керування доступом до кіберфізичної системи «Розумний будинок» із функцією сповіщення передбачено використання п’єзобузера, який виконує роль звукового індикатора подій. Його призначення – забезпечити оперативний зворотний зв’язок користувачу у вигляді звукових сигналів під час взаємодії з пристроєм.

Рисунок 2.8 – П’єзобузер

Бузер підключено безпосередньо до одного з GPIO-пінів мікроконтролера ESP32. Інший контакт бузера з’єднано з загальним проводом (GND), отже, коли відповідний GPIO встановлюється в логічний високий рівень, через бузер протікає струм і генерується звук. Такий спосіб підключення характерний для активних п’єзобузерів, які мають вбудований генератор і починають звучати просто при подачі напруги. Це значно спрощує апаратну частину, оскільки не вимагає зовнішнього керування частотою чи модулювання сигналу.

Відсутність проміжного транзистора у електричній схемі означає, що живлення бузера та струм, необхідний для його роботи, забезпечується безпосередньо з GPIO ESP32. Це допустимо у випадку використання малопотужного активного бузера, який споживає дуже невеликий струм (зазвичай до 20 мА), тобто в межах, які здатен витримати GPIO-пін мікроконтролера без шкоди для його працездатності.

Функціонально бузер використовується для подачі коротких сигналів при кожному натисканні клавіш на матричній клавіатурі, довшого звукового сповіщення про успішне введення коду або, навпаки, попереджувального сигналу у випадку невірного коду чи спроби несанкціонованого доступу. Завдяки цьому навіть без використання дисплея користувач може зрозуміти, що відбувається в системі, орієнтуючись лише на звук.

2.5 Визначення вартості компонентів клієнта пристрою

Визначення вартості компонентів клієнта пристрою полягає в підрахунку загальної ціни усіх апаратних елементів, що входять до складу схеми. До переліку входять: мікроконтролер ESP32, матрична клавіатура 4×4, LCD-дисплей (без інтерфейсу I2C), стабілізатор напруги LM1117-3.3, п’єзобузер, два електролітичні конденсатори, а також з’єднувальні дроти, плата та допоміжні елементи. Вартість компонентів клієнта пристрою представлено у таблиці 2.1.

Таким чином проаналізувавши вартість обраних апаратних компонентів, було встановлено, що загальна вартість програмно-апаратного пристрою не перевищує 400 грн, що з урахуванням виконання покладених функцій, є оптимальним рішенням.

Таблиця 2.1 – Вартість компонентів клієнта пристрою

№	Назва компоненту	Вартість, грн
1	Мікроконтролер ESP32 ESP-WROOM-32 ESP32 30 Pin (Wi-Fi + Bluetooth)	230,00
2	LCD диспей 1602	55,00
3	П’єзодинамік	20,00
4	Стабілізатор напруги LM1117-3.3	50,00
5	Матрична клавіатура 4x4	25,00
6	Конденсатори, резистори	20,00
Разом 400,00 грн		

2.6 Висновки

У цьому розділі були розглянуті питання, пов’язані із визначенням вимог до програмно-апаратного засобу та проєктуванням його структури. Зокрема було визначено такі вимоги як реалізація віддаленого керування доступом через MQTT,

локальна автентифікацію з клавіатури та зміна пароля, а також ведення журналу подій і опрацювання помилкових спроб входу з генерацією сповіщень.

Описано структуру програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" на основі ESP32, включно з трьома основними компонентами системи: клієнтом пристрою, хмарним MQTT-брокером HiveMQ та клієнтом керування Node-RED. Було проаналізовано функціональні можливості ESP32 як локального елемента авторизації, який забезпечує автентифікацію користувачів, обробку спроб доступу та автономну зміну пароля. Крім того, розглянуто роль клієнта керування у Node-RED як інтерфейсу для централізованого моніторингу, управління доступом і ведення журналу подій. Особлива увага приділялася функціям сповіщення про події та надзвичайні ситуації, що реалізуються через MQTT-топіки broker'а HiveMQ, який виступає надійним посередником для обміну повідомленнями між усіма компонентами системи. Також проаналізовано апаратні компонети та оцінено їх вартість. Було визначено, що загальна вартість програмно-апаратного пристрою не перевищує 400 грн, що з урахуванням виконання покладених функцій, є оптимальним рішенням.

3 ТЕСТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ КЕРУВАННЯ ДОСТУПОМ ДО КІБЕРФІЗИЧНОЇ СИСТЕМИ "РОЗУМНИЙ БУДИНОК" ІЗ ФУНКЦІЄЮ СПОВІЩЕННЯ НА ОСНОВІ ESP32

3.1 Створення хмарного брокера HiveMQ для організації обміну між клієнтами

Відповідно до поставлених завдань програмно-апаратний засіб має реалізовувати функції дистанційного керування доступом до приміщення, зберігання журналу подій та миттєвого сповіщення про несанкціоновані спроби входу. Для виконання цих функцій необхідна постійна взаємодія між програмно-апаратним засобом на базі ESP32 та середовищем візуального програмування Node-RED, яке забезпечує інтерфейс користувача й логіку обробки подій. Щоб організувати цю взаємодію незалежно від фізичного розташування пристроїв і забезпечити стабільну передачу даних у реальному часі, необхідно впровадити хмарний механізм обміну повідомленнями.

Одним із найбільш зручних та ефективних варіантів для цього є використання протоколу MQTT, який дозволяє організувати легкий і надійний обмін інформацією між клієнтами системи. У такій архітектурі як ESP32, так і Node-RED виступають MQTT-клієнтами, які обмінюються повідомленнями через центральний сервер – MQTT-брокер. Щоб забезпечити таку можливість без додаткового розгортання власної інфраструктури, доцільно використати хмарний MQTT-брокер, зокрема пропонуєтьяс обати HiveMQ. Цей брокер дозволяє безкоштовно створити кластер, до якого можна підключати обидва клієнти – як ESP32 так і Node-RED, що дзволить забезпечити обмін повідомленнями через визначені MQTT-топіки.

Для створення хмарного MQTT-брокера на платформі HiveMQ насамперед потрібно перейти на офіційний сайт сервісу HiveMQ Cloud [24]. Користувач проходить реєстрацію, де вказує електронну пошту та пароль, після чого отримує

доступ до особистого кабінету, де можна керувати брокерними кластерами. У цьому кабінеті користувач ініціює створення нового кластеру, який у подальшому виконуватиме функцію хмарного брокера для обміну повідомленнями між ESP32 та Node-RED.

Під час створення кластера було вказано його назву, обрано регіон розміщення сервера (Європа), а також обрано тарифний план із безкоштовною підтримкою Free Tier.

В результаті було створено кластер хмарного брокера у HIVEMQ та отримано у розпорядження такі параметри: URL сервера, порт (зазвичай 8883 для безпечного з'єднання через TLS), а також можливість створення облікових записів клієнтів із унікальними іменами користувачів і паролями. Ці дані використовувалися як на стороні мікроконтролера ESP32, так і в середовищі Node-RED, щоб підключитися до брокера та здійснювати публікацію чи підписку на відповідні MQTT-топіки. Параметри створеного кластеру хмарного брокера HIVEMQ подано на рис. 3.1.

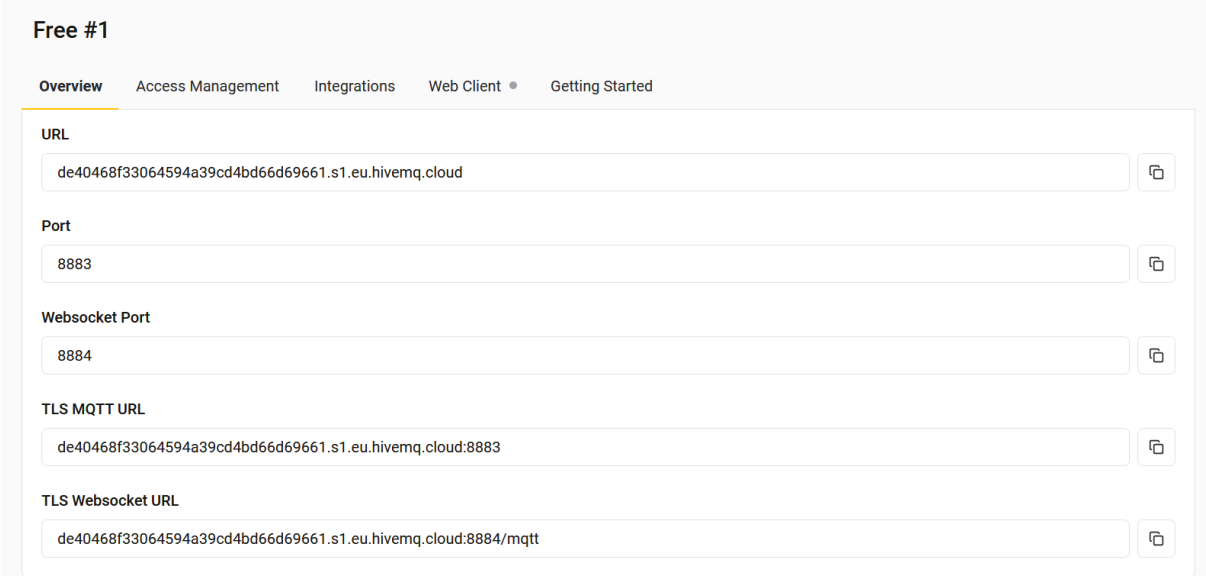


Рисунок 3.1 – Створений кластер хмарного брокера у HIVEMQ

Крім того, у налаштуваннях брокера можна вмикати або вимикати обробку з'єднань, переглядати журнал активності, а також перевіряти стан клієнтів у реальному часі. Завдяки використанню HiveMQ Cloud, не потрібно розгортати власний локальний брокер або сервер, що значно спрощує розробку системи

доступу до розумного будинку, робить її масштабованою і доступною з будь-якого місця, де є інтернет.

Після створення хмарного MQTT-брокера на платформі HiveMQ було виконано налаштування модуля керування доступом (Access Management), що дозволяє контролювати, які клієнти мають право підключатися до брокера та взаємодіяти з певними топіками (рис. 3.2). У межах цього налаштування було створено окремого користувача із ім'ям guest і встановлено для нього пароль. Цей обліковий запис використовується як для мікроконтролера ESP32, так і для середовища Node-RED з метою автентифікації при з'єднанні з брокером (рис. 3.3, 3.4).

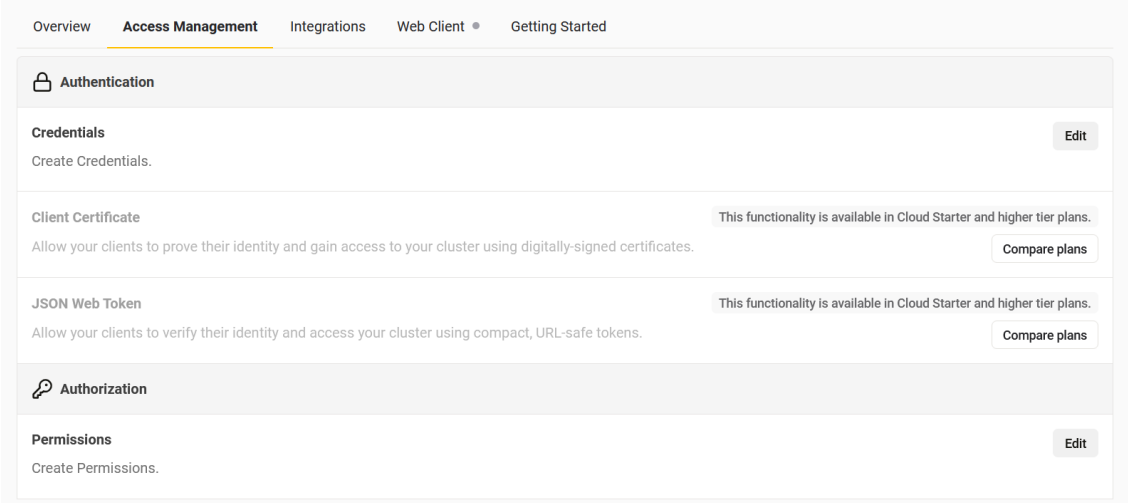


Рисунок 3.2 – Керування політикою доступу

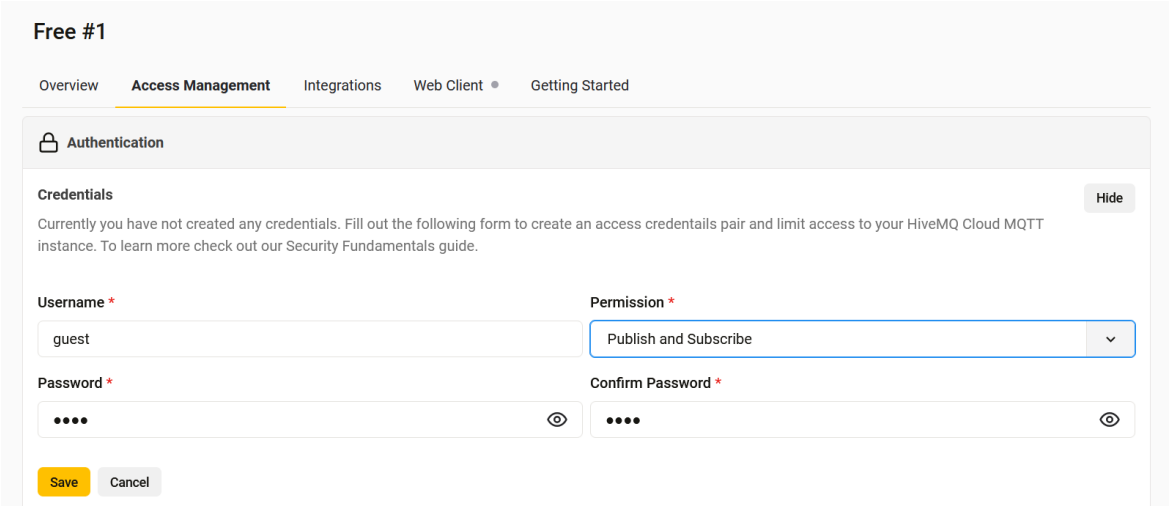


Рисунок 3.3 – Створення користувача та налаштування його параметрів

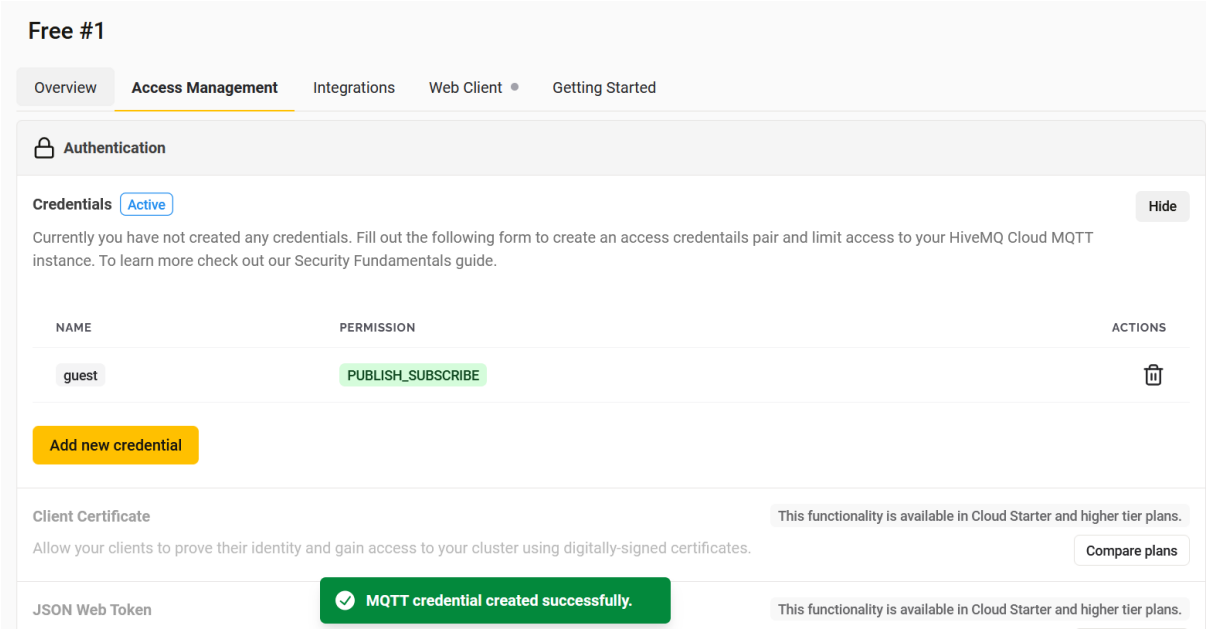


Рисунок 3.4 – Створений користувач у HIVEMQ

3.2 Реалізація клієнта керування

У запропонованій системі передбачено два основних MQTT-клієнти – клієнт пристрою та клієнт керування. Обидва клієнти здійснюють комунікацію через хмарний брокер.

Клієнтом керування виступає середовище Node-RED, яке виконує функції моніторингу, керування доступом та обробки подій. Саме через Node-RED здійснюється надсилання команд до пристрою (наприклад, відкриття або закриття дверей), а також прийом повідомлень від клієнта пристрою для формування журналу подій та надсилання сповіщень у випадку перевищення кількості дозволених спроб доступу.

Спочатку в середовищі Node-RED було встановлено додаткові вузли, такі як ui_table, mail та dashboard, які потрібні були для реалізації основних функцій програмно-апаратного пристрою. Вузол ui_table дозволив організувати зручне відображення журналу подій у вигляді таблиці, вузол mail забезпечив можливість автоматичної відправки електронних повідомлень про помилкові спроби входу, а dashboard став основою для створення інтерактивного інтерфейсу керування доступом (рис. 3.5).

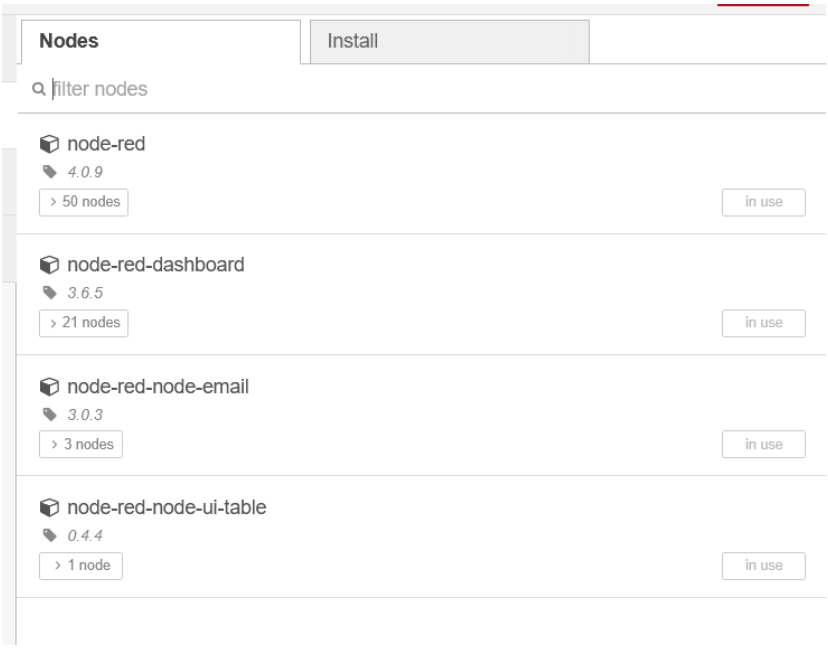


Рисунок 3.5 – Встановлені вузли у Node-RED

Далі наступним кроком було виконано підключення та тестування до створеного хмарного брокера HIVEMQ. Для цього додано MQTT out вузол і виконано додавання нового брокера (рис. 3.6).

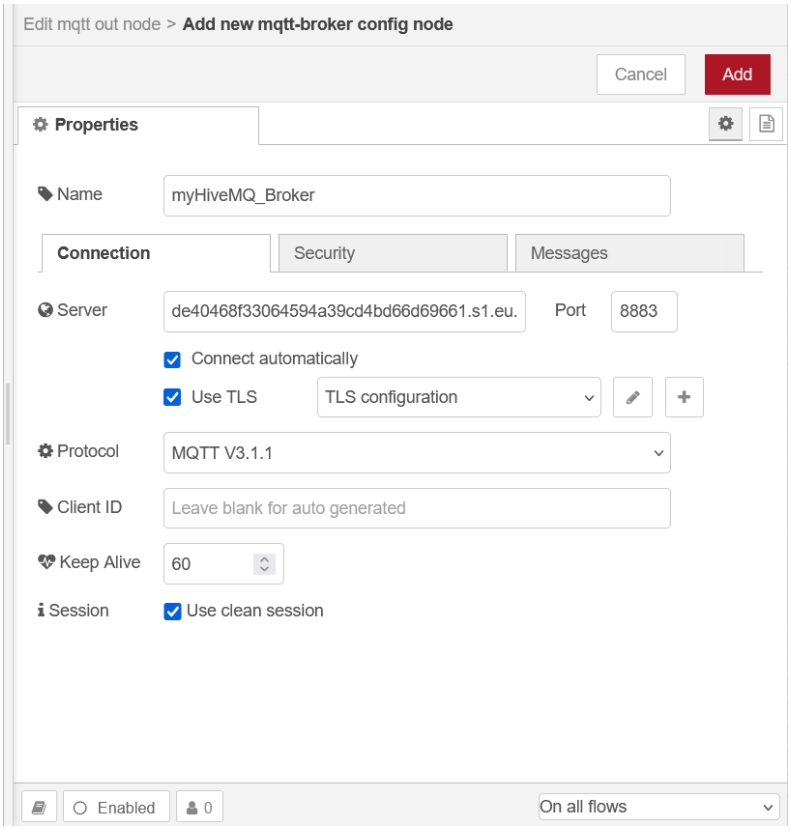


Рисунок 3.6 – Створення брокера у NODE-RED

При налаштуванні брокера було встановлено параметри, отримані від хмарного брокера HiveMQ. Зокрема в якост адреси сервера було вказано de40468f33064594a39cd4bd66d69661.s1.eu.hivemq.cloud, а також порт 8883. Також активовано опції Use TLS та автоматичного підключення.

Також у вкладці Security було обрано користувача, що був створений у HIVEMQ брокері (рис. 3.7).

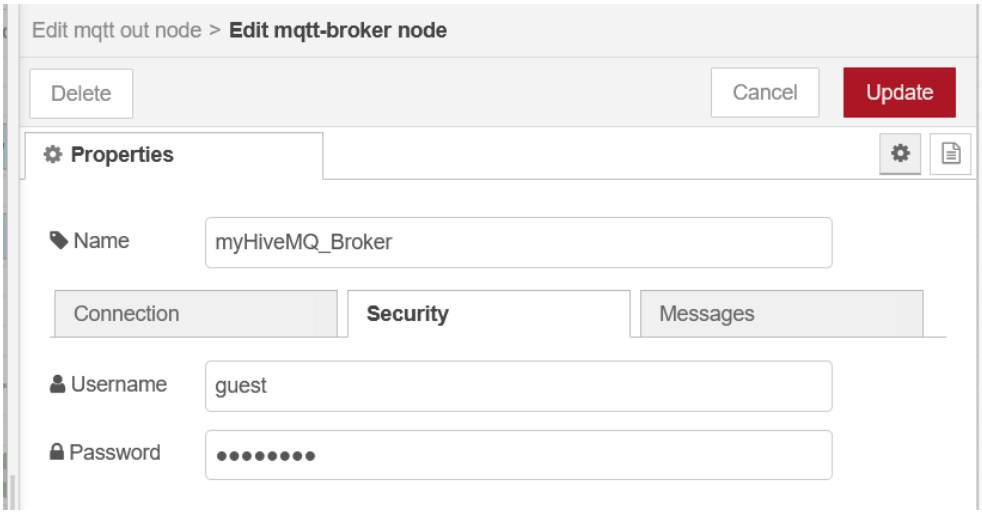


Рисунок 3.7 – Налаштування параметрів безпеки

Після цього було протестовано комунікацію клієнта керування із хмарним брокером HIVEMQ. Для цього було додано 4 вузли: timestamp – для активації сценарію, MQTT out вузол – здійснює публікацію у тему test/topic часової мітки, MQTT in вузол – здійснює підписку на тему test/topic (отримує повідомлення) і вузол debug – для виведення повідомлення (рис. 3.8).

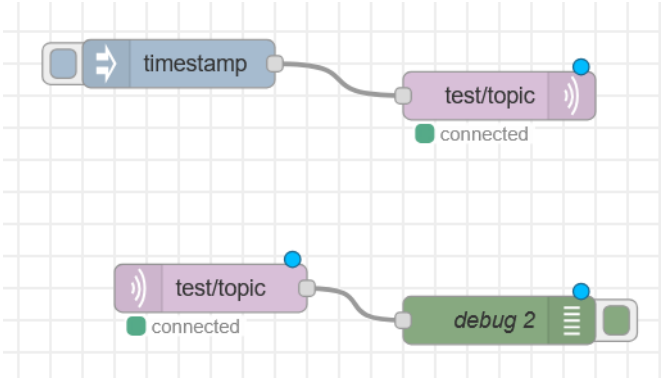


Рисунок 3.8 – Тестування взаємодії клієнта керування із хмарним брокером

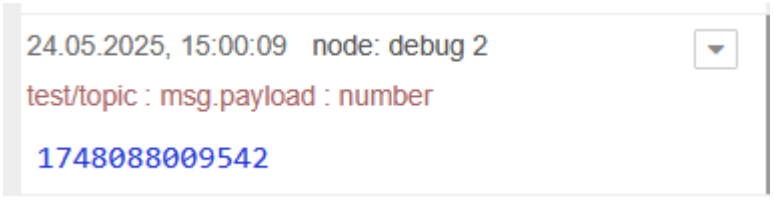


Рисунок 3.9 – Отримне повідомлення

Також було використано MQTT Explorer – безкоштовну GUI-програма, яка дозволяє бачити всі топіки і повідомлення в реальному часі (3.10).

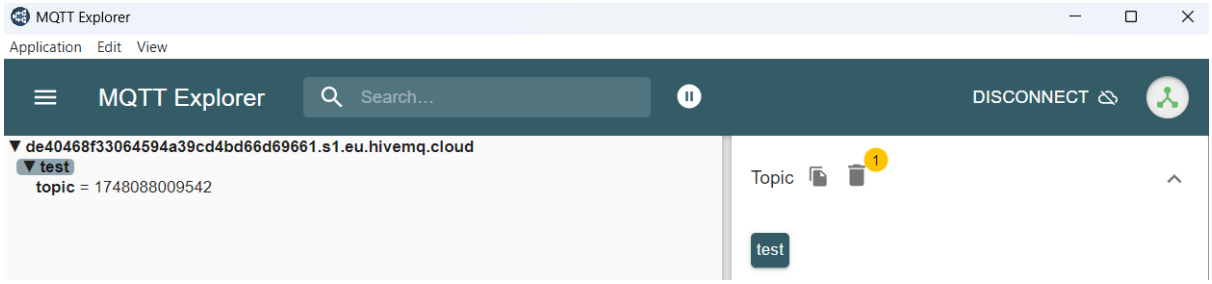


Рисунок 3.10 – Перехоплене повідомлення у тему test/topic у MQTT Explorer

Після тестування взаємодії клієнта керування із хмарним брокером було реалізовано сценарії для клієнта керування.

Сценарії NODE-RED для клієнта керування наведено на рис. 3.11. Цей потік Node-RED реалізує управління доступом до дверей на основі ESP32, MQTT-брокера HiveMQ та Node-RED як середовища обробки повідомлень. Основна мета – забезпечити дистанційне керування дверима, фіксацію подій у журналі та надсилання повідомлень про помилкові спроби доступу.

У верхній частині потоку розміщено два вхідних вузли – "Close Door Button" та "Open Door Button", які імітують натискання кнопок для закриття або відкриття дверей. Ці кнопки надсилають відповідні команди через MQTT до топіка access/control, використовуючи вузол "MQTT Open Door". Це дозволяє ESP32 отримати команду відкриття або закриття дверей.

У центральній частині реалізовано обробку подій доступу. Вузол "Log Events" підписаний на MQTT-топік access/log, в який ESP32 надсилає повідомлення про всі спроби доступу – як вдалі, так і невдалі. Ці повідомлення проходять через

функціональний вузол Wrap data, який обгортає вхідні дані у список (тобто msg.payload = [msg.payload]; return msg;). Це робилось для правильного відображення у таблиці. Далі оброблені дані надсилаються у вузол "Access Log Table", де візуалізується журнал подій, та паралельно дублюються у "Debug log" для зневадження.

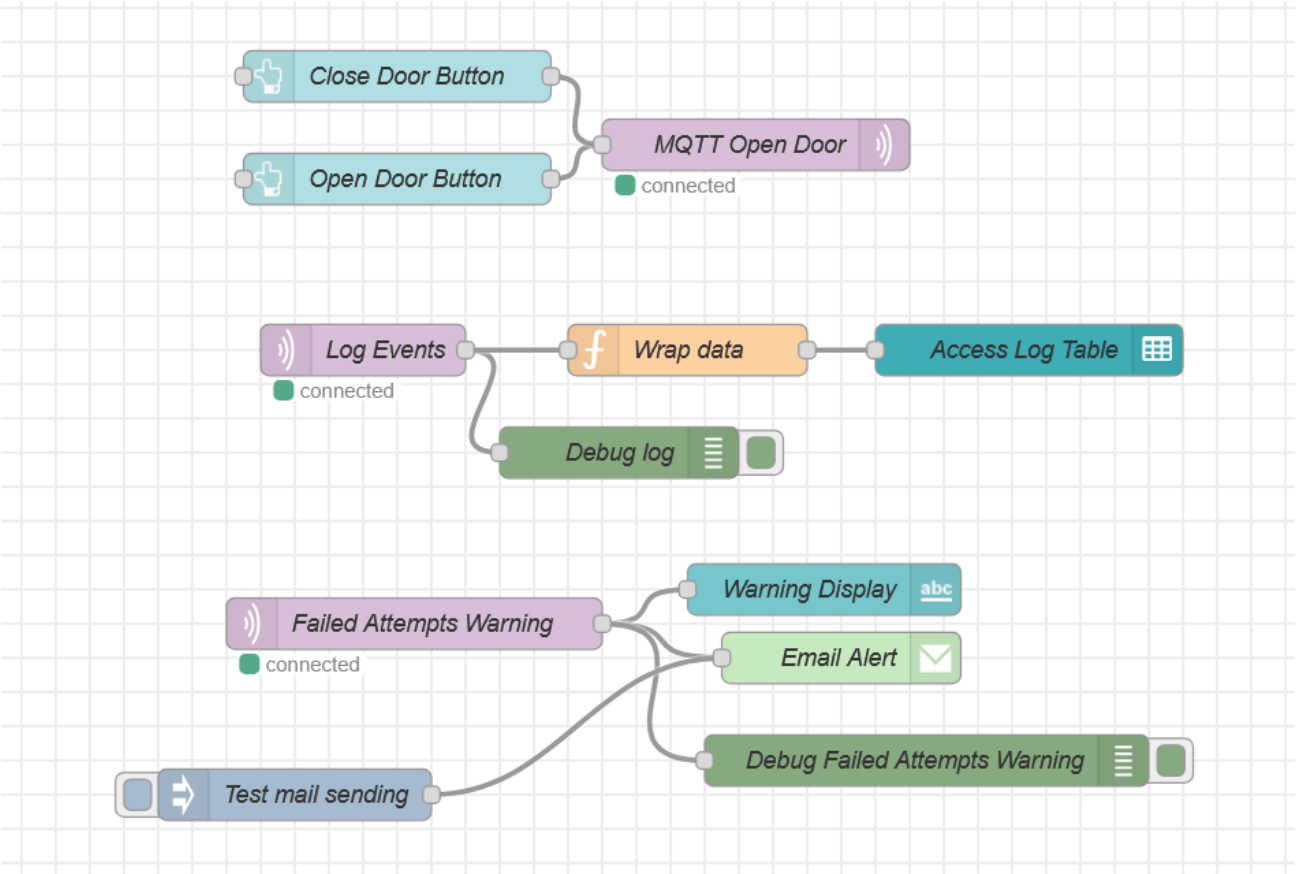


Рисунок 3.11 – Сценарії NODE-RED для клієнта керування

У нижній частині потоку оброблялись помилкові спроби входу. Вузол "Failed Attempts Warning" підписаний на топік access/warning, куди ESP32 надсилає повідомлення у разі невдалої авторизації (наприклад, неправильний PIN-код). Node-RED реагує на це трьома способами: показує повідомлення через вузол "Warning Display", надсилає електронного листа за допомогою вузла "Email Alert" і дублює інформацію у "Debug Failed Attempts Warning". Для тестування відправки листів передбачено окрему кнопку "Test mail sending", що дозволяє вручну ініціювати перевірку поштового сповіщення.

Крім цього, для тестування додано кнопку "Test mail sending", яка, використовується для перевірки коректності конфігурації email-відправлення без необхідності генерувати реальні події попередження.

Розглянемо детальніше вузли, що були використані у NODE-RED та їх налаштування.

Загалом у реалізованому потоці Node-RED було використано три вузли для роботи з протоколом MQTT, кожен з яких виконує певну функцію в межах системи контролю доступу. У всіх цих вузлах в якості MQTT-сервера було обрано хмарний брокер HiveMQ, який попередньо налаштовано як централізований посередник для обміну повідомленнями між ESP32 та Node-RED (рис. 3.12). Це дозволило забезпечити віддалене з'єднання між пристроями.

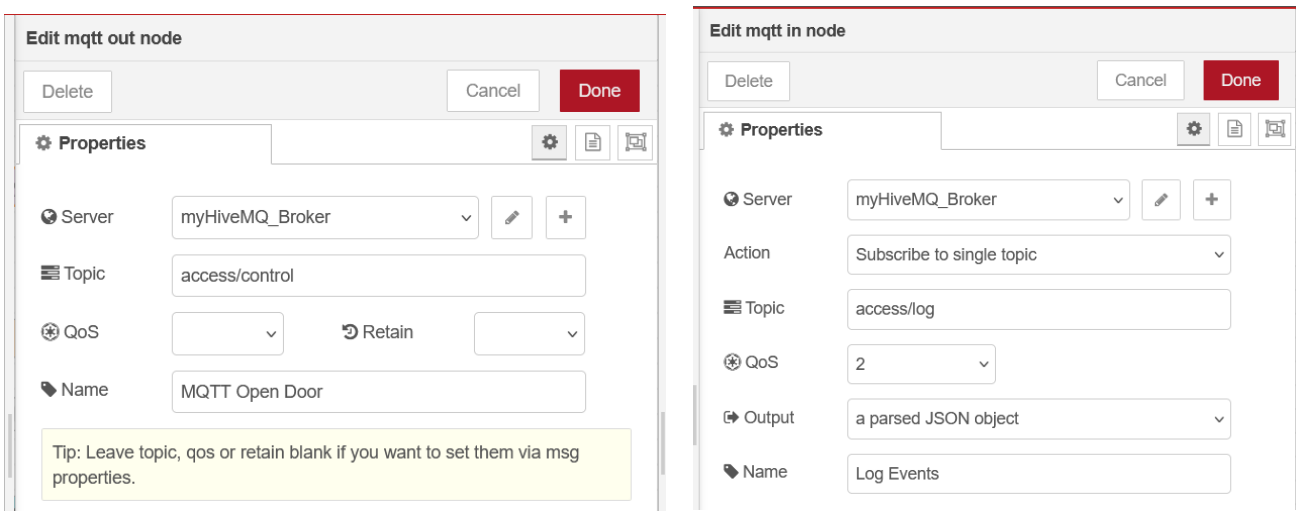


Рисунок 3.12 – Властиості вузлів MQTT (MQTT Open Door та Log Event)

Вузол Access Log Table у потоці Node-RED виконує функцію візуального журналу подій доступу, який дозволяє адміністратору або користувачу системи переглядати всю історію входів у реальному часі. Він приймає дані, які надсилає ESP32 через MQTT у топік access/log, і після обробки у функціональному блоці Wrap data ці події виводяться у табличному форматі. Передбачено три поля: status time та input. Ці поля отримуються із об'єкта, який надходить в топік, і трансформуютьяс у стовпці таблиці.

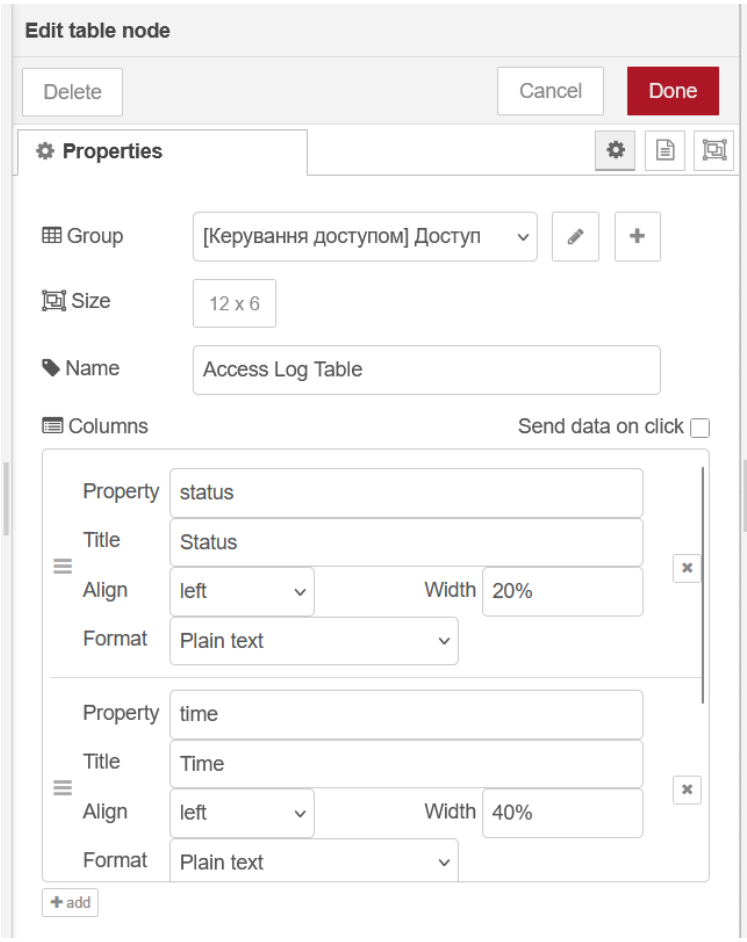


Рисунок 3.13 – Властивості вузла Access Log Table

Ще одним вузлом у реалізованому потоці є Email Alert, який належить до типу send mail. Цей вузол використовується для автоматичного надсилання електронних листів із попередженнями про несанкціоновані спроби доступу, наприклад, у випадках, коли ESP32 фіксує кілька помилкових введень коду. Надсилання листа відбувається автоматично, що дозволяє оперативно інформувати користувача про можливу загрозу безпеці.

Під час налаштування вузла send mail у середовищі Node-RED виникла потреба використати наявний обліковий запис Gmail як поштовий сервер для надсилання повідомлень. Проте стандартна аутентифікація з використанням пароля облікового запису Gmail не працює з Node-RED через політику безпеки Google, яка забороняє стороннім застосункам із низьким рівнем захисту використовувати звичайний пароль.

Щоб обійти це обмеження, було створено спеціальний пароль додатка (App Password) у налаштуваннях Google-акаунту. Для цього, насамперед, у обліковому записі Gmail було увімкнено двофакторну автентифікацію (2FA), яка є обов'язковою умовою для генерації паролів додатків. Після цього, у розділі "Безпека" в налаштуваннях акаунта було згенеровано окремий 16-значний пароль, призначений спеціально для сторонніх застосунків.

Саме цей згенерований пароль було введено у полі Password під час налаштування вузла send mail в Node-RED (рис. 3.14).

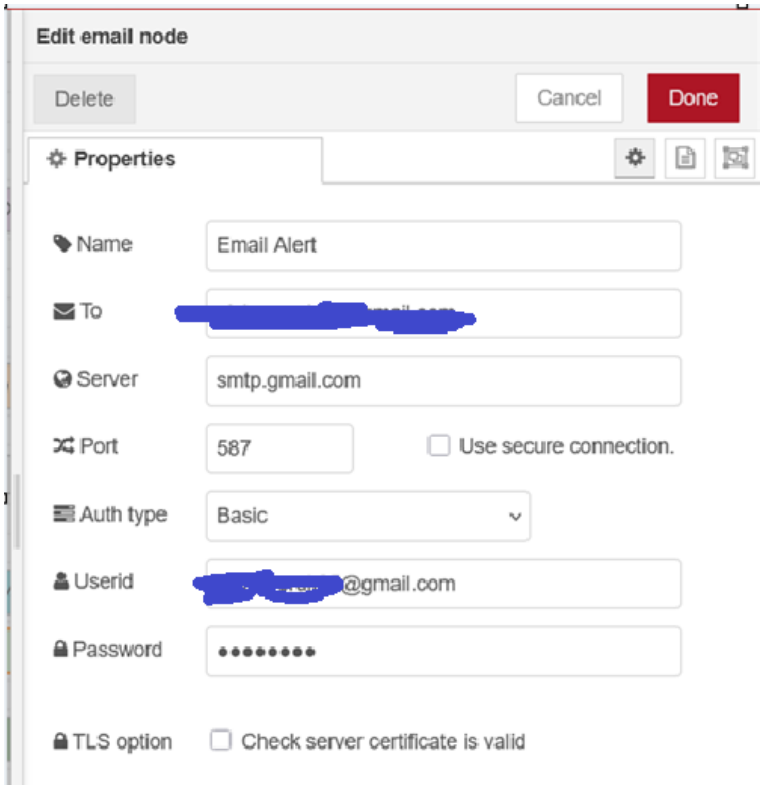


Рисунок 3.14 – Властивості вузла Email Alert

3.3 Реалізація клієнта пристрою

Для для реалізації клієнта пристрою було створено програму для мікроконтролера ESP32, яка забезпечує роботу прграмно-апаратного пристрою. Діаграма алгоритму роботи програмно-апаратного засобу керування доступом до кіберфізичної системи Розумний будинок із функцією сповіщення на основі ESP32 наведено на рис. 3.15.

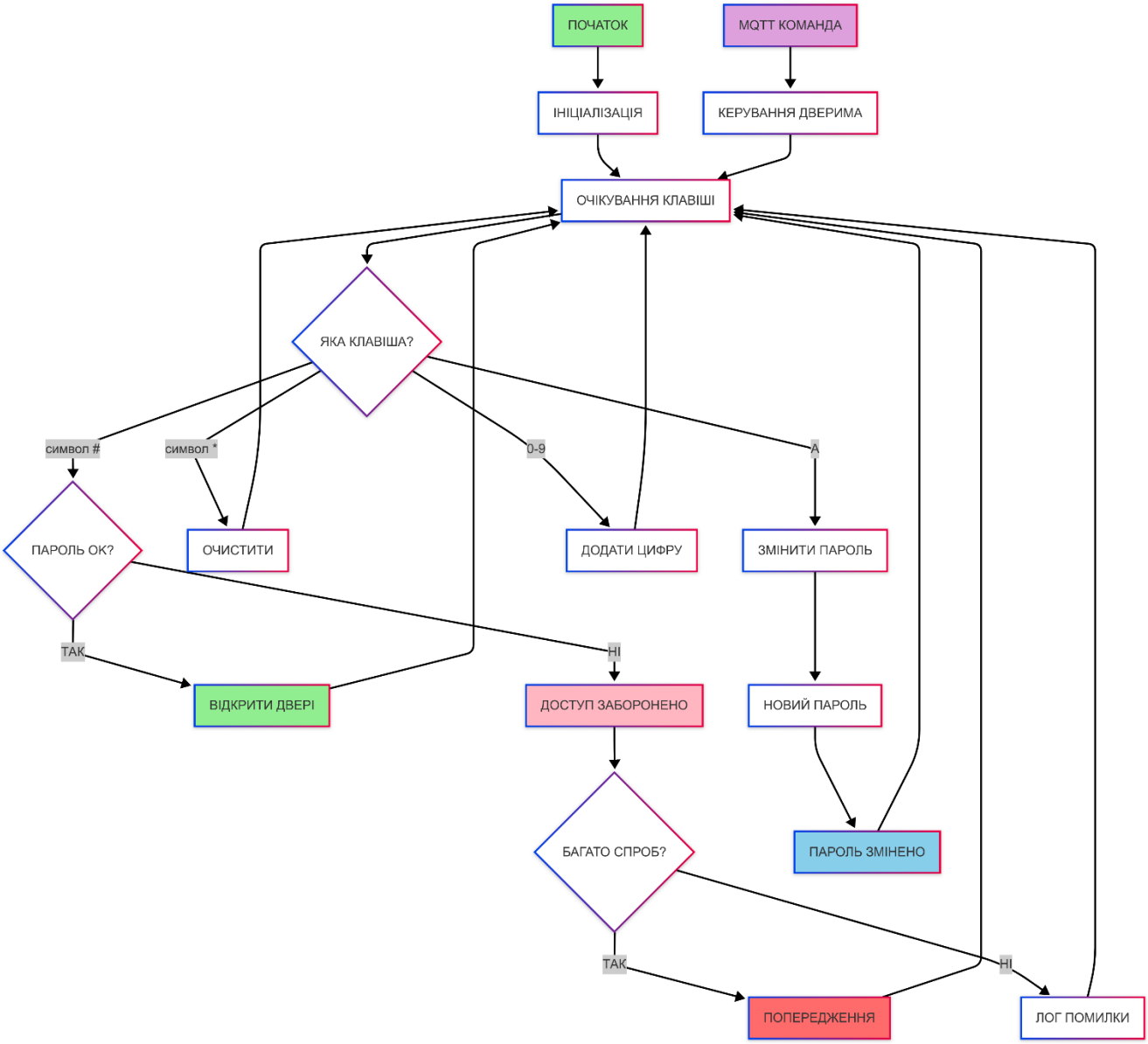


Рисунок 3.15 – Діаграма алгоритму роботи програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32

Клієнт пристрою виконує взаємодію з користувачем через клавіатуру 4x4 та дисплей LCD з інтерфейсом I2C. У центрі системи – модуль ESP32, який виконує обчислення, керування сервомотором, підключення до мережі Wi-Fi, а також взаємодію з хмарним брокером HiveMQ через захищене з'єднання за протоколом MQTTs.

На початку програми відбувається ініціалізація всіх апаратних компонентів – підключення клавіатури, LCD-дисплея, сервоприводу, а також встановлення

параметрів для MQTT-з'єднання. У якості мережевого інтерфейсу використовується WiFiClientSecure, що дозволяє встановити SSL-з'єднання з брокером HiveMQ, який працює в хмарному середовищі. Користувач для MQTT доступу має ім’я guest, що були попередньо створені в системі керування доступом брокера.

Після підключення до мережі Wi-Fi ESP32 намагається під’єднатися до брокера MQTT та підписується на топік access/control, щоб реагувати на команди від керуючого клієнта – зокрема клієнта керування Node-RED. У випадку отримання повідомлення open, ESP32 активує сервомотор, що фізично відкриває двері (повертає важіль на 180 градусів), а при отриманні команди close – повертає його назад у закрите положення.

Основна логіка взаємодії з користувачем побудована на обробці натискань клавіш. Користувач вводить пароль, натискає # для підтвердження, і система перевіряє правильність введеного значення. У разі успішного введення відкриваються двері, публікується повідомлення у вигляді JSON у топік access/log з відміткою про надання доступу. Якщо пароль неправильний, система так само публікує повідомлення про відмову у доступі. При більш ніж трьох поспіль невдалих спробах авторизації ESP32 надсилає попередження у топік access/warning – це дозволяє керуючому клієнту Node-RED миттєво зреагувати, наприклад, надіслати повідомлення електронною поштою.

Окрім звичайної перевірки паролю, реалізовано функцію зміни паролю. Якщо користувач натискає клавішу A, йому пропонується ввести новий пароль, а потім підтвердити його повторним введенням. Лише у випадку співпадіння обох введених значень новий пароль зберігається в системі, а також публікується MQTT-журнал із відміткою про зміну паролю.

Таким чином, клієнт пристрою на базі ESP32 не тільки здійснює апаратну взаємодію з користувачем і фізичними компонентами (дисплеєм, клавіатурою, сервомотором), а й є активним учасником MQTT-комунікації, що дозволяє централізовано контролювати події в системі доступу через Node-RED. Його програмна реалізація включає не лише базові можливості контролю, а й підтримує

розширені функції на кшталт журналювання, попередження про підозрілу активність, зміну облікових даних тощо.

3.4 Тестування роботи програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32

У ході розробки було проведено тестування взаємодії двох клієнтів, які разом утворюють програмно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32. Першим клієнтом виступав безпосередньо пристрій, побудований на базі мікроконтролера ESP32, до якого була підключена клавіатура для введення коду доступу. Цей пристрій забезпечував первинну обробку введених даних, публікацію повідомлень до MQTT-брокера HiveMQ, а також генерував попередження у разі виявлення кількох невдалих спроб доступу. Другим клієнтом був програмний інтерфейс керування, реалізований у середовищі Node-RED, який підписувався на відповідні MQTT-топіки, приймав повідомлення від ESP32, виконував логування подій, здійснював відображення інформації та, у разі потреби, надсилав електронні сповіщення.

У процесі тестування було перевірено відповідність проєктованого пристрою встановленим вимогам, а саме: забезпечено стабільну роботу функції віддаленого розблокування дверей за допомогою інтерфейсу Node-RED, що дозволяє керувати станом замка в режимі реального часу. Також перевірено ведення журналу подій – кожна спроба входу, незалежно від її результату (успішна чи помилкова), коректно публікується до MQTT-топіку, після чого дані обробляються у Node-RED і зберігаються у вигляді таблиці для подальшого аналізу. Окрім того, було протестовано механізм миттєвого сповіщення про помилкові спроби входу: якщо ESP32 фіксує понад три невірні спроби введення пароля, пристрій автоматично генерує попереджувальне повідомлення і надсилає його до відповідного MQTT-топіку. Node-RED, отримавши це повідомлення, одразу формує електронне

сповіщення, яке надсилається на вказану поштову скриньку, що дозволяє оперативно реагувати на потенційні загрози безпеці системи.

Для тестування клієнта пристрою було задіяно симулятор Wokwi. Було зібрано схему, яка подана на рис. 3.16. Вона включала мікроконтрлер ESP32, клавіатуру 4x4, LCD дисплей та сервопривід.

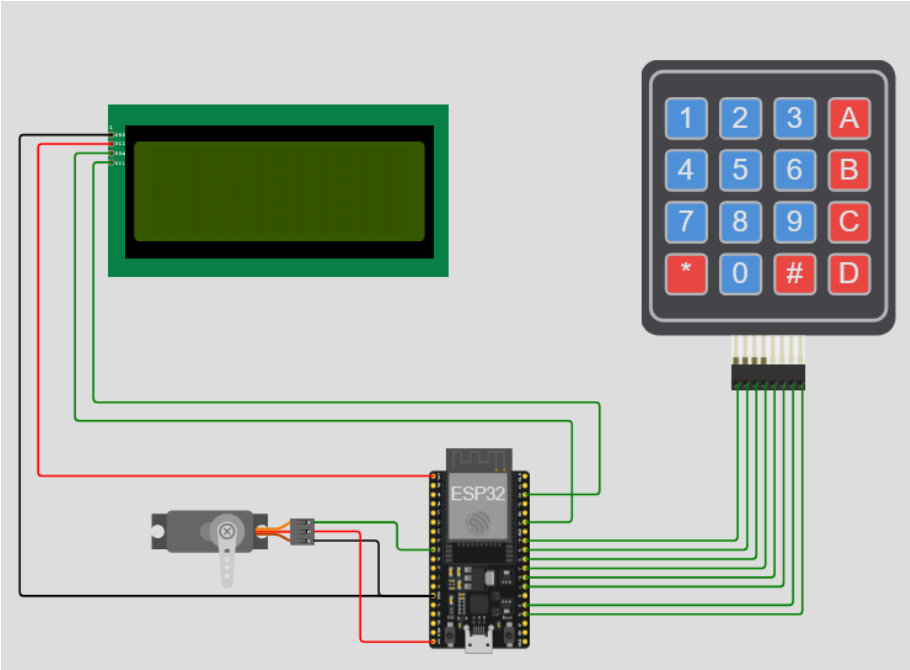


Рисунок 3.16 – Схема з’єднань клієнта пристрою проєктованого програмно-апаратного засобу у Wokwi

Спочатку було протестовано роботу пристрою локально без взаємодії із хмарним брокером. Було перевірено коректність зчитування натискань на клавіатурі та відображення введених символів на LCD-дисплеї. Також було протестовано механізм розпізнавання правильного пароля та відкриття дверей за допомогою серводвигуна – при введенні вірного пароля сервопривід коректно відкривав і зачиняв замок. Також перевірено обробку помилкових спроб входу – після кожного невірного введення пароля система виводила відповідне повідомлення та оновлювала лічильник невдалих спроб. Було протестовано функцію зміни пароля, яка реалізована через клавішу 'A': новий пароль успішно вводиться, підтверджується і зберігається в оперативній пам’яті пристрою, після

чого використовується для подальшої перевірки доступу. Усі локальні функції працювали стабільно і відповідно до очікувань.

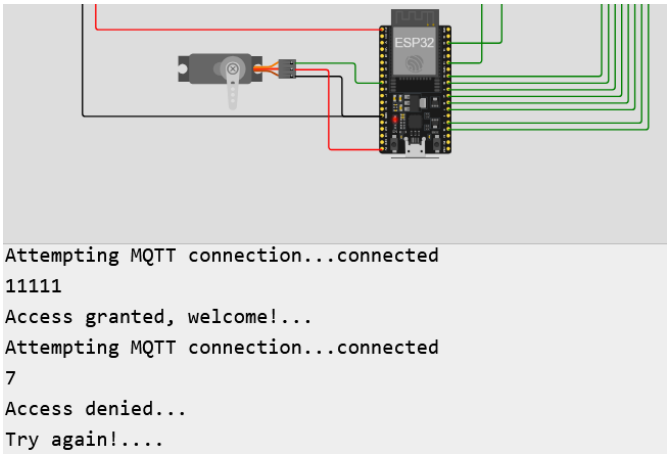


Рисунок 3.17 – Локальне тестування пристрою (успішний та не успішний вхід)

Далі було протестовано взаємодії із іншим клієнтом використовуючи комунікацію через хмарний брокер. На рис. 3.18 можна відмітити, що при отриманні повідомлення open (Message arrived [access/control]) відбулось відкриття дверей.

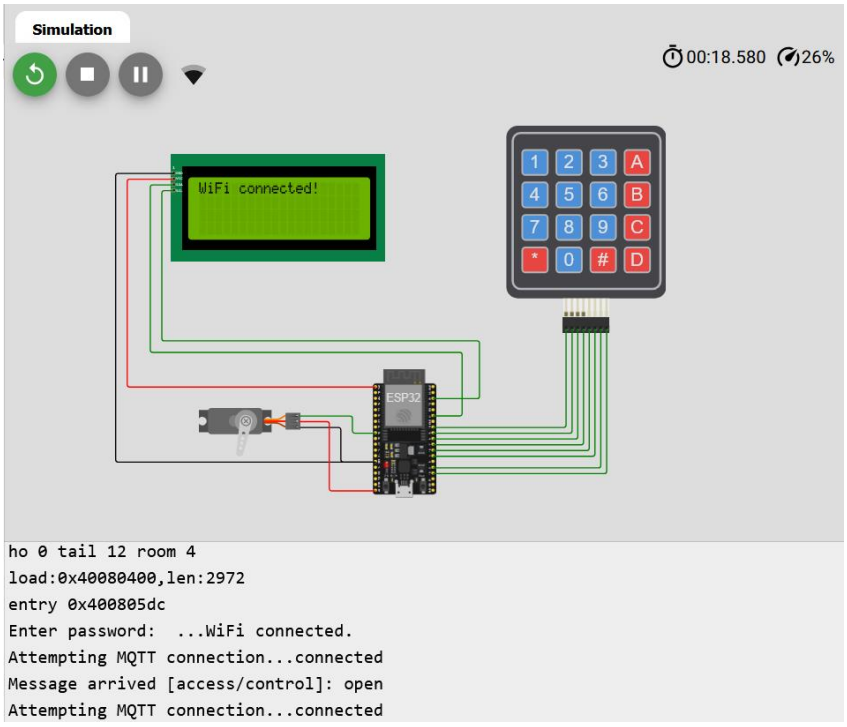


Рисунок 3.18 – Керування пристроєм через отримання команди від клінта управління

Також, як і очікувалось, було спостережено повідомлення open у переглядачі MQTT Explorer (рис. 3.19).

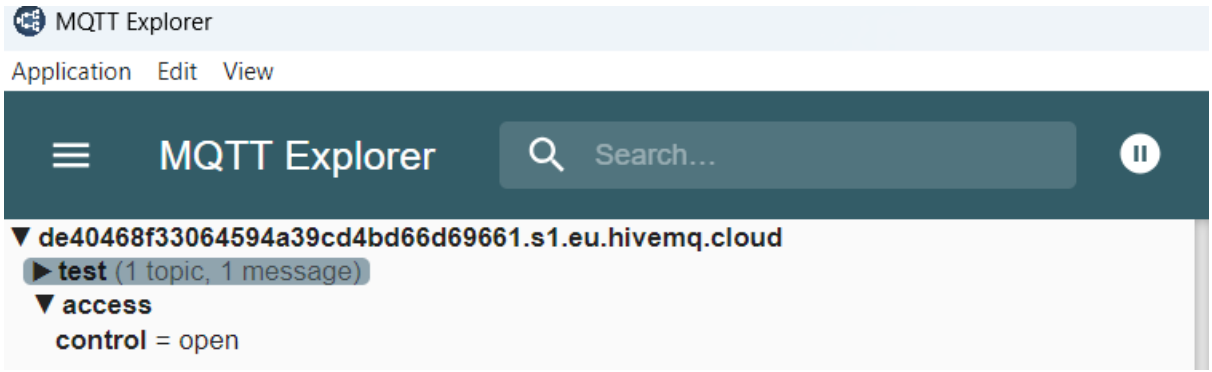


Рисунок 3.19 – Отримане повідомлення open у MQTT Explorer

Надсилання повідомлення для відкриття/закриття дверей здійснюється із клієнта керування через інтерфейс (рис. 3.20). Розроблений інтерфейс призначений для керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32. У верхній частині розміщені дві кнопки: «Відкрити двері» та «Закрити двері», які надсилають відповідні команди. Нижче відображається таблиця з журналом подій, що містить поля Status (статус доступу), Time (час події) та Input (введене значення).

В якості ще одного тестового випадку було здійсне серію невдалих спроб доступу (рис. 3.21 та рис. 3.22).

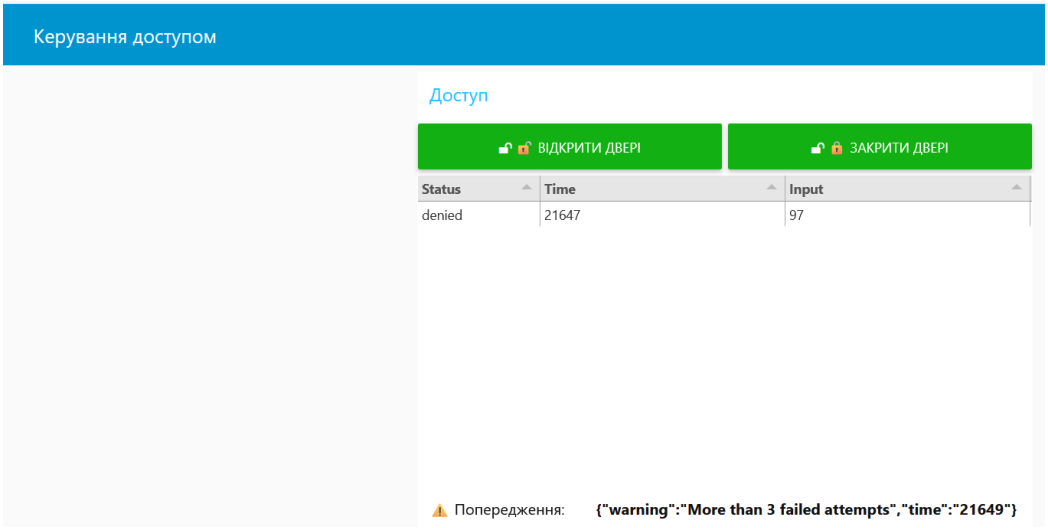


Рисунок 3.20 – Інтерфейс клієнта керування

```
Attempting MQTT connection...connected
1
Access denied...
Try again!....9
Access denied...
Try again!....1
Access denied...
Try again!....97
Access denied...
Try again!....Attempting MQTT connection...
```

Рисунок 3.21 – Відображення спроб доступу у клієнті пристрою (Wokwi)

Рисунок 3.22 – Відображення спроб доступу у клієнті керування (NODE-RED)

В результаті було зафіксовано спробу доступу зі статусом denied – доступ заборонено. В такому випадку було отримано попередження, яке відоображалось в інтерфейсі у форматі JSON: {"warning":"More than 3 failed

attempts","time":"21649"}, яке повідомляє про перевищення кількості невдалих спроб.

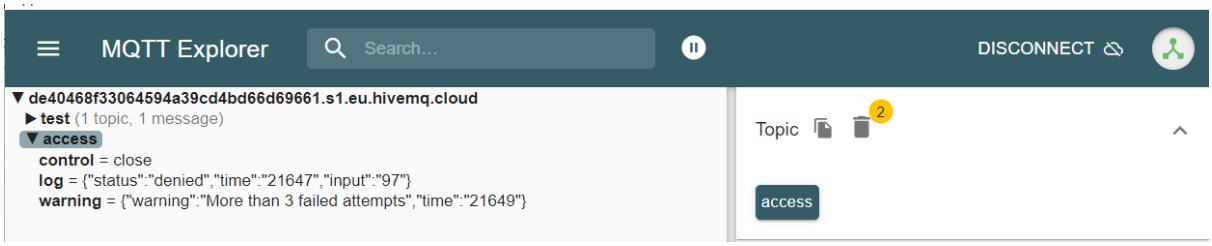


Рисунок 3.23 – Фіксація надходження повідомлень у трьох темах MQTT – control, log та warning

Також було отримано повідомлення про перевищення спроб доступу на електронну пошту (рис. 3.23).

Рисунок 3.23 – Отримане повідомлення про перевищення спроб доступу на електронну пошту

3.5 Висновки

У результаті проведеного дослідження та практичної реалізації було змодельовано й протестовано повнофункціональну систему керування доступом до кіберфізичної системи типу «Розумний будинок», що поєднує сучасні апаратні компоненти, програмну логіку та хмарні сервіси для забезпечення безпеки, зручності та гнучкості керування. Центральним елементом системи є мікроконтролер ESP32, який виконує функції перевірки автентичності користувача, керування виконавчими механізмами (зокрема, відкривання та закривання дверей), а також формування сповіщень про підозрілу активність.

На рівні взаємодії користувача з системою реалізовано функцію введення коду доступу, що аналізується в реальному часі. У разі правильного введення коду ESP32 генерує сигнал на відкривання дверей і надсилає підтвердження до інтерфейсу користувача. Якщо ж код неправильний, система фіксує невдалу спробу. Після перевищення встановленого ліміту спроб (наприклад, трьох послідовних невдач) система формує тривожне повідомлення, що передається до клієнта у вигляді попередження – це дозволяє оперативно реагувати на можливі загрози фізичної безпеки.

Для реалізації обміну повідомленнями між ESP32 та інтерфейсом керування використано хмарний MQTT-брокер HiveMQ, що забезпечує передачу даних у режимі реального часу. MQTT дозволяє організувати двосторонню комунікацію між мікроконтролером та користувачем без необхідності прямого мережевого підключення до пристрою, що є ключовою перевагою в умовах обмеженого доступу до внутрішньої локальної мережі або динамічної IP-адресації.

У системі створено два окремі MQTT-клієнти. Перший клієнт, реалізований у вигляді інтерфейсу користувача (може бути вебзастосунок або мобільний додаток), виконує роль керуючого пристрою: він надсилає команди, отримує повідомлення про стан системи, а також реагує на критичні сповіщення. Другий клієнт – це сам ESP32, який отримує команди, виконує відповідні дії та публікує інформацію про події (наприклад, відкриття дверей або спробу злому).

Для віртуального моделювання роботи пристрою з ESP32 використано онлайн-симулятор Wokwi, який дозволив протестувати логіку роботи мікроконтролера без необхідності фізичного збирання схеми. Через Wokwi реалізовано повноцінне підключення ESP32 до MQTT-брокера, завдяки чому стала можливою емуляція комунікації з реальним користувачем.

ВИСНОВКИ

Таким чином було змодельовано та протестовано систему керування доступом для кіберфізичної системи типу "Розумний будинок". На базі мікроконтролера ESP32 створено пристрій, який забезпечує перевірку введених даних, відкривання та закривання дверей, а також сповіщення про підозрілу активність, зокрема перевищення кількості невдалих спроб доступу.

У першому розділі виконано досліджено сучасні програмно-апаратні рішення, що реалізують доступ до системи Розумного будинку, проаналізовано відомі апаратні та хмарні платформи, виконано аналіз традиційних та сучасних технологій віддаленого контроолю.

В другому розділі були визначені вимоги до програмно-апаратного засобу та проєктуванням його структури. Зокрема було визначено такі вимоги як реалізація віддаленого керування доступом через MQTT, локальна автентифікацію з клавіатури та зміна пароля, а також ведення журналу подій і опрацювання помилкових спроб входу з генерацією сповіщень. Також описано структуру програмно-апаратного засобу керування доступом до кіберфізичної системи "Розумний будинок" на основі ESP32, включно з трьома основними компонентами системи: клієнтом пристрою, хмарним MQTT-брокером HiveMQ та клієнтом керування Node-RED. Було проаналізовано функціональні можливості ESP32 як локального елемента авторизації, який забезпечує автентифікацію користувачів, обробку спроб доступу та автономну зміну пароля. Крім того, розглянуто роль клієнта керування у Node-RED як інтерфейсу для централізованого моніторингу, управління доступом і ведення журналу подій. Особлива увага приділялася функціям сповіщення про події та надзвичайні ситуації, що реалізуються через MQTT-топіки broker'а HiveMQ, який виступає надійним посередником для обміну повідомленнями між усіма компонентами системи. Також проаналізовано апаратні компонети та оцінено їх вартість. Було визначено, що загальна вартість програмно-апаратного пристрою не перевищує 400 грн, що з урахуванням виконання покладених функцій, є оптимальним рішенням.

В третьому розділі було зосереджено увагу на практичній реалізації програмно-апаратного засобу керування доступом до кіберфізичної системи типу «Розумний будинок». Основну частину робіт присвячено налаштуванню хмарного середовища обміну повідомленнями, створенню клієнтів системи та проведенню тестування для перевірки працездатності та функціональності розробленого рішення.

Першим етапом було створено хмарний MQTT-брокер на платформі HiveMQ, який виступає центральною ланкою для обміну повідомленнями між клієнтом пристрою (ESP32) та клієнтом керування. Використання HiveMQ дозволило забезпечити стабільний зв’язок у режимі реального часу, а також масштабованість і доступність сервісу з будь-якого місця, що має доступ до Інтернету.

На другому етапі реалізовано клієнт керування, що функціонує в середовищі Node-RED. Він забезпечує зручний графічний інтерфейс для взаємодії з системою, дає змогу віддалено вводити код доступу, контролювати стан дверей, змінювати пароль, а також отримувати сповіщення про події системи. Крім того, в Node-RED реалізовано логіку перевірки коректності введених кодів і обробки подій із подальшою візуалізацією журналу доступу.

Далі було створено клієнт пристрою – програмно реалізований у середовищі Wokwi віртуальний макет системи на основі ESP32, що моделює роботу фізичного пристрою. Він виконує автентифікацію користувача через клавіатуру, керує реле, яке умовно відкриває або закриває двері, фіксує помилкові спроби доступу та взаємодіє з MQTT-брокером через відповідні топіки.

На завершальному етапі проведено тестування всієї системи, під час якого перевірено коректність взаємодії між усіма компонентами, стабільність передачі повідомлень через HiveMQ, відповідність реакції пристрою введеним командам, а також працездатність механізму сповіщення у разі перевищення встановленої кількості помилкових спроб входу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Januário F., Leitão J., Cardoso A. and GilResilience P. Enhancement in Cyber-Physical Systems: A Multiagent-Based Framework, Rocha, Jorge, editor. *Multi-Agent Systems*. InTech, 13 Sept. 2017. Chapter 10, doi:10.5772/66595.

2. Nuki, URL: https://nuki.io/en-de/products/smart-lock-go (дата звернення 24.05.2025).

3. August smsrt locks, URL: https://august.com/products/yale-august-wi-fi-smart-lock-with-keypad-touch (дата звернення 24.05.2025).

4. Електронний контролер YALE LINUS чорний до циліндру, URL: https://lock.ua/catalog/zamki-dlya-smart-home/elektronniy-kontroler-yale-linus-chorniy-do-tsilindru.html (дата звернення 24.05.2025).

5. Andreas A., Aldawira C.R., Putra H.W., Hanafiah N., Surjarwo S., Wibisurya A. Door Security System for Home Monitoring Based on ESP32. *Procedia Computer Science*. 2019. Vol. 157. P. 673–682.

6. Dutta J., Wang Y., Maitra T., Islam S.H., Rawal B.S., Giri D. ES3B: Enhanced Security System for Smart Building using IoT. *IEEE International Conference on Smart Cloud (SmartCloud)*. 2018. P. 158–165.

7. Kamat M.N., Shinde D. Smart Door Security Control System Using Raspberry Pi. *International Journal of Innovations & Advancement in Computer Science*. 2017. Vol. 6, No. 11. P. 1–4.

8. Kodali R.K., Jain V., Bose S., Boppana L. IoT Based Smart Security and Home Automation System. International Conference on Computing, *Communication and Automation (ICCCA)*. 2016. P. 1286–1289.

9. Agarwal A., Hada N., Virmani D., Gupta T. A Novel Design Approach for Smart Door Locking and Home Security using IoT. *A High Impact Factor & UGC Approved Journal*. 2017. Vol. 6, No. 8. P. 1–5.

10. Babiuch M., Postulka J. Smart Home Monitoring System Using ESP32 Microcontrollers. *Internet of Things*. 2020. Vol. 1. P. 1–20.

11. Cahyono F.Y.A., Suharto N., Mustafa L.D. Design and Build a Home Security System based on an ESP32 Cam Microcontroller with Telegram Notification. *Journal of Telecommunication Network*. 2022. Vol. 12, No. 2. P. 1–10.

12. Ramadhani S., Putri D.P. Design of a Home Door Security System Based on NodeMCU ESP32 Using a Magnetic Reed Switch Sensor and Telegram Bot Application. *Sinkron: Jurnal dan Penelitian Teknik Informatika*. 2023. Vol. 8, No. 4. P. 1–8.

13. Ziad A., Darnila E., Kurniawati. Development and Implementation of an ESP32 Microcontroller and Monitoring System for Smart Door Lock Using RFID Sensor for E-KTP ID and Fingerprint Based on the Internet of Things. *Proceedings of Malikussaleh International Conference on Multidisciplinary Studies (MICoMS)*. 2023. Vol. 4. P. 1–7.

14. Cahyono F.Y.A., Suharto N., Mustafa L.D. Design and Build a Home Security System based on an ESP32 Cam Microcontroller with Telegram Notification. *Journal of Telecommunication Network*. 2022. Vol. 12, No. 2. P. 1–10.

15. Lee J., Bagheri B., Kao H.A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*. 2015. Vol. 3. P. 18–23.

16. Humayed A., Lin J., Li F., Luo B. Cyber-Physical Systems Security – A Survey. *IEEE Internet of Things Journal*. 2017. Vol. 4, No. 6. P. 1802–1831.

17. Bagheri B., Yang S., Kao H.A., Lee J. Cyber-physical systems architecture for self-aware machines in Industry 4.0 environment. *IFAC-PapersOnLine*. 2015. Vol. 48, No. 3. P. 1622–1627.

18. Zacchia Lun Y., D'Innocenzo A., Malavolta I., Di Benedetto M.D. Cyber-Physical Systems Security: a Systematic Mapping Study. *arXiv preprint* arXiv:1605.09641. 2016.

19. Ara A., Al-Rodhaan M., Tian Y., Al-Dhelaan A. A secure service provisioning framework for cyber physical cloud computing systems. *arXiv preprint* arXiv:1611.00374. 2015.

20. Rubio-Hernan J., Sahay R., De Cicco L., Garcia-Alfaro J. Cyber-Physical Architecture Assisted by Programmable Networking. *arXiv preprint* arXiv:1802.02360. 2018.

21. Lee E.A. The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. *Sensors*. 2015. Vol. 15, No. 3. P. 4837–4869.

22. Gunes V., Peter S., Givargis T., Vahid F. A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems. *KSII Transactions on Internet and Information Systems*. 2014. Vol. 8, No. 12. P. 4242–4268.

23. Shi J., Wan J., Yan H., Suo H. A Survey of Cyber-Physical Systems. *International Journal of Computer Applications*. 2011. Vol. 55, No. 6. P. 57–62.

24. HIVEMQ, URL: https://www.hivemq.com/ (дата звернення 24.05.2025).

25. Rajkumar R., Lee I., Sha L., Stankovic J. Cyber-Physical Systems: The Next Computing Revolution. *Proceedings of the 47th Design Automation Conference*. 2010. P. 731–736.

26. Cyber-Physical Systems (CPS), URL: https://www.fortinet.com/lat/resources/cyberglossary/cyber-physical-systems (дата звернення 24.05.2025).

27. What are Cyber-Physical Systems? URL: https://www.otorio.com/resources/what-are-cyber-physical-systems/ (дата звернення 24.05.2025).

28. Introduction to Cyber-Physical System, URL: https://www.tutorialspoint.com/cyber-physical-system/cyber-physical-system-introduction.htm (дата звернення 24.05.2025).

29. ESP32 for IoT: A Complete Guide, URL: https://www.nabto.com/guide-to-iot-esp-32/ (дата звернення 24.05.2025).

30. Getting Started with Node-RED on Raspberry Pi, URL: https://randomnerdtutorials.com/getting-started-node-red-raspberry-pi/ (дата звернення 24.05.2025).

31. Turn your smartphone into an IoT device, URL: https://developer.ibm.com/tutorials/iot-mobile-phone-iot-device-bluemix-apps-trs/ (дата звернення 24.05.2025).

32. Build an Earthquake Early Warning (EEW) system and visualize historical seismic data sets, URL: https://developer.ibm.com/tutorials/build-an-openeew-earthquake-early-warning-node-red-dashboard/ (дата звернення 24.05.2025).

33. Hasan R., Khan M., Ashek A., Rumpa I. Microcontroller Based Home Security System with GSM Technology. *Open Journal of Safety Science and Technology*. 2015. Vol. 5, No. 2. P. 55–62.

34. Kowsalya R., Karthigha M. A Survey on Microcontroller Based Intelligent Irrigation System. *International Journal of Emerging Technology in Computer Science & Electronics*. 2015. Vol. 13, No. 2. P. 1–5.

35. Doliny J., Dostálek P., Vašek V. Microcontroller Software Library for Process Control. *WSEAS Transactions on Systems and Control*. 2015. Vol. 10. P. 233–240.

36. Haque U. Diversified Projects in Microcontroller Class Enhances Learning Experience. *Cedarville University Publications*. 2015. P. 1–6.

37. Khan S.R., Dristy F.S. Android Based Security and Home Automation System. *arXiv preprint arXiv*:1504.03564. 2015. P. 1–7.

38. Saini P., Bansal A., Sharma A. Time Critical Multitasking for Multicore Microcontroller using XMOS Kit. *arXiv preprint arXiv*:1504.02555. 2015. P. 1–6.

39. Moro N., Dehbaoui A., Heydemann K., Robisson B., Encrenaz E. Electromagnetic Fault Injection: Towards a Fault Model on a 32-bit Microcontroller. *arXiv preprint arXiv*:1402.6421. 2014. P. 1–10.

40. Hanafiah N., Surjarwo S., Wibisurya A. Door Security System for Home Monitoring Based on ESP32. *Procedia Computer Science*. 2019. Vol. 157. P. 673–682.

41. Kamat M.N., Shinde D. Smart Door Security Control System Using Raspberry Pi. *International Journal of Innovations & Advancement in Computer Science*. 2017. Vol. 6, No. 11. P. 1–4.

42. Kodali R.K., Jain V., Bose S., Boppana L. IoT Based Smart Security and Home Automation System. *International Conference on Computing, Communication and Automation (ICCCA)*. 2016. P. 1286–1289.

43. Getting Started with Wokwi: Arduino Simulation Made Easy, URL: https://maker.pro/arduino/tutorial/getting-started-with-wokwi-arduino-simulation-made-easy (дата звернення 24.05.2025).

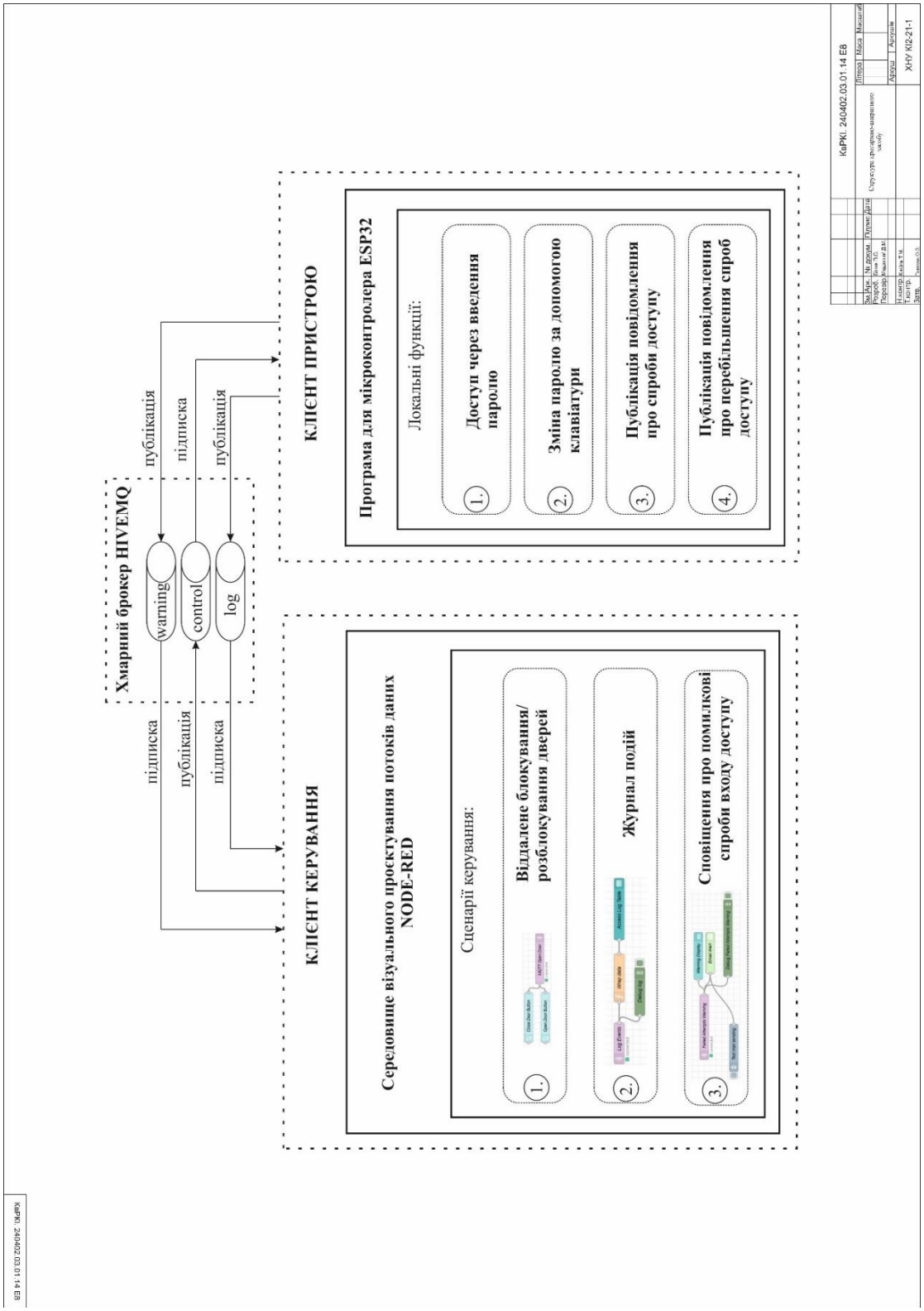
44. Мережі IoT: порівняння 4 типів і приклади використання, URL: https://e-server.com.ua/uk/poradi/merezi-iot-porivniannia-4-tipiv-i-prikladi-vikoristannia?srsltid=AfmBOorVcgfzvEkXsW0RFdh425GAI8Xf38rizsGJJkSVxDrn_g9X629O (дата звернення 24.05.2025).

45. Розробка програмного забезпечення. IoT, URL: https://startit.ua/rozrobka-programnogo-zabezpechenja-iot (дата звернення 24.05.2025).

ДОДАТОК А

(обов’язковий)

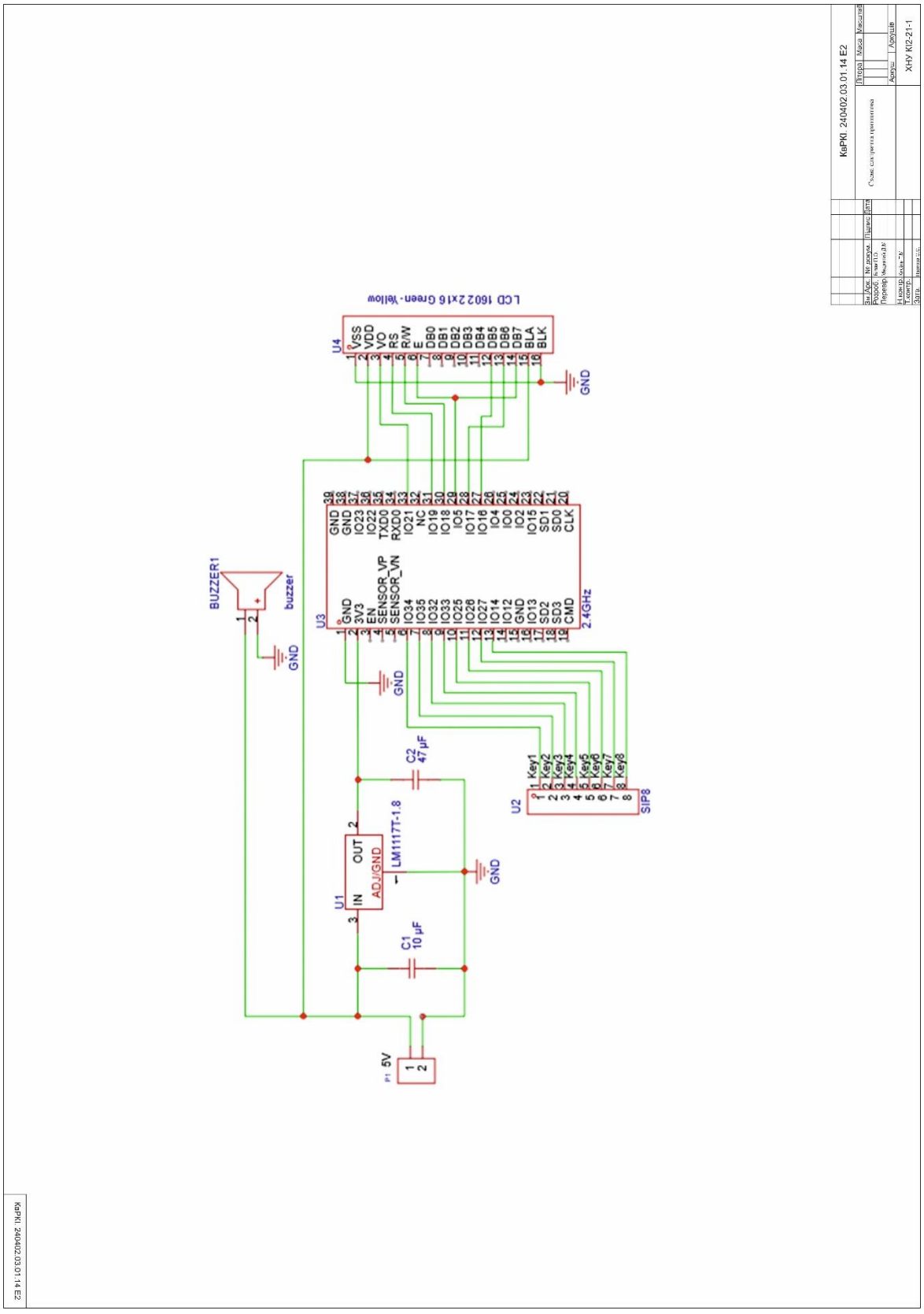
КОПІЯ КРЕСЛЕННЯ «СТРУКТУРА ПРОГАРМНО-АПАРАТНОГО ПРИСТРОЮ»



ДОДАТОК Б

(обов’язковий)

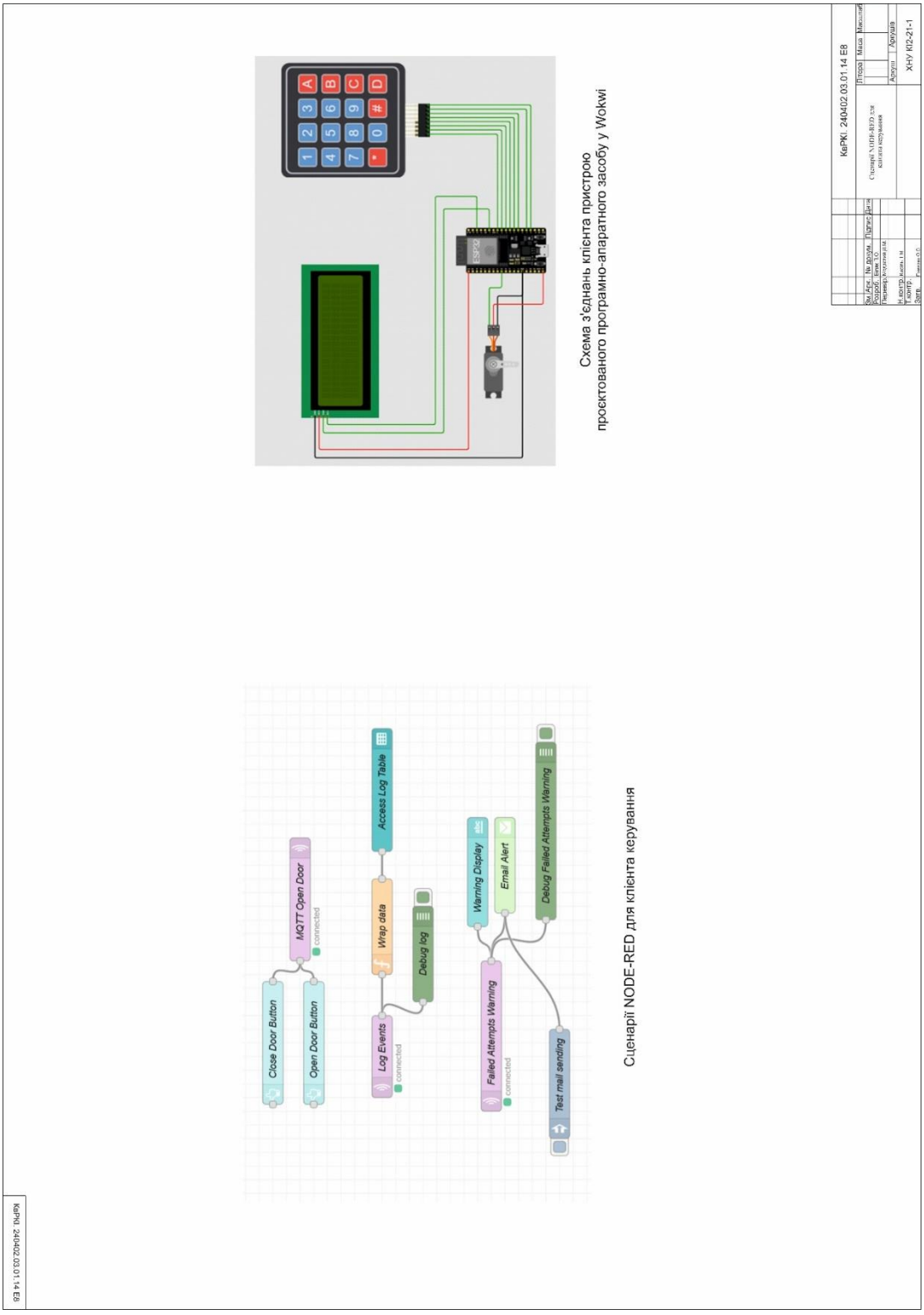
КОПІЯ КРЕСЛЕННЯ «СХЕМА ЕЛЕКТРИЧНА ПРИНЦИПОВА»



ДОДАТОК В

(обов’язковий)

КОПІЯ КРЕСЛЕННЯ «СЦЕНАРІЇ NODE-RED ДЛЯ КЛІЄНТА КЕРУВАННЯ»



Завідувачу кафедри КІІС
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Павла БІЛЯКА
ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

09.06.2025р.
дата

підпис

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 12%**

	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
ID: 244286 Title: БКР Програмно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32 Added in a DB: 2025-06-09 Authors: Павло БІЛЯК Heads: Дмитро МЕДЗАТИЙ Consultants: Opponents:	77850	548	1243 (2%)	20 (4%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

09.06.2025,

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32
Автор Павло БІЛЯК
Освітня програма Комп'ютерна інженерія та програмування
Рівень вищої освіти перший (бакалаврський) рівень
Спеціальність 123– Комп'ютерна інженерія
Науковий керівник: к.т.н., доцент Дмитро МЕДЗАТИЙ

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	Відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	Не виявлено

Підтвердження:
Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:
1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 5.62% і адресується до 30 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Дмитро МЕДЗАТИЙ
Гарант ОП _____ Сергій ЛИСЕНКО
Завідувач кафедри КІІС _____ Ольга ПАВЛОВА

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Павло БІЛЯК

Співавтор:

Назва: Біляк_Програмно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:5.6%

Коефіцієнт подібності 2:1.5%

Мікропробіли: 18

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-09 15:27:32.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-06-09 Доцент Андрій Нічепорук

Дата експерт

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Павло БІЛЯК

Тема: Програмно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 60

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано програмно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32

2. Висновок про відповідність роботи дипломному завданню Дипломний проект відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено огляд відомих систем керування доступом до кіберфізичної системи "розумний будинок". У другому розділі здійснено проєктування програмно-апаратного засобу до кіберфізичної системи "розумний будинок". У третьому розділі виконано тестування та реалізацію програмно-апаратного засобу до кіберфізичної системи "розумний будинок"

4. Позитивні сторони роботи: Спроєковано прогармно-апаратний засіб керування доступом до кіберфізичної системи "Розумний будинок" із функцією сповіщення на основі ESP32

5. Негативні сторони роботи: Варто було б доповнити функціонал спроєктованого засобу функцією зміни паролю не тільки через клавіатуру, а й через створену панель керування.

6. Оцінка графічного оформлення та пояснювальної записки роботи: пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на високому рівні.

8. Інші зауваження: -

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «відмінно» 4,75 (А)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Праворська Наталія Іванівна, доцент кафедри ПВЗ к. пед. наук

"09" 06 2025р.