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

КВАЛІФІКАЦІЙНА РОБОТА

Апаратно-програмний комплекс «Розумний сейф» з багатофакторною автентифікацією та захистом від несанкціонованого переміщення
Назва теми

Рівень вищої освіти *перший (бакалаврський)*

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *123 «Комп'ютерна інженерія»*
Шифр, назва

Освітня програма *«Комп'ютерна інженерія та програмування»*
Назва

Шифр КвРКІ 022010.22.01.61 ПЗ

Виконав здобувач IV курсу, група КІ2-22-1 — Підпис — Іван ВОВК (Ініціали, прізвище)

Керівник — Науковий ступінь, учене звання — Підпис — Олексій ЛИГУН (Ініціали, прізвище)

Нормоконтролер канд. фіз.-мат. наук, доцент — Науковий ступінь, учене звання — Підпис — Тетяна КИСІЛЬ (Ініціали, прізвище)

До захисту допускаю:
завідувач кафедри КІІС — Підпис — Ольга ПАВЛОВА (Ініціали, прізвище)
«01» червня 2026 р.
дата

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ
Завідувачка кафедри КІС
Ольга ПАВЛОВА

" 10 " 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Вовку Івану Івановичу
Прізвище, ім'я, по батькові студента

1. Тема проєкту (роботи) Апаратно-програмний комплекс «Розумний сейф» з багатофакторною автентифікацією та захистом від несанкціонованого переміщення
Керівник проєкту (роботи) Лигун Олексій Олегович
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7
2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.
3. Вихідні дані до роботи Завдання на кваліфікаційну роботу
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз відомих рішень та вибір стратегії і засобів для реалізації завдання
Проєктування архітектури апаратно-програмного комплексу «Розумний сейф»
Програмно-апаратна реалізація апаратно-програмного комплексу «Розумний сейф»
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
Структурна схема кіберфізичної системи
Алгоритми роботи ПЗ мікроконтролера
Схема електрична принципова

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прий

7. Дата видачі завдання «_10_»_____01_____2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проєкту (роботи)	Термін виконання етапів проєкту (роботи)	Приме
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	викона
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	викона
3	Робота над розділом 1 – аналіз відомих рішень та вибір стратегії і засобів для реалізації завдання	01.03.2026	викона
4	Робота над розділом 2 – проєктування архітектури апаратно-програмного комплексу «Розумний сейф»	01.04.2026	викона
5	Робота над розділом 3 – програмно-апаратна реалізація апаратно-програмного комплексу «Розумний сейф»	29.04.2026	викона
6	Оформлення пояснювальної записки згідно вимог	24.05.2026	викона
7	Попередній захист ВКР	25.05.2026	викона
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач _____ Іван ВОВК
Підпис Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи _____ Олексій ЛИГУН
Підпис Ім'я, ПРІЗВИЩЕ

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 022010.22.01.61 ПЗ	Пояснювальна записка	62		
2		КвРКІ 022010.22.01.61 Е8	Структурна схема кіберфізичної системи	1		
3		КвРКІ 022010.22.01.61 Е8	Алгоритми роботи ПЗ мікроконтролера	1		
4		КвРКІ 022010.22.01.61 Е8	Схема електрична принципова	1		

Зм	Арк	№ докум	Підпис	Дата	КвРКІ 022010.22.01.61 ВП		
Розробив		Вовк		01.06	Відомість проєкту	Літера	Аркуш 1 / Аркушів 1
Перевір.		Лигун		01.06		У	
Н. контр.		Кисіль		01.06	ХНУ, КІ2-22-1		
Затв.		Павлова		01.06			

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Апаратно-програмний комплекс «Розумний сейф» з багатофакторною автентифікацією та захистом від несанкціонованого переміщення».

Автор роботи: Іван ВОВК.

Керівник роботи: Олексій ЛИГУН.

Пояснювальна записка: 62 с., 25 рис., 4 табл., 4 дод., 52 джерел.

Графічна частина: 3 креслення.

ESP32, MQTT, БАГАТОФАКТОРНА АВТЕНТИФІКАЦІЯ, ІНЕРЦІАЛЬНИЙ МОНІТОРИНГ, ІНТЕРНЕТ РЕЧЕЙ, КІБЕРФІЗИЧНА СИСТЕМА.

Кваліфікаційна робота бакалавра присвячена розробці та дослідженню апаратно-програмного комплексу «Розумний сейф» з функціями багаторівневого контролю доступу та просторового моніторингу. Актуальність теми зумовлена необхідністю переходу від традиційних механічних засобів захисту до розподілених кіберфізичних систем. Це дає змогу нівелювати вразливості локальних електронних замків та забезпечити безперервний аудит безпеки.

Метою роботи є проєктування та практична реалізація інтелектуального пристрою, здатного функціонувати в середовищі Інтернету речей (IoT) для віддаленого керування та оперативного сповіщення про загрози на засадах парадигми нульової довіри (zero-trust). Для досягнення поставленої мети було розроблено трирівневу архітектуру системи та створено робочий макет на базі мікроконтролера ESP32. Програмно реалізовано алгоритм трифакторної автентифікації (PIN-код, RFID-картка, дистанційне підтвердження) та впроваджено підсистему виявлення фізичного переміщення за допомогою інерціального MEMS-датчика. В рамках роботи спроєктовано захищену логіку асинхронного обміну даними через брокер MQTT та розроблено мобільний застосунок для віддаленого адміністрування прав доступу. Тестування комплексу підтвердило його високу надійність та стійкість до несанкціонованого втручання.

Підпис здобувача

30.05.2026

Дата

ЗМІСТ

ВСТУП

Сучасні вимоги до безпеки матеріальних цінностей зумовлюють перехід від суто механічних засобів захисту до розподілених кіберфізичних систем. Традиційні сейфи та локальні електронні замки мають суттєвий недолік: вони функціонують як ізольовані об’єкти. У разі інтелектуального зламу, викрадення ідентифікатора або фізичного переміщення сховища, власник дізнається про інцидент лише постфактум.

Для забезпечення надійного захисту в умовах сучасних загроз виникає об’єктивна потреба в інтеграції фізичних запірних механізмів із технологіями Інтернету речей. Актуальним завданням є розробка систем, що базуються на концепції віддаленого контролю та нульової довіри. Створення комплексу «Розумний сейф», який поєднує багатофакторну автентифікацію з обов’язковою верифікацією людиною, безперервний просторовий моніторинг на базі MEMS-датчиків та захищений асинхронний обмін даними, дозволяє вирішити проблему сліпих зон традиційних систем безпеки. Це перетворює засіб зберігання з пасивного об’єкта на активний інтелектуальний вузол, здатний миттєво реагувати на загрози та забезпечувати повний аудит подій.

Метою кваліфікаційної роботи є проєктування, розробка та дослідження апаратно-програмного комплексу «Розумний сейф», що забезпечує високий рівень стійкості до несанкціонованого втручання завдяки використанню розподіленої архітектури, алгоритмів трифакторної автентифікації та механізмів інерціального моніторингу в реальному часі.

Об’єктом дослідження є процеси організації захищеного доступу, мережевої взаємодії та моніторингу фізичного стану в розподілених кіберфізичних системах.

Предметом дослідження є апаратно-програмні засоби, архітектурні рішення та алгоритми реалізації багаторівневої автентифікації та просторового моніторингу в апаратно-програмному комплексі «Розумний сейф».

1 АНАЛІЗ ВІДОМИХ РІШЕНЬ ТА ВИБІР СТРАТЕГІЇ І ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ЗАВДАННЯ

1.1 Огляд існуючих систем та відомих реалізацій сейфів

Традиційні механічні засоби зберігання матеріальних цінностей не забезпечують необхідного рівня оперативного моніторингу та контролю доступу. Сучасні вимоги до безпеки зумовлюють перехід до кіберфізичних систем (КФС), де надійність захисту залежить не лише від механічної міцності корпусу, а й від наявності багатофакторної автентифікації та криптографічної стійкості алгоритмів обробки даних. Проєктування такого апаратно-програмного комплексу вимагає інтеграції електромеханічних запірних пристроїв із високотехнологічними обчислювальними мережами.

На початковому етапі розвитку рівень безпеки визначався виключно механічною складністю конструкції. Домінували сувальдні та дискові замки. Сувальдні механізми (рисунок 1.1) засновані на використанні набору пластин (сувальд) із фігурними вирізами. Правильний ключ піднімає всі пластини на визначену висоту, дозволяючи ригелю рухатися.

Рисунок 1.1 – Сувальдний механізм [51]

Попри широке практичне використання, головним недоліком сувальдних механізмів є наявність прямого фізичного доступу до внутрішніх елементів конструкції через замкову щілину. Це зумовлює вразливість пристроїв до інтелектуального зламу за допомогою відмичок або ендоскопічного обладнання. Альтернативою виступали кодові дискові замки (рисунок 1.2), які працюють за принципом суміщення пазів на системі коаксіальних дисків.

Рисунок 1.2 – Кодовий дисковий замок [52]

Вони забезпечують вищу секретність, оскільки усувають проблему відкритої замкової щілини, проте залишаються вразливими до методів акустичного аналізу та рентгеноскопії. Головною ж проблемою всього механічного етапу є повна неможливість аудиту: такі системи не здатні фіксувати спроби несанкціонованого доступу, вести журнал подій чи дистанційно сповіщати власника про загрозу.

Традиційні системи зберігання, такі як сейфи компанії «Паритет-К» (рисунок 1.3) [44], історично зосереджені на забезпеченні максимальної фізичної стійкості металевого корпусу та замкових механізмів до механічного злому. Однак в умовах цифровізації суто фізичного захисту виявляється недостатньо для оперативного моніторингу та своєчасного реагування на інциденти.

Рисунок 1.3 – Сейф Griffon S.20.E виробництва компанії «Паритет-К» [44]

З розвитком мікроелектроніки розпочався електронний етап еволюції засобів зберігання: механічну автентифікацію замінили мікроконтролери, а фізичний ключ – PIN-код або RFID-картка [33]. Блокування ригеля в таких системах найчастіше реалізується за допомогою соленоїдів, керованих електронною схемою. Хоча такі рішення частково вирішили проблему аудиту подій доступу, вони принесли нові вразливості. Найбільш поширеною є фізична інерційність підпружиненого соленоїда: різкий удар по корпусу в момент повороту ручки може призвести до відкриття дверцят. Водночас, наявність сервісних роз'ємів або додаткових ключів часто стає слабкою ланкою для зловмисників [12].

Паралельно з цим, передові системи безпеки, такі як рішення від Ajax Systems [3], ілюструють парадигму переходу до розподіленої архітектури Інтернету речей (IoT), де основою є використання бездротових датчиків (рисунок 1.4), постійний зв'язок з хмарним сервером та дистанційне керування. Інтеграція подібних підходів у концепцію «розумного сейфа» дозволяє суттєво розширити його функціонал, перетворюючи пасивний об'єкт на активний вузол мережі безпеки.

Рисунок 1.4 – Датчик Ajax DoorProtect Plus [3]

Попри розвиток технологій розумних замків, актуальні дослідження кібербезпеки виявляють значні концептуальні та архітектурні вразливості в існуючих реалізаціях [12]. Комплексна оцінка ризиків використання бездротових протоколів у системах розумного дому та IoT підтверджує високу ймовірність компрометації пристроїв у разі використання недостатньо захищених каналів зв’язку або слабких методів ідентифікації [24, 25].

Для подолання цих недоліків сучасні розробники звертаються до впровадження багаторівневих систем перевірки [50]. Зокрема, пропонується використання систем блокування на базі технології RFID [33, 42], а також розробка ефективних протоколів багатофакторної автентифікації, адаптованих для умов розумного середовища [4, 34].

Практичні реалізації подібних апаратно-програмних комплексів все частіше будуються на базі енергоефективних мікроконтролерів, зокрема платформ сімейства ESP32, які дозволяють створювати легковагові, але функціональні IoT-шлюзи [36], а також забезпечувати апаратну безпеки для ідентифікації пристроїв в Інтернеті речей [14]. Аналіз відомих рішень показує, що хоча питання фізичної оболонки класичних сейфів вирішено на високому

рівні, логіка апаратно-програмної взаємодії прототипів часто потребує оптимізації [12, 19].

Існує об’єктивна потреба в розробці комплексів із інтеграцією датчиків просторового переміщення для захисту від крадіжки самого сейфа [9], а також із впровадженням суворих політик керування – зокрема, ізоляції системних функцій від локальних фізичних інтерфейсів на користь виключно авторизованого віддаленого каналу зв’язку.

1.2 Аналіз стратегій багатофакторної автентифікації в розподілених середовищах

На відміну від класичних інформаційних систем, IoT-пристрої часто функціонують у середовищах з підвищеним ризиком фізичного доступу зловмисників та характеризуються обмеженими обчислювальними ресурсами, що вимагає розробки так званих легковагових протоколів автентифікації [13, 25, 47]. Впровадження концепції багатофакторної автентифікації (multi-factor authentication, MFA) забезпечує надійність сучасних кіберфізичних систем та Інтернету речей (IoT). Додатковим викликом є гетерогенність IoT-середовища: пристрої різних виробників, що працюють під управлінням різних операційних систем та протоколів, мають бути здатні до безпечної взаємодії в єдиній мережі [21].

Традиційно фактори ідентифікації користувача поділяються на три основні категорії: фактор знання (наприклад, PIN-код або пароль), фактор володіння (апаратні токени, смарт-картки, RFID-мітки) та фактор властивості (біометричні дані користувача) [4, 5, 48]. Ефективна система безпеки повинна поєднувати декілька факторів з різних категорій, щоб гарантувати, що компрометація одного з них не призведе до повного зламу системи. Це положення є основоположним принципом побудови сучасних протоколів контролю доступу [34].

Біометричні методи, такі як розпізнавання обличчя, сканування відбитків пальців або безперервний поведінковий аналіз, сьогодні активно досліджуються та демонструють високий рівень захисту від спуфінгу [1, 7]. Проте їхня інтеграція в IoT-системи стикається з низкою перешкод. Обробка складних алгоритмів або нейронних мереж для розпізнавання мультимодальної біометрії вимагає значних обчислювальних потужностей, що часто змушує переносити ці завдання на рівень граничних обчислень (edge computing) [31, 32, 49]. Застосування біометрії в ресурсообмежених пристроях викликає додаткові виклики щодо збереження конфіденційності даних та захисту шаблонів від перехоплення – зокрема, активно досліджуються підходи на базі технології Fuzzy Vault для приватного зберігання біометричних шаблонів [43].

Альтернативним та більш оптимізованим підходом для кінцевих IoT-вузлів є використання технології радіочастотної ідентифікації (RFID) як надійного фактора володіння. Сучасні дослідження підтверджують ефективність RFID-систем для побудови розумних електронних замків, особливо за умови застосування протоколів із захистом від клонування міток та збереженням приватності [33, 42]. Для ресурсообмежених IoT-середовищ розроблено спеціалізовані постквантові протоколи автентифікації RFID-пристроїв, що забезпечують стійкість до атак навіть за умов обмеженої обчислювальної потужності вузла [41]. Водночас використання лише одного фізичного ідентифікатора залишає систему вразливою до його крадіжки, що підкреслює необхідність комбінування факторів.

Для досягнення максимального рівня безпеки в системах контролю доступу дослідники пропонують багаторівневі (multi-tiered) фреймворки, що інтегрують апаратні модулі з хмарними обчисленнями [21]. Практичні реалізації доводять ефективність трифакторної автентифікації при побудові розумних систем блокування [30, 34]. У контексті розробки кіберфізичної системи безпеки найоптимальнішою стратегією є послідовне застосування фактора знання (введення PIN-коду), перевірки фізичного фактора володіння (RFID-

ідентифікатор) та фінального логічного підтвердження легітимності дій через віддалений авторизований клієнт – мобільний застосунок [34]. Такий підхід усуває недоліки виключно локальної перевірки, унеможливлює обхід захисту через фізичне втручання та забезпечує гнучкий, стійкий до апаратного втручання механізм контролю [12, 19].

1.3 Методи та технології фіксації переміщення у просторі

Для забезпечення комплексного захисту кіберфізичної системи недостатньо лише контролювати доступ до її вмісту; одним з векторів атак залишається фізичне викрадення самого об’єкта охорони. У зв’язку з цим, концепція нульової довіри (zero-trust), яка все частіше застосовується при проєктуванні кіберфізичних систем [19, 28], вимагає постійного моніторингу не лише логічного, але й фізичного стану пристрою. Відповідно, сучасні системи безпеки обов’язково включають підсистеми просторового моніторингу для фіксації несанкціонованого переміщення або вібраційного впливу, що свідчить про спроби механічного злому [26].

Традиційним методом відстеження переміщення є використання систем глобального позиціонування (GPS). Хоча GPS-трекери є ефективним інструментом для логістичних завдань та пошуку викрадених об’єктів на відкритому просторі, їх застосування як первинного тригера тривоги для сейфів має суттєві обмеження. Всередині будівель, а особливо в екранованих металевих корпусах, супутниковий сигнал зазнає сильного загасання або повністю зникає. Ці модулі характеризуються високим енергоспоживанням та затримкою у визначенні початкових координат, що робить їх недостатньо реактивними для миттєвої фіксації початку крадіжки [10].

Більш надійним, швидким та енергоефективним підходом для локального виявлення руху є використання мікроелектромеханічних систем (MEMS) – просторових інерційних датчиків. Практичні дослідження підтверджують

доцільність та високу надійність використання мікромеханічних акселерометрів для захисту розумних пристроїв та виявлення несанкціонованих фізичних маніпуляцій [9].

Акселерометри здатні вимірювати проекції вектора статичного прискорення вільного падіння, що дозволяє системі з високою точністю визначати поточний кут нахилу об’єкта відносно поверхні землі. Гіроскопи, у свою чергу, вимірюють кутову швидкість, дозволяючи реєструвати обертання об’єкта навколо своїх осей у просторі.

Для мінімізації похибок (дрейфу нуля) та шумів, що властиві окремим типам датчиків, у сучасних комплексах використовують інтегральні модулі з шістьма ступенями свободи. Вони об’єднують триосьовий акселерометр та триосьовий гіроскоп на одному кристалі. Безперервний потік просторових даних обробляється мікроконтролером у режимі реального часу, що відповідає парадигмі граничних обчислень, де виявлення аномалій відбувається безпосередньо на кінцевому вузлі Інтернету речей без необхідності постійної передачі сирих даних на сервер [32, 38].

Застосування алгоритмів виявлення аномалій у фізичних показниках системи дозволяє миттєво ідентифікувати загрозу ще до того, як зловмисник спробує отримати доступ до інтерфейсів авторизації або пошкодити корпус [20, 40]. Використання гібридних інерційних MEMS-датчиків є найбільш оптимальною стратегією для реалізації підсистеми захисту від переміщення в апаратно-програмному комплексі розумного сейфа, адже це забезпечує можливість миттєвого генерування сигналу тривоги при будь-якій зміні просторового положення пристрою [9, 26].

1.4 Аналіз мережевих протоколів для IoT-систем

Для забезпечення безперервного моніторингу та оперативного реагування на інциденти, кіберфізична система розумного сейфа потребує надійного та

захищеного каналу зв’язку між кінцевим апаратним вузлом та сервером керування. Специфіка Інтернету речей полягає в тому, що кінцеві пристрої, як правило, функціонують в умовах обмежених обчислювальних ресурсів та енергоспоживання, що вимагає впровадження так званих легковагових протоколів передачі даних [13]. Окрім продуктивності, вибір протоколу безпосередньо визначає захищеність усього каналу зв’язку: некоректно налаштований або апріорі незахищений протокол стає точкою входу для зловмисника [25].

Практичні реалізації IoT-шлюзів та систем безпеки на базі мікроконтролерів демонструють необхідність оптимізації мережевого трафіку для уникнення затримок під час передачі сигналів тривоги [36]. Традиційні веб-протоколи часто виявляються неефективними у таких сценаріях. Для вибору оптимальної стратегії зв’язку проведено порівняльний аналіз трьох найбільш поширених архітектурних рішень: HTTP/HTTPS, CoAP та MQTT.

HTTP/HTTPS (HyperText Transfer Protocol) працює за синхронною моделлю «клієнт-сервер». Хоча застосування сучасних стандартів шифрування (наприклад, TLS 1.3 [22]) забезпечує високий рівень безпеки передачі даних, архітектура HTTP має суттєві обмеження при розгортанні в IoT-мережах: великий розмір заголовків та необхідність встановлення нового з’єднання для кожного запиту. Це призводить до надмірної витрати ресурсів мікроконтролера та унеможливлює миттєву двосторонню комунікацію без постійного опитування сервера (polling), що є суттєвим недоліком для систем сигналізації [13].

CoAP (Constrained Application Protocol) розроблений спеціально для обмежених вузлів та мереж. Він використовує протокол UDP, що значно зменшує накладні витрати на передачу пакетів. Однак використання UDP знижує загальну надійність доставки повідомлень, оскільки не гарантує їх отримання адресатом, що є неприпустимим для трансляції подій безпеки розумного сейфа [25].

MQTT (Message Queuing Telemetry Transport) – це асинхронний протокол обміну даними, що базується на моделі «видавець-підписник» (publish/subscribe) і працює поверх TCP/IP. Центральним елементом архітектури є брокер

повідомлень, який маршрутизує дані між пристроями за визначеними темами (топіками). MQTT розроблявся спеціально для умов з нестабільним зв'язком та обмеженою пропускною здатністю [36]. Перевагою протоколу є підтримка рівнів якості обслуговування (Quality of Service, QoS), що дозволяє гарантувати доставку важливих повідомлень. З точки зору безпеки, MQTT-брокер виступає централізованим вузлом довіри: коректне розмежування прав доступу до топіків та обов'язкова автентифікація клієнтів на рівні брокера є необхідними умовами для унеможливлення несанкціонованого підписання на канали даних системи [34].

Для наочності порівняння основних характеристик протоколів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз мережевих протоколів для IoT систем [13, 22, 25]

Характеристика	HTTP / HTTPS	CoAP	MQTT
Архітектурна модель	Клієнт-серверна	Клієнт-серверна	«Видавець-підписник»
Протокол транспортного рівня	TCP	UDP	TCP
Розмір заголовку	Великий (від десятків байт до кілобайт)	Дуже малий (4 байти)	Малий (від 2 байт)
Синхронність	Синхронний	Асинхронний / Синхронний	Асинхронний
Гарантія доставки	Відсутня на рівні протоколу	Базова (confirmable messages)	Три рівні: QoS 0, QoS 1, QoS 2

Кінець таблиці 1.1

Характеристика	HTTP / HTTPS	CoAP	MQTT
Підтримка постійного з’єднання	Ні	Ні	Так
Придатність для систем тривоги	Низька (затримки через polling)	Середня (ризик втрати пакетів)	Висока (миттєве push-сповіщення)

З огляду на вимоги до розроблюваного апаратно-програмного комплексу, протокол MQTT є найбільш доцільним вибором. Його архітектура ідеально підходить для інтеграції апаратного макета на базі мікроконтролера, хмарного бекенду та клієнтського застосунку [36]. Брокер повідомлень дозволяє ефективно ізолювати підсистеми: мікроконтролер публікує дані про стан датчиків у відповідні топіки, а бекенд-служба миттєво обробляє їх та ретранслює на мобільний клієнт. Використання захищеного з’єднання MQTTS (MQTT over TLS) у поєднанні з багатофакторною автентифікацією забезпечує комплексний енергоефективний фреймворк безпеки, що відповідає сучасним вимогам до кіберфізичних систем [25, 34].

1.5 Постановка задачі

На основі проведеного аналізу сучасних рішень у сфері Інтернету речей та вивчення методів багатофакторної автентифікації [4, 34], метою даної роботи є проєктування та дослідження апаратно-програмного комплексу «Розумний сейф» з багаторівневим захистом. Оскільки головним завданням є розробка алгоритмів безпеки та логіки мережевої взаємодії, практична реалізація виконуватиметься у вигляді функціонального макета, зібраного на макетній платі.

Для досягнення поставленої мети необхідно вирішити низку завдань. По-перше, вимагається реалізація трифакторної автентифікації, яка поєднує

послідовне введення числового пароля, зчитування RFID-мітки та отримання підтвердження від користувача через віддалений інтерфейс [33, 42]. Такий підхід дозволяє нівелювати вразливості окремих факторів доступу. По-друге, необхідно створити підсистему просторового моніторингу на основі інерційних MEMS-датчиків для миттєвого виявлення переміщення або нахилу пристрою [9]. По-третє, потребує розробки розподілена архітектура зв'язку на базі протоколу MQTT, що забезпечить надійний обмін даними між апаратним вузлом на базі ESP32 та хмарними сервісами [25, 36].

Програмна логіка має гарантувати апаратну стійкість до саботажу: деактивація режиму тривоги повинна бути технічно неможливою через локальні органи вводу і здійснюватися виключно через мобільний застосунок [19]. Таким чином, об'єктом розробки виступають процеси захищеної взаємодії в кіберфізичних системах, а результатом має стати працездатний прототип, що демонструє ефективність обраних алгоритмів.

1.6 Висновки до першого розділу

У першому розділі було проведено аналіз сучасного стану технологій у сфері фізичної та кібербезпеки, що дозволило сформувати теоретичне підґрунтя для розробки комплексу. Огляд існуючих рішень показав, що сучасні системи безпеки активно еволюціонують у бік розподілених кіберфізичних систем, де традиційні методи фізичного захисту доповнюються активними засобами моніторингу та віддаленого керування через IoT-мережі.

Дослідження теорії багатофакторної автентифікації підтвердило, що для забезпечення високого рівня надійності в ресурсообмежених системах доцільним є поєднання різних типів ідентифікаторів. Обґрунтовано використання трифакторної моделі доступу, яка інтегрує введення пароля, зчитування RFID-мітки та фінальну верифікацію через мобільний застосунок. Порівняльний аналіз засобів фіксації переміщення дозволив встановити, що для роботи всередині

приміщень та в умовах металевого екранування найбільш ефективними є інерційні MEMS-датчики, які забезпечують миттєву реакцію на зміну положення пристрою.

Встановлено, що MQTT є оптимальним стандартом для взаємодії компонентів комплексу завдяки його асинхронній моделі та низьким накладним витратам. У підсумку сформовано стратегію реалізації та поставлено технічні завдання на проєктування, де головним аспектом визначено апаратну ізоляцію системних функцій. Зокрема, деактивація тривоги має бути можливою лише через довірений віддалений інтерфейс. Отримані результати аналізу є основою для подальшої розробки архітектури системи та її програмного забезпечення.

2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ «РОЗУМНИЙ СЕЙФ»

2.1 Проєктування архітектури системи

Проєктований апаратно-програмний комплекс базується на архітектурі розподіленої кіберфізичної системи, де логіка управління та захисту децентралізована між локальним інтелектуальним вузлом, хмарним брокером повідомлень та віддаленим клієнтським інтерфейсом. Такий підхід відповідає сучасним концепціям побудови систем безпеки, що вимагають не лише локальної стійкості, але й інтеграції в глобальну інформаційну інфраструктуру для забезпечення безперервного моніторингу та верифікації подій у реальному часі [19, 26]. Для наочного представлення логічних зв’язків та архітектурних рівнів розроблюваного комплексу розроблено структурну схему (рисунок 2.1).

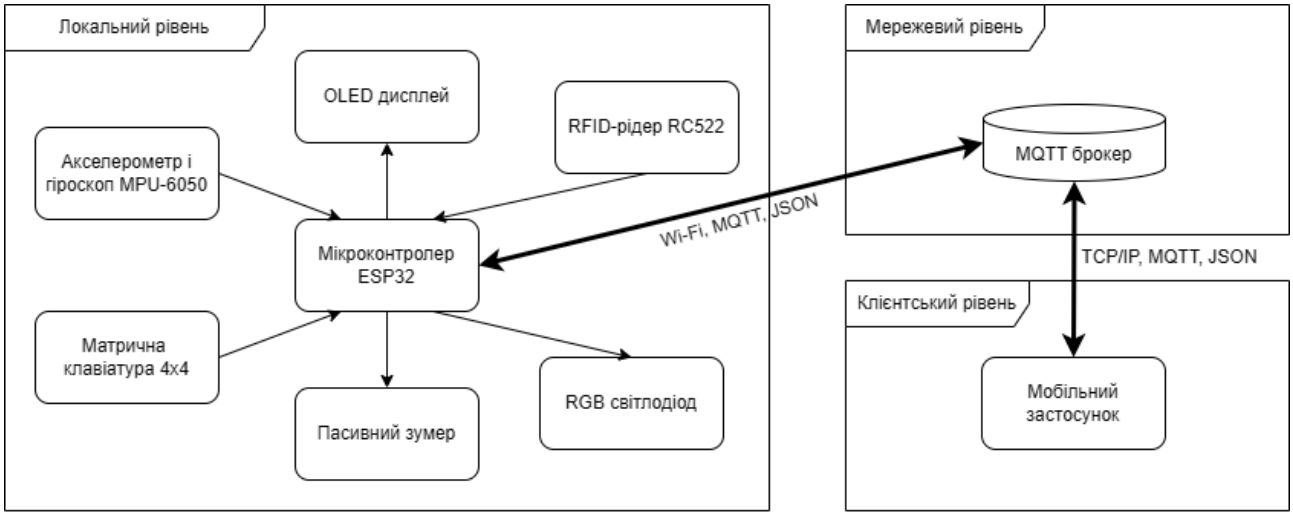


Рисунок 2.1 – Структурна схема апаратно-програмного комплексу

Як видно з наведеної схеми, архітектура має трирівневу топологію. Перший рівень (локальний) представлений граничним вузлом на базі ESP32, який безпосередньо взаємодіє з фізичними модулями: отримує дані від

інерціальної підсистеми (MPU-6050) та засобів ідентифікації, а також керує модулями зворотного зв’язку (OLED-дисплей, RGB світлодіод, зумер).

Другий рівень (мережевий) виконує роль посередника та базується на хмарному MQTT-брокері. Він забезпечує асинхронний обмін структурованими повідомленнями у форматі JSON, що дозволяє ізолювати апаратну частину від прямого доступу з глобальної мережі.

Третій рівень (клієнтський) – це мобільний застосунок, який виступає кінцевим терміналом для користувача, дозволяючи отримувати всі необхідні сповіщення про стан безпеки та здійснювати фінальну верифікацію запитів на доступ.

Основою архітектури є реалізація скінченного автомата станів (finite state machine, FSM), що керує переходами між режимами очікування, багатофакторної перевірки та тривоги. Локальний рівень системи, побудований на базі мікроконтролера ESP32, виступає як основних вузол обробки даних, що забезпечує мінімальні затримки при взаємодії з периферійними пристроями через інтерфейси I2C та SPI. Використання мікроконтролера з двоядерною архітектурою дозволяє розділити обчислювальні завдання на рівні прошивки: одне ядро фокусується на підтриманні стеку TCP/IP та MQTT-з’єднання, тоді як інше забезпечує безперервне опитування датчиків інерціального моніторингу та зчитувачів ідентифікаторів [36].

Архітектурною особливістю системи є інтеграція концепції граничних обчислень, де первинний аналіз аномалій просторового положення та валідація локальних факторів доступу відбувається безпосередньо на пристрої. Це мінімізує обсяг трафіку, що передається в мережу, і забезпечує працездатність базових захисних функцій навіть за умов тимчасової відсутності зв’язку з сервером. Комунікаційний рівень системи базується на протоколі MQTT, який через механізм публікацій та підписок забезпечує асинхронну взаємодію між компонентами. Використання формату JSON для обміну повідомленнями

дозволяє передавати структуровані дані про стан системи, UID RFID-міток та команди управління, забезпечуючи гнучкість і масштабованість комплексу.

Центральним елементом безпекової політики в архітектурі системи є винесення фінального центру прийняття рішень за межі фізичного доступу до пристрою. У розроблюваному комплексі реалізовано логіку, де апаратна частина виступає ініціатором запиту на доступ, тоді як право на остаточне розблокування або деактивацію тривоги делеговано віддаленому авторизованому клієнту. Така топологія унеможливлює обхід захисту шляхом прямого фізичного втручання в компоненти макета, оскільки керуючі сигнали надходять виключно через захищений канал зв’язку з хмарного брокера [19].

Енергонезалежність та цілісність конфігураційних даних забезпечується використанням спеціалізованих абстракцій пам’яті (Preferences), що дозволяє системі зберігати PIN-код та історію станів у внутрішній Flash-пам’яті мікроконтролера. Це гарантує швидке відновлення працездатності комплексу після скидання живлення або перезавантаження, підтримуючи високий рівень доступності системи як одного з основних вузлів кіберфізичної безпеки.

2.2 Вибір та обґрунтування апаратних компонентів

Вибір мікроконтролерної плати ESP32 DevKit v1 (рисунок 2.2) як центрального обчислювального вузла апаратно-програмного комплексу обумовлений не лише його високою продуктивністю, а й специфічною архітектурою. Цю архітектуру зручно використовувавати для розробки складних кіберфізичних систем із розподіленою логікою управління. Основою плати є високоефективний модуль ESP32-WROOM-32, який об’єднує в собі систему на кристалі (SoC), пам’ять та засоби бездротового зв’язку.

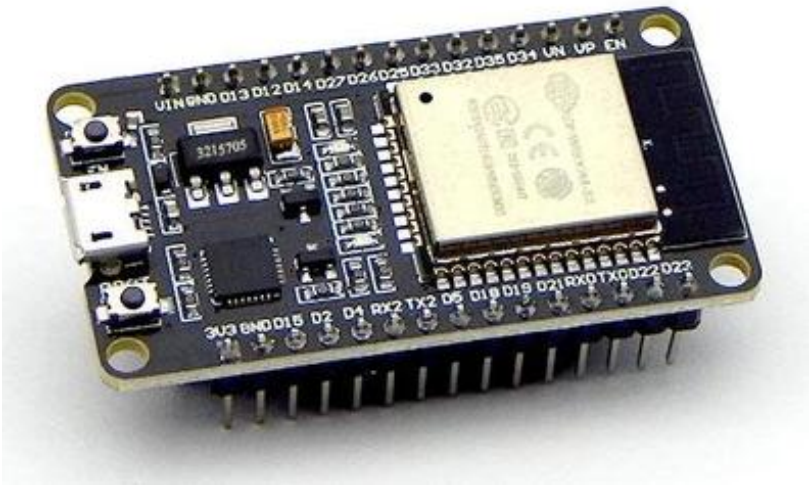


Рисунок 2.2 – Мікроконтролерна плата ESP32 DevKit v1 [45]

Основною модуля є два тридцятидвохбітні ядра Xtensa Dual-Core LX6, здатні функціонувати на тактовій частоті до 240 МГц. Така дворівнева обчислювальна структура дозволяє реалізувати справжній паралелізм на рівні прошивки пристрою. У контексті проєктування «Розумного сейфа» це дозволяє виділити одне ядро виключно для обслуговування мережевого стека та підтримки стабільного MQTT-з'єднання з хмарною інфраструктурою, тоді як друге ядро забезпечує безперервний інерціальний моніторинг просторового стану та обробку сигналів від підсистеми ідентифікації. Висока швидкодія процесора гарантує мінімальну затримку при обробці алгоритмів трифакторної автентифікації, що включають зчитування RFID-міток та валідацію PIN-коду.

Наявність 520 КБ вбудованої оперативної пам'яті (SRAM) забезпечує достатній ресурс для розгортання складних програмних структур, таких як буфери для JSON-пакетів та стеки для асинхронних функцій зворотного виклику. Для довготривалого зберігання конфігураційних даних, зокрема паролів та ідентифікаторів користувачів, модуль оснащений 4 МБ зовнішньої Flash-пам'яті, доступ до якої здійснюється через оптимізовану бібліотеку Preferences. Це дозволяє реалізувати надійне збереження стану системи навіть за умов повного вимкнення живлення.

Інтегрований радіочастотний блок підтримує стандарти Wi-Fi 802.11 b/g/n та Bluetooth (Classic/BLE), що робить ESP32 універсальним IoT-шлюзом. Вбудована антена та підсилювач сигналу забезпечують стабільну взаємодію з локальною мережею без необхідності використання зовнішніх активних компонентів. Багатий набір периферійних інтерфейсів, включаючи апаратні модулі SPI, I2C та UART, дозволяє підключати периферію датчиків без програмної емуляції протоколів, що суттєво підвищує надійність системи та знижує навантаження на основні обчислювальні ядра.

Завдяки поєднанню високої енергоефективності у режимах глибокого сну та значної пікової потужності, дана платформа є найбільш оптимальним рішенням для створення автономного інтелектуального вузла, здатного функціонувати в умовах обмежених ресурсів живлення при збереженні високого рівня захищеності.

Підсистема ідентифікації користувача використовує RFID-зчитувач RC522 (рисунок 2.3), який функціонує в діапазоні 13,56 МГц. Цей модуль побудований на базі мікросхеми MFRC522 та підтримує стандарти ISO/IEC 14443 A. Вибір даного компонента обумовлений його високою завадостійкістю та можливістю безконтактного зчитування даних на відстані до 50 мм, що робить його ідеальним для реалізації фактора володіння.

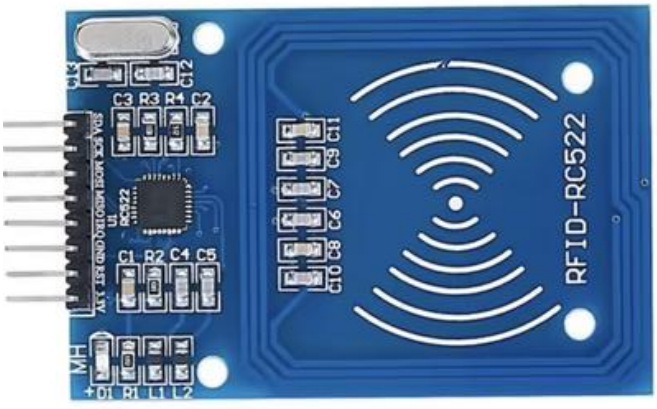


Рисунок 2.3 – RFID-зчитувач RC522 [46]

Як фізичний фактор знання застосовано шістнадцятикнопкову мембранну клавіатуру, організовану за матричною схемою 4x4. Це дозволяє економити піни мікроконтролера, використовуючи лише вісім ліній вводу-виводу для обробки натискань шістнадцяти окремих клавіш, що забезпечує надійне введення PIN-коду.

Рисунок 2.4 – Шістнадцятикнопкова мембранна клавіатура [46]

Захист від фізичного втручання та переміщення сейфа реалізовано за допомогою інтегрального датчика MPU-6050 (рисунок 2.5). Цей модуль містить трьохосьовий акселерометр та трьохосьовий гіроскоп (6DOF) на одному кристалі. Використання даного датчика дозволяє системі не лише фіксувати факт переміщення у просторі, але й визначати кут нахилу макета з високою точністю завдяки вбудованому 16-бітному аналого-цифровому перетворювачу для кожного каналу.

Рисунок 2.5 – Акселерометр і гіроскоп MPU-6050 [46]

Для візуального відображення інформації обрано OLED-дисплей діагоналлю 0,91 дюйма з роздільною здатністю 128x32 пікселів (рисунок 2.6). На відміну від рідкокристалічних аналогів, OLED-технологія забезпечує вищий контраст і менше енергоспоживання, оскільки кожен піксель є самовипромінювальним джерелом світла. Керування дисплеєм здійснюється через інтерфейс I2C, що мінімізує кількість з’єднань.

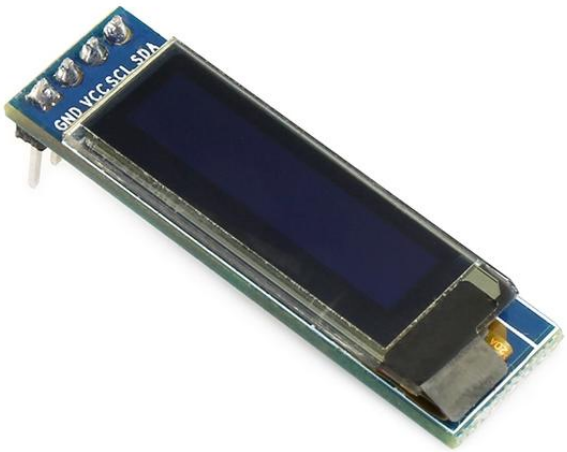


Рисунок 2.6 – OLED-дисплей [46]

Світлову сигналізацію реалізовано за допомогою модуля RGB-світлодіода KY-016 (рисунок 2.7), який дозволяє формувати довільний колір шляхом змішування трьох базових каналів за допомогою широтно-імпульсної модуляції

(PWM). Звукове сповіщення забезпечує пасивний зумер (рисунок 2.8), здатний генерувати сигнал тривоги достатньої гучності при подачі керуючого сигналу.

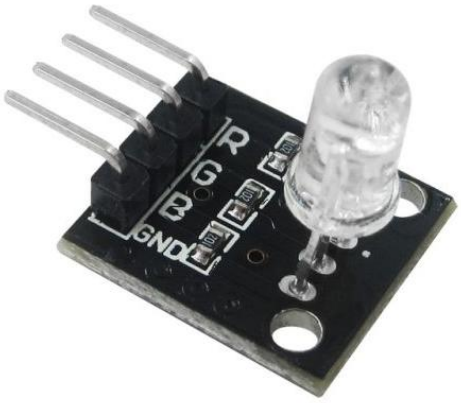


Рисунок 2.7 – Модуль RGB-світлодіода KY-016 [46]

Рисунок 2.8 – Пасивний зумер [46]

Оскільки комплекс реалізується у форматі експериментального прототипу, для побудови електричної схеми використано безпаєчну макетну плату MB-102 (рисунок 2.9) та набір перемичок різного типу. Це рішення дозволяє оперативно змінювати конфігурацію та додавати нові модулі без проведення паяльних робіт. Програмне забезпечення для енергонезалежного зберігання конфігураційних даних базується на бібліотеці Preferences, яка задіює вбудовану Flash-пам’ять мікроконтролера, гарантуючи цілісність коду доступу навіть у разі повного вимкнення живлення.

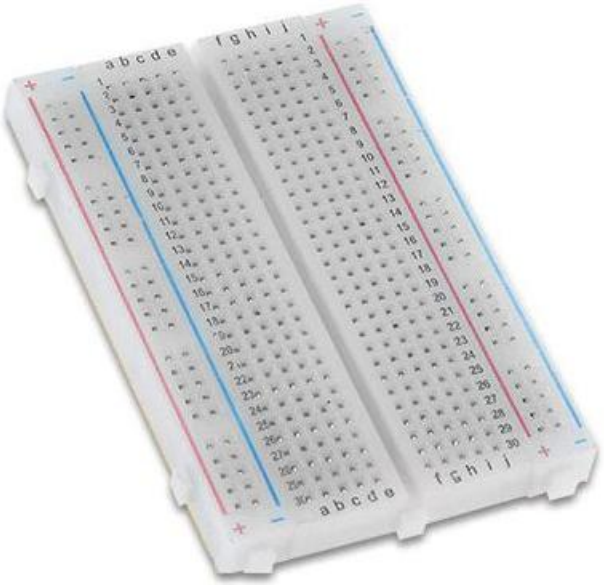


Рисунок 2.9 – Макетна плата безпаєчна MB-102 [46]

Оскільки ядром системи є мікроконтролер ESP32, який має обмежену кількість контактів вводу-виводу (GPIO) трасування сигнальних ліній необхідно проводити найоптимальнішим шляхом для уникнення апаратних конфліктів. Початкова валідація електричних з'єднань та тестування логіки взаємодії компонентів здійснювалися у середовищі симуляції Wokwi, що дозволило перевірити розподіл виводів та усунути потенційні колізії на апаратному рівні ще до етапу фізичної збірки макета.

Периферійні пристрої підключені з використанням двох основних цифрових шин: I2C (Inter-Integrated Circuit) та SPI (Serial Peripheral Interface). Шина I2C використовується для паралельного підключення OLED-дисплея та інерціального модуля MPU-6050. Завдяки адресній архітектурі протоколу I2C, обидва модулі функціонують на одній парі контактів мікроконтролера (SDA та SCL), маючи унікальні шістнадцяткові адреси.

Зчитувач RFID-міток RC522 вимагає вищої швидкості передачі даних для мінімізації затримок під час автентифікації, тому його інтегровано через шину SPI. Клавіатура формату 4x4 підключена безпосередньо до вільних цифрових

виводів ESP32, які налаштовані в режимі підтягуючих резисторів (PULL_UP). Розподіл контактів мікроконтролера наведено у таблиці 2.1.

Таблиця 2.1 – Розподіл контактів мікроконтролера

Компонент	Інтерфейс (тип сигналу)	Вивід компонента	Контакт мікроконтролера
Акселерометр і гіроскоп MPU-6050	I2C	SDA SCL	GPIO21 GPIO22
OLED-дисплей	I2C	SDA SCL	GPIO21 GPIO22
Модуль RGB-світлодіода KY-016	ШІМ	R G B	GPIO4 GPIO2 GPIO15
RFID-зчитувач RC522	SPI	MOSI MISO SCK CS	GPIO23 GPIO19 GPIO18 GPIO5
Матрична клавіатура 4x4	Цифровий	R1 R2 R3 R4 C1 C2 C3 C4	GPIO32 GPIO33 GPIO25 GPIO26 GPIO27 GPIO14 GPIO12 GPIO13
Пасивний зумер	Цифровий	Signal	GPIO16

Такий розподіл ресурсів мікроконтролера залишає резервні виводи для можливого майбутнього масштабування системи, забезпечує відсутність

апаратних конфліктів між підсистемами та створює технічну основу для реалізації паралельного виконання завдань, описаного у наступному підрозділі.

На основі розробленого розподілу ресурсів мікроконтролера було спроєктовано повну електричну принципову схему (рисунок 2.10), яка відображає фізичну топологію макета. Процес трасування сигнальних ліній враховує не лише кількісні обмеження виводів ESP32, а й специфіку електричних характеристик кожного модуля.

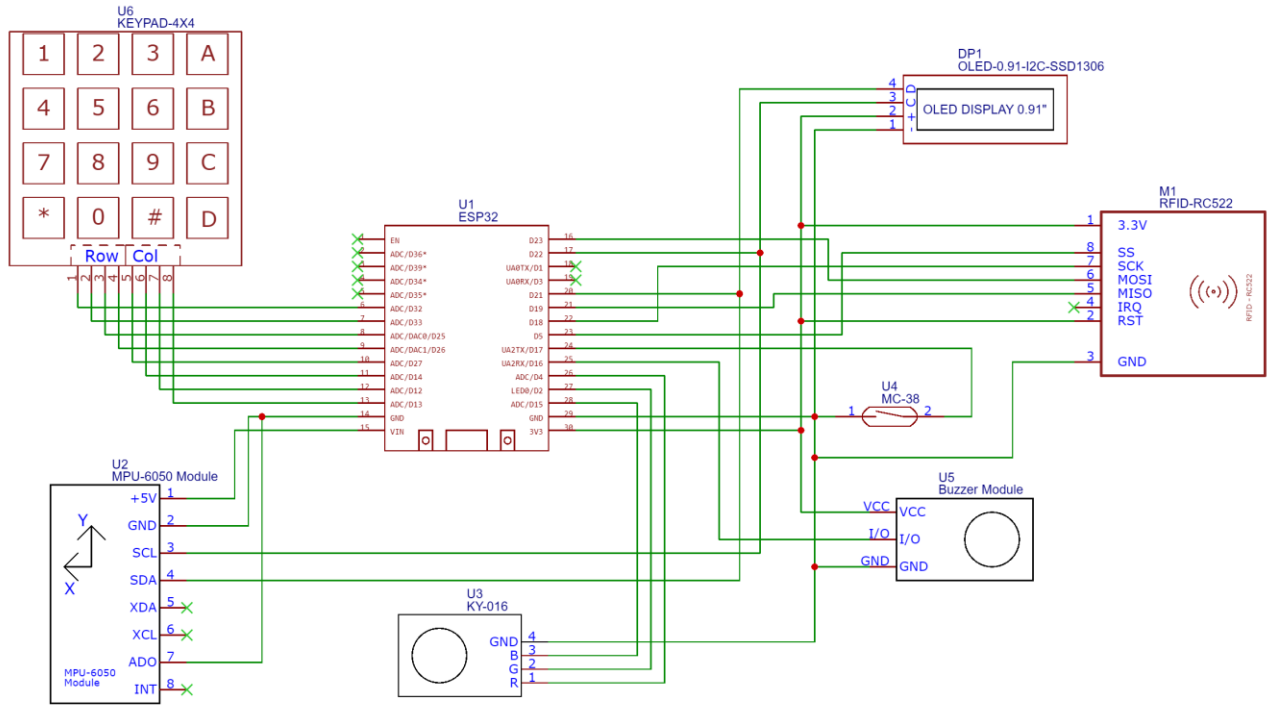


Рисунок 2.10 – Схема електрична принципова

Центральною частиною схеми є шинна архітектура, яка дозволяє паралельно обслуговувати декілька інтелектуальних модулів. Зокрема, використання шини I2C (лінії SDA та SCL) для OLED-дисплея та акселерометра MPU-6050 базується на принципі адресної ідентифікації пристроїв на фізичному рівні. Це дозволяє уникнути потреби у виділенні окремих керуючих ліній (Chip Select) для кожного пристрою, забезпечуючи економії GPIO. На схемі чітко

відображено спільні вузли з’єднання, які формують локальну мережу передачі телеметрії та даних візуалізації.

Для підсистеми ідентифікації через необхідність високої швидкості обміну даними, використано інтерфейс SPI. На відміну від I2C, він потребує чотирьох сигнальних ліній, проте забезпечує повнодуплексну передачу, що мінімізує час перебування системи у стані очікування при зчитуванні UID RFID-мітки.

Герконовий датчик підключений до GPIO17 за схемою з використанням внутрішнього підтягуючого резистора мікроконтролера. Це дозволяє системі однозначно ідентифікувати стан дверцят сейфа (відкрито чи закрито) без використання додаткової зовнішньої обв’язки, що спрощує конструкцію макета.

Модуль RGB-світлодіода KY-016 використовує три незалежні канали ШІМ (GPIO4, GPIO2, GPIO15). Це дає можливість реалізувати складні алгоритми колірної сигналізації, де кожен колір відповідає конкретному стану скінченного автомата: синій – очікування, помаранчевий – процес верифікації, червоний – тривога, зелений – успішний доступ.

Пасивний зумер на GPIO16 виконує роль акустичного сповіщувача. На схемі він винесений як окремий блок, що дозволяє генерувати сигнали різної частоти та тривалості для аудіального зворотного зв’язку з користувачем.

2.3 Функціональна декомпозиція та організація паралельного виконання завдань системи

Ефективне функціонування апаратно-програмного комплексу потребує одночасного виконання декількох процесів, що вимагає специфічної організації програмної логіки для уникнення блокувань обчислювальних ресурсів. Особливістю обраної архітектури на базі ESP32 є необхідність підтримки високої чутливості до фізичних впливів при одночасному обслуговуванні мережевих запитів та інтерфейсів користувача. Для реалізації такого функціонала застосовано метод функціональної декомпозиції, де кожна задача

розглядається як окремий неблокуючий процес у межах єдиного циклу виконання. На відміну від справжньої багатопоточності, така організація реалізує принцип кооперативної багатозадачності: кожна підсистема отримує процесорний час по черзі, але завдяки мінімальному часу виконання кожної ітерації загальна реактивність системи залишається достатньою для задач реального часу.

Першочерговим завданням системи є безперервний інерціальний моніторинг просторового положення макета. Опитування датчика MPU-6050 організовано через фіксовані часові інтервали відповідно до парадигми граничних обчислень, де первинний аналіз аномалій відбувається безпосередньо на пристрої без передачі сирих даних на сервер. Це дозволяє виявляти динамічні прискорення та нахили незалежно від поточного етапу автентифікації користувача. Така організація гарантує, що навіть під час введення PIN-коду або очікування відповіді від сервера підсистема безпеки залишається активною і здатною миттєво згенерувати сигнал тривоги при спробі переміщення пристрою [20].

Мікроелектромеханічний датчик MPU-6050 надає безперервний потік даних про проекції прискорення на три ортогональні осі X, Y та Z. Для визначення факту переміщення макета мікроконтролер обчислює модуль вектора результуючого прискорення за формулою:

$$A = \sqrt{A_x^2 + A_y^2 + A_z^2}, \tag{2.1}$$

де A – модуль вектора результуючого прискорення;

A_x, A_y, A_z – значення прискорення вздовж відповідних осей, виражені в одиницях g (прискорення вільного падіння).

У стані спокою на ідеально горизонтальній поверхні вектор результуючого прискорення системи дорівнює 1 g. При будь-якому динамічному впливі (удар, підняття, зсув) виникає додаткове лінійне прискорення. Система безпеки генерує

сигнал тривоги, якщо абсолютне відхилення вектора прискорення від еталонного значення перевищує програмно заданий поріг чутливості:

$$|A - 1| > A_{th}, \tag{2.2}$$

де A – модуль вектора результуючого прискорення;

A_{th} – програмно заданий поріг чутливості.

Для виявлення повільного, акуратного нахилу сейфа (який може не викликати різкого стрибка лінійного прискорення) паралельно обчислюються кути орієнтації у просторі – тангаж та крен. Математично значення кута тангажу отримується з проекцій статичного вектора гравітації:

$$\theta = arctg\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right), \tag{2.3}$$

де θ – кут тангажу вздовж горизонтальної осі;

A_x, A_y, A_z – значення прискорення вздовж відповідних осей, виражені в одиницях g (прискорення вільного падіння).

Аналогічним чином отримується значення кута крену:

$$\varphi = arctg\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right), \tag{2.4}$$

де φ – кут крену вздовж вертикальної осі;

A_x, A_y, A_z – значення прискорення вздовж відповідних осей, виражені в одиницях g (прискорення вільного падіння).

Мікроконтролер зберігає початкові значення кутів орієнтації під час ініціалізації охоронного режиму. Тривога активується за умови, якщо поточні значення відхиляються від початкових на кут, більший за допустиму похибку

(наприклад, понад п’ять градусів), що свідчить про спробу нахилити або перевернути об’єкт. Така математична модель забезпечує комплексний захист від будь-яких видів маніпуляцій з корпусом.

Паралельна обробка вхідних та вихідних потоків даних через протокол MQTT реалізована таким чином, що функція підтримки мережевої сесії виконується ітераційно, дозволяючи системі миттєво реагувати на зовнішні команди управління – такі як віддалене розблокування або деактивація тривоги. Така реалізація політики безпеки забезпечує блокування фізичних пристрої введення макета в режимі тривоги та передачу керування виключно віддаленому авторизованому клієнту. Вхідні повідомлення обробляються через механізм callback-функцій: при надходженні команди в підписаний топік система негайно змінює свій стан без очікування завершення поточної ітерації основного циклу.

Для забезпечення надійного та структурованого обміну даними між мікроконтролером та хмарним середовищем архітектуру MQTT-взаємодії поділено на логічні потоки (топіки). Це дозволяє чітко розмежувати керуючі команди, телеметричні дані та повідомлення системи безпеки. У проєкті реалізовано таке дерево топіків:

1) vovk/safe/auth – публікація запитів на розблокування після успішного зчитування локальних факторів (PIN-коду та RFID);

2) vovk/safe/control – підписка мікроконтролера на команди віддаленого клієнта (дозвіл на відкриття або відмова);

3) vovk/safe/status – періодична публікація поточного стану автомата станів пристрою;

4) vovk/safe/alarm – канал для сповіщення про тривогу (спрацювання акселерометра).

Надійність системи забезпечується використанням різних рівнів якості обслуговування (quality of service, QoS). Для рутинної телеметрії використовується QoS 0 (доставка максимум один раз), що знижує навантаження на мережу. Однак для сигналів тривоги та команд авторизації обов’язково

застосовується QoS 1 (доставка щонайменше один раз) з отриманням підтвердження від брокера (PUBACK). Форматування даних здійснюється у стандарті JSON. Такий підхід забезпечує повну сумісність з сучасними веб-технологіями та спрощує парсинг даних на стороні мобільного застосунку.

Для забезпечення високої реактивності системи програмна реалізація базується на неблокуючому циклі опитування периферії. На рисунку 2.11 зображено блок-схему алгоритму функціонування комплексу. Підсистема інерціального моніторингу має найвищий пріоритет в ітерації циклу, що гарантує миттєве виявлення фізичного впливу незалежно від стану мережевого з'єднання. Використання механізму callback-функцій для обробки MQTT-пакетів дозволяє асинхронно переривати логіку очікування для виконання команд віддаленого розблокування.

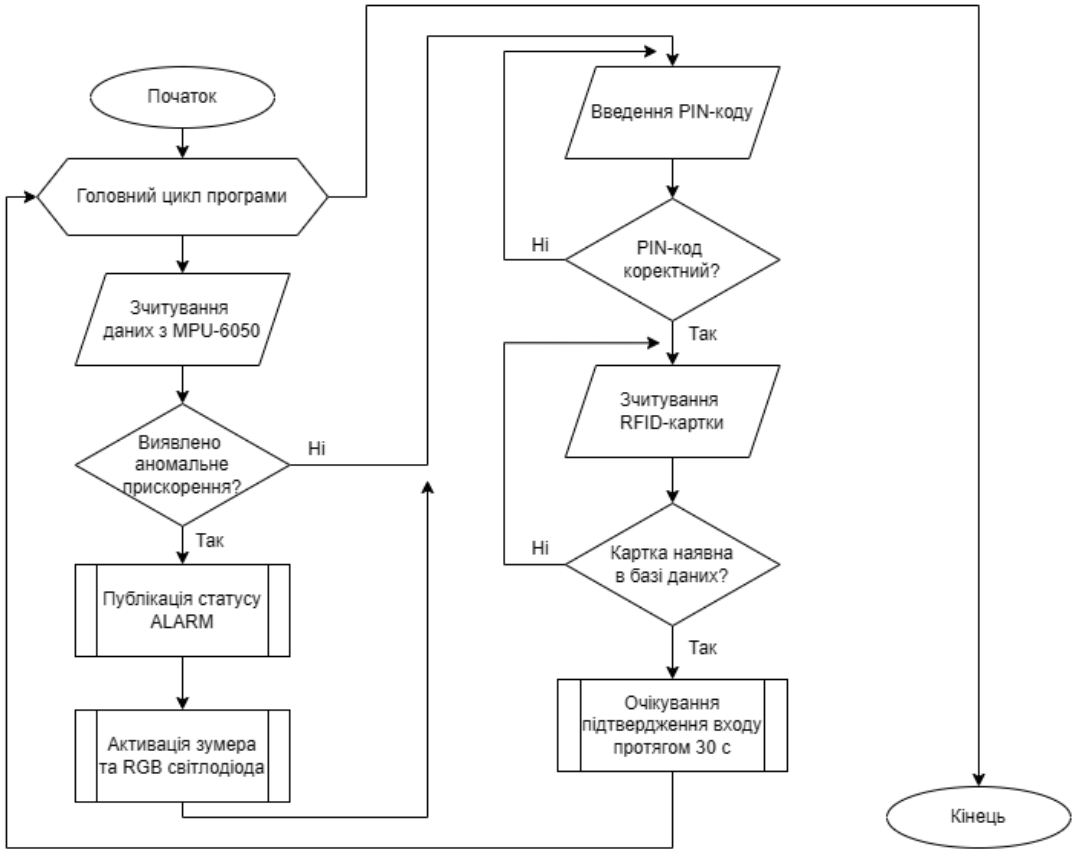


Рисунок 2.11 – Блок-схема алгоритму головного циклу функціонування системи

Система забезпечує квазіпаралельне управління засобами візуальної та акустичної індикації. Використання широтно-імпульсної модуляції для RGB-світлодіода та неблокуючих алгоритмів на основі апаратних таймерів для генерації звукових сигналів зумера дозволяє системі створювати складні динамічні ефекти без зупинки основного циклу. Розпаралелювання завдань на рівні неблокуючих викликів та апаратних переривань забезпечує високу живучість комплексу одночасно у трьох функціональних ролях: реєстратора подій – фіксація та публікація телеметрії стану датчиків у хмарний брокер; охоронного модуля – безперервний інерціальний моніторинг з миттєвою генерацією тривоги; термінала автентифікації – послідовна перевірка трьох факторів доступу з керуванням станами скінченного автомата станів.

2.4 Механізми самоорганізації та забезпечення стійкості архітектури

Стійкість архітектури апаратно-програмного комплексу забезпечується через впровадження механізмів автономного відновлення та строгого розмежування рівнів доступу [19]. У системі реалізовано принцип самоорганізації, що полягає у здатності пристрою самостійно ідентифікувати стани тривоги та адаптувати свою поведінку без зовнішнього втручання. Одним із основних аспектів такої самоорганізації є логіка обробки мережевих з’єднань: у разі втрати зв’язку з брокером повідомлень або Wi-Fi мережею мікроконтролер переходить у режим циклічного фонового відновлення сесії, не зупиняючи при цьому виконання основних функцій безпеки та локального моніторингу.

Механізмом забезпечення стійкості системи є використання часових тайм-аутів для запобігання ситуаціям «зависання» системи у проміжних станах. Оскільки процес автентифікації розділений на декілька етапів, існує ризик залишення пристрою в режимі очікування після успішного проходження перших рівнів перевірки. Для нівелювання цієї загрози в архітектуру закладено алгоритм автоматичного скидання сесії до початкового стану безпеки, якщо фінальне

підтвердження від віддаленого застосунку не надійшло протягом встановленого часового вікна. Це гарантує цілісність захисту навіть у разі неуважності користувача або технічних збоїв на стороні клієнта.

Стійкість до фізичного та логічного зламу реалізується через концепцію віддаленого детермінованого управління в режимі тривоги, що відповідає принципам моделі нульової довіри (zero-trust) [19, 28]. При реєстрації несанкціонованого переміщення або вібрації система переходить у стан активної небезпеки, у якому будь-яка локальна взаємодія через клавіатуру ігнорується на рівні алгоритму. Центр управління в цей момент повністю переміщується до віддаленого застосунку, і лише авторизована команда через захищений MQTT-топік здатна повернути комплекс до штатного режиму роботи. Така самостійна зміна пріоритетів управління дозволяє захистити систему від спроб деактивації тривоги шляхом фізичного пошкодження або маніпуляцій з локальними інтерфейсами вводу.

Надійність збереження конфігураційних даних забезпечується програмним механізмом взаємодії з енергонезалежною пам’яттю. Використання структурованих сховищ для PIN-кодів та ідентифікаторів дозволяє системі миттєво самовідновлюватися після перезавантажень або раптових перебоїв у живленні. Таким чином, поєднання алгоритмів самодіагностики, динамічного перерозподілу прав доступу та енергонезалежності формує стійку архітектурну модель, здатну функціонувати в умовах високих ризиків безпеки.

2.5 Висновки до другого розділу

У другому розділі було спроєктовано та обґрунтовано архітектуру апаратно-програмного комплексу, що базується на принципах розподілених кіберфізичних систем. Розроблена модель взаємодії між локальним інтелектуальним вузлом на базі ESP32, хмарним MQTT-брокером та мобільним застосунком забезпечує високий рівень гнучкості та дозволяє реалізувати

концепцію віддаленого контролю безпеки в реальному часі. Вибір компонентної бази, зокрема мікроконтролера з двоядерною архітектурою, дозволив ефективно розділити обчислювальні ресурси для паралельного виконання завдань мережевої комунікації та безперервного моніторингу периферії.

Проведена функціональна декомпозиція завдань підтвердила можливість реалізації неблокуючої логіки управління, що дозволяє підтримувати чутливості інерціальних датчиків під час активної взаємодії користувача з інтерфейсами автентифікації. Впровадження механізмів самоорганізації, таких як автоматичне відновлення мережевих з’єднань, тайм-аути сесій та делегування повноважень управління віддаленому клієнту в режимі тривоги, суттєво підвищує живучість системи та її стійкість до спроб фізичного саботажу.

Обрані методи збереження даних у енергонезалежній пам’яті гарантують стабільну роботу комплексу навіть за умов нестабільного живлення. Таким чином, сформована архітектурна та апаратна база є цілісним підґрунтям для подальшої програмної реалізації алгоритмів трифакторної автентифікації та розробки користувацьких інтерфейсів, що будуть розглянуті у наступному розділі.

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ «РОЗУМНИЙ СЕЙФ»

3.1 Обґрунтування вибору програмних засобів та інструментарію розробки

Програмна реалізація кіберфізичної системи «Розумний сейф» вимагає інтеграції різнорідних технологічних стеків, що забезпечують стабільну роботу на рівні мікроконтролера, хмарного середовища та мобільного клієнта. Вибір інструментальних засобів базується на критеріях продуктивності, безпеки та можливості підтримки асинхронної взаємодії в реальному часі.

Для розробки внутрішнього програмного забезпечення граничного вузла обрано екосистему PlatformIO, що функціонує на базі середовища Visual Studio. PlatformIO надає розширені можливості управління бібліотеками та конфігураціями, що значно перевершує стандартні можливості Arduino IDE. Використання мови C++ дозволяє оптимізувати обчислювальні ресурси мікроконтролера ESP32, забезпечуючи низькорівневий доступ до периферійних інтерфейсів SPI та I2C для взаємодії з RFID-зчитувачем та акселерометром.

Мережевий рівень системи базується на протоколі MQTT та бібліотеці PubSubClient. Вибір даного протоколу обумовлений його мінімальними накладними витратами та орієнтованістю на подієву модель (event-driven architecture), що ідеально підходить для передачі тривожних сповіщень. Для структурування повідомлень використано формат JSON, що забезпечує гнучкість при передачі складних об’єктів, таких як UID карток або статуси інерціальних аномалій.

Клієнтська частина реалізована за допомогою фреймворку React Native. Це дозволяє створити кросплатформовий мобільний застосунок із реактивним інтерфейсом, здатним миттєво відображати зміни стану сейфа, що надходять від MQTT-брокера. Головною перевагою React Native є єдина кодова база для платформ iOS та Android, що суттєво скорочує час розробки та спрощує підтримку застосунку. Бібліотека react-native-mqtt забезпечує нативну

інтеграцію з MQTT-брокером безпосередньо на стороні клієнта, а розвинена екосистема пакетів дозволяє використовувати готові рішення для роботи з push-сповіщеннями та локальним захищеним сховищем токенів автентифікації. Компонентна архітектура React забезпечує чіткий поділ логіки відображення та бізнес-логіки, що спрощує масштабування застосунку при додаванні нових функцій керування сейфом. Хмарна інфраструктура проєкту спирається на використання брокера HiveMQ.

3.2 Алгоритмічне забезпечення та програмна реалізація інтелектуального вузла

Програмна реалізація інтелектуального вузла на базі мікроконтролера ESP32 є основним етапом створення апаратно-програмного комплексу, оскільки саме на цьому рівні забезпечується безпосередня взаємодія між фізичним та інформаційним середовищами. Основним викликом при розробці прошивки стала необхідність підтримки високої реактивності системи в умовах багатозадачності, де процеси мережевого обміну не повинні блокувати необхідні функції безпеки.

Програмний код структуровано відповідно до архітектурного принципу поділу відповідальності: функція setup виконується одноразово при старті та відповідає за перевірку і конфігурацію апаратного оточення, тоді як функція loop реалізує нескінченний цикл обробки подій. Така структура є стандартом для Arduino-сумісних платформ і забезпечує чіткий розподіл між фазою ініціалізації та фазою виконання.

У функції setup реалізовано послідовну ініціалізацію периферійних інтерфейсів: спочатку активується шина I2C на контактах GPIO 21 та GPIO 22 для підключення дисплея SSD1306 та акселерометра MPU-6050, після чого ініціалізується шина SPI для взаємодії з RFID-модулем RC522. Така послідовність є принциповою – ініціалізація дисплея раніше за мережеві

підсистеми дозволяє відображати діагностичні повідомлення вже на етапі підключення до Wi-Fi, що спрощує налагодження системи в польових умовах. Блок-схему процедури ініціалізації наведено на рисунку 3.1.

Рисунок 3.1 – Блок-схема процедури ініціалізації

Підсистема енергонезалежного зберігання реалізована за допомогою бібліотеки Preferences. При кожному старті система зчитує PIN-код із пам’яті NVS (Non-Volatile Storage) мікроконтролера із резервним значенням за замовчуванням. Сейф зберігає індивідуальний код доступу після відключення

живлення, а команда new_pin, отримана через MQTT, одразу записується в NVS і набуває чинності при наступному перезавантаженні.

Основою керування логікою безпеки є скінченний автомат (Finite State Machine, FSM), реалізований через перелічуваний тип SafeState та глобальну змінну currentState. Вибір FSM як архітектурного патерну обумовлений природою задачі: система безпеки повинна мати суворо визначену поведінку в кожен момент часу, виключаючи можливість паралельного виконання несумісних операцій – наприклад, одночасного очікування RFID-картки та підтвердження від застосунку.

Автомат має чотири стани, між якими відбуваються переходи, спричинені зовнішніми подіями – натисканням клавіш, піднесенням картки або отриманням MQTT-повідомлення:

1. LOCKED – початковий стан та стан після будь-якої невдалої спроби. Система очікує введення PIN-коду через матричну клавіатуру. Клавіша «#» завершує введення і запускає перевірку; клавіша «*» скидає введений рядок без зміни стану.

2. WAITING_RFID – стан після успішної валідації PIN. Модуль RC522 активно опитується на наявність нової картки. При виявленні авторизованого носія UID серіалізується у формат JSON та публікується в топік vovk/safe/auth; система переходить до наступного стану та фіксує часову мітку початку очікування.

3. WAITING_APP – стан очікування підтвердження від мобільного застосунку. Якщо протягом 30 секунд не надходить команда app_confirm або app_deny, спрацьовує таймаут і система повертається до стану LOCKED. Без механізму таймауту втрата з’єднання з MQTT-брокером заблокувала б систему назавжди.

4. UNLOCKED – стан відкритого сейфу, наданого доступу. Система активно моніторить геркон, підключений до GPIO 17: щойно дверцята фізично закриваються (надходження сигналу LOW із затримкою 1 с для усунення

брязкоту контактів), автомат повертається до стану LOCKED та публікує відповідний статус.

Граф станів системи наведено на рисунку 3.2.

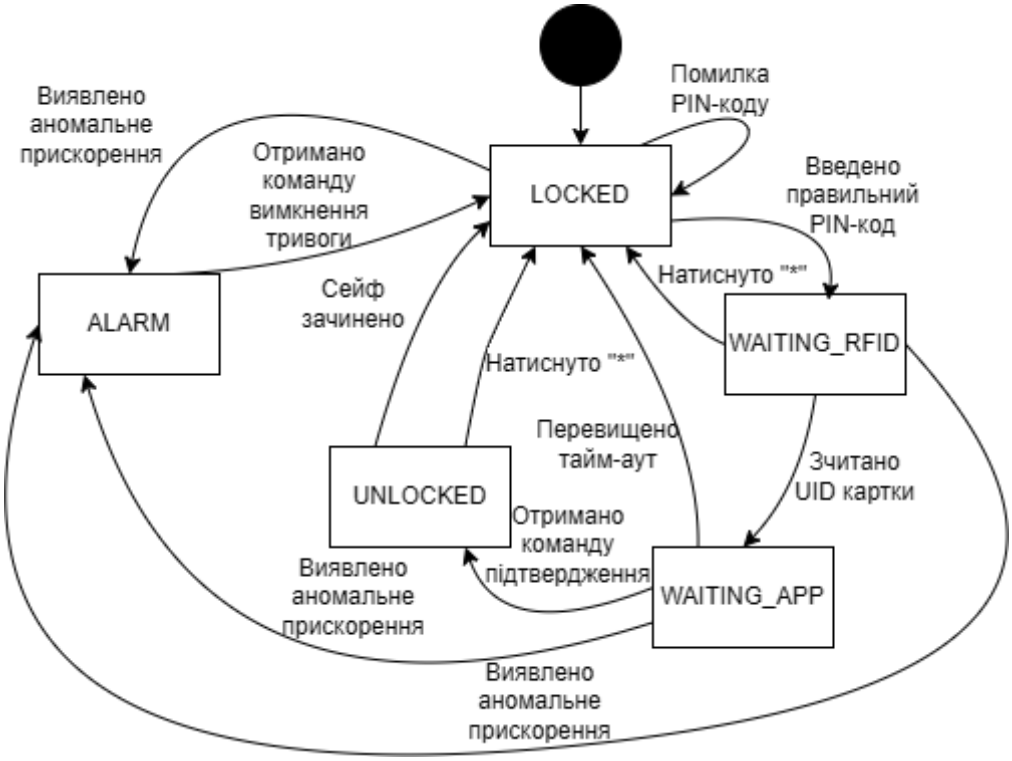


Рисунок 3.2 – Граф станів системи

Комунікація між граничним вузлом та хмарною інфраструктурою здійснюється через чотири MQTT-топіки, що утворюють асиметричний канал керування: три топіки використовуються для публікації подій від ESP32, і лише один – для отримання команд від мобільного застосунку. Перелік топіків, напрями обміну та формат корисного навантаження наведено у таблиці 3.1.

Таблиця 3.1 – MQTT-топіки системи «Розумний сейф»

Топік	Напрям	Призначення та корисне навантаження
vovk/safe/auth	Від ESP32 до застосунку	Публікує UID зчитаної RFID-картки для верифікації у мобільному застосунку. Наприклад, {«uid»: «A1:B2:C3:D4»}.
vovk/safe/alarm	Від ESP32 до застосунку	Надсилає сповіщення по виявлення аномального прискорення (спроба переміщення сейфа). Наприклад, «TAMPER DETECTED!».
vovk/safe/status	Від ESP32 до застосунку	Публікує поточний стан сейфа при кожній зміні стану FSM. Наприклад, «UNLOCKED» чи «LOCKED».
vovk/safe/control	Від застосунку до ESP32	Канал керування сейфом: підтвердження або відхилення доступу, гостьовий режим, скидання тривоги, зміна PIN-коду. Наприклад, {«command: «app_confirm»}, {«command: «alarm_off»} тощо.

Функція зворотного виклику callback обробляє всі вхідні повідомлення з топіка vovk/safe/control. Десеріалізація виконується бібліотекою ArduinoJson у статично виділений буфер розміром 300 байт, що виключає фрагментацію динамічної пам’яті, адже дана мікроконтролерна система має обмежений обсяг RAM. Команда new_pin додатково перевіряє мінімальну довжину нового коду (не менше 4 символів) перед записом у NVS, забезпечуючи базову валідацію на стороні пристрою.

Підсистема виявлення несанкціонованого переміщення реалізована на основі кооперативної багатозадачності з використанням функції millis. Кожні 150 мс основний цикл зчитує показники акселерометра MPU-6050 по осях X та

Y і порівнює їх з попередньо збереженими значеннями. Якщо абсолютне відхилення по будь-якій з осей перевищує програмно заданий поріг чутливості, активується прапорець alarmActive та публікується повідомлення про тривогу.

Інерційний моніторинг активується лише після 10 секунд від старту. Це запобігає хибним спрацюванням під час самого процесу встановлення сейфа або його переміщення авторизованою особою одразу після увімкнення. Моніторинг деактивовано у стані UNLOCKED – коли доступ вже надано, переміщення сейфа є очікуваною поведінкою.

Функція triggerAlarm реалізує звуко-світлову індикацію через немодифікований таймер з інтервалом 250 мс, що забезпечує переривчастий сигнал зумера без використання delay і не блокує обробку MQTT-повідомлень. Це дозволяє команді alarm_off бути прийнятою і виконаною навіть під час активної тривоги. Алгоритм інерціального моніторингу наведено на рисунку 3.3.

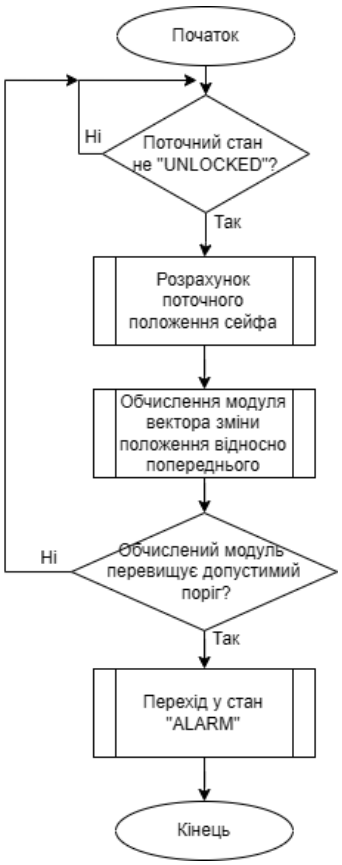


Рисунок 3.3 – Блок-схема алгоритму виявлення аномального переміщення

3.3 Проєктування структури даних та стратегії збереження інформації

Ефективність функціонування розподіленої кіберфізичної системи значною мірою визначається раціональністю побудови моделі даних та надійністю механізмів їхнього зберігання. У проєкті реалізовано гібридну стратегію збереження інформації, що поєднує локальне сховище на граничному вузлі для забезпечення автономності в умовах нестабільного мережевого з'єднання та централізовану реляційну базу даних для управління профілями користувачів, ідентифікаторами доступу і ведення повного журналу аудиту. Така дворівнева архітектура дозволяє досягти балансу між реактивністю системи та цілісністю даних.

Для забезпечення цілісності та структурованості інформації на боці хмарної інфраструктури розроблено логічну модель бази даних, що складається з трьох основних сутностей, зв'язки між якими наведено на рисунку 3.4. Вибір реляційної моделі обумовлений необхідністю суворого дотримання посилальної цілісності між ідентифікаторами фізичного доступу та персональними обліковими записами користувачів. Порушення цих зв'язків могло б призвести до вразливості – надання доступу за карткою, власник якої був деактивований в системі, або фіксації події в журналі без прив'язки до відповідального користувача.

Сутність Профілі є центральним вузлом моделі та містить такі атрибути: ідентифікатор профілю, ім'я користувача, електронну пошту, роль у системі, PIN-код доступу та поле «PIN-код діє до», що визначає термін дії режиму спрощеної автентифікації. Сутність RFID картки описує фізичні носії доступу і містить ідентифікатор картки, UID картки, ідентифікатор власника, назву картки, ознаку активності та дату внесення до бази. Сутність Логи призначена для аудиту і містить ідентифікатор запису, дату створення, опис дії та ідентифікатор автора події.

Між сутностями визначено два зв’язки. Сутності Профілі та RFID картки пов’язані зв’язком «один-до-багатьох» – один профіль може мати довільну кількість прив’язаних карток, тоді як кожна картка належить рівно одному власнику. Сутності Профілі та Логи також пов’язані зв’язком «один-до-багатьох» – один профіль може бути автором багатьох записів журналу, однак поле ідентифікатора автора є необов’язковим, що дозволяє фіксувати системні події без прив’язки до конкретного користувача.

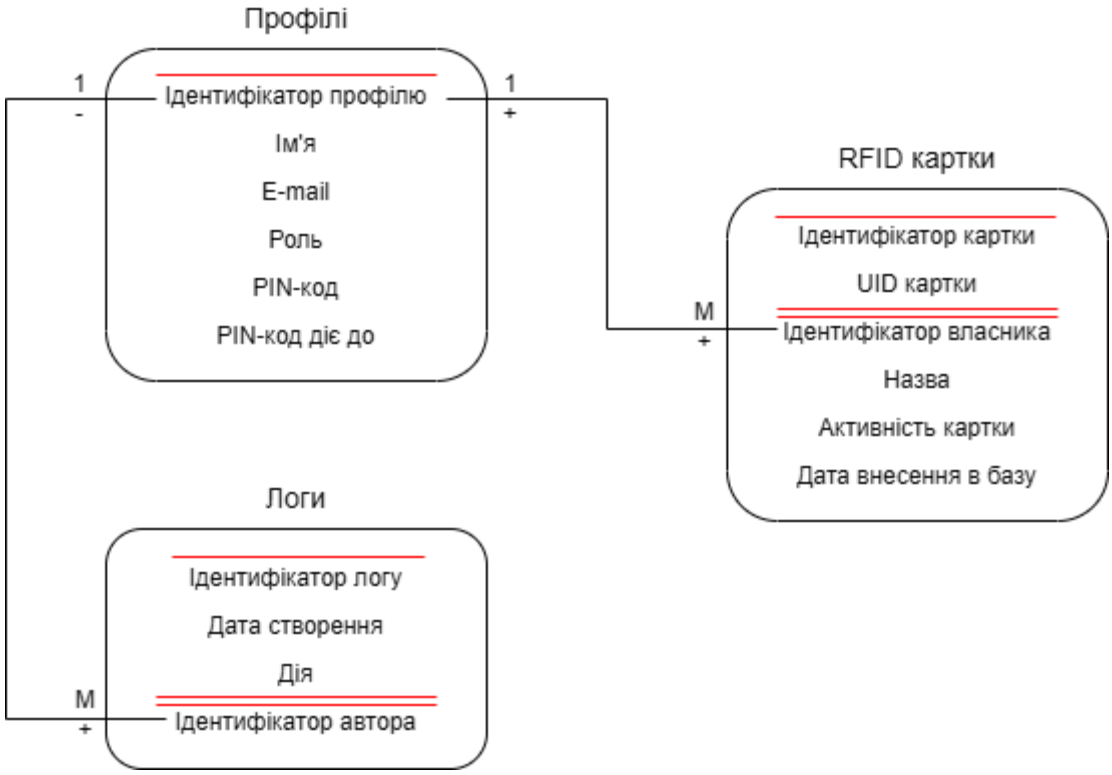


Рисунок 3.4 – Логічна модель даних (ER-діаграма)

Центральним вузлом ієрархії є таблиця profiles. Вона зберігає унікальні ідентифікатори користувачів у форматі UUID, що гарантує глобальну унікальність запису незалежно від середовища розгортання. Поля name та email забезпечують ідентифікацію користувача на рівні мобільного застосунку та слугують для надсилання push-сповіщень про події безпеки. Поле role визначає рівень привілеїв облікового запису в системі – зокрема, розмежовує права

власника сейфа та тимчасових користувачів. Поле pin_code зберігає актуальне значення персонального коду доступу у відкритому вигляді, що обумовлено необхідністю прямого порівняння на стороні мікроконтролера ESP32, який не має обчислювальних ресурсів для виконання криптографічних хеш-функцій у реальному часі в межах основного циклу обробки подій.

Функціональне призначення поля pin_bypass_until типу timestamptz полягає в апаратній реалізації механізму тимчасового спрощення автентифікації: якщо поточний момент часу передує значенню цього поля, система автоматично переходить з трифакторного режиму до однофакторного – для отримання доступу достатньо лише ввести коректний PIN-код, тоді як піднесення RFID-картки та підтвердження від мобільного застосунку не вимагаються. Це дозволяє реалізувати сценарій надання власником сейфа тимчасового доступу довіреній особі, яка знає PIN, але не має фізичної картки і не встановлювала мобільний застосунок. Після закінчення встановленого терміну – без будь-якого додаткового втручання – система автоматично повертається до стандартного трирівневого захисту, оскільки умова поточний час не повинен перевищувати pin_bypass_until перестає виконуватись. Якщо поле містить NULL або минулу дату, режим bypass вважається неактивним.

Таблиця rfid_cards реалізує управління фактором володіння в системі автентифікації. Кожен запис представляє окремий фізичний носій і пов’язаний із конкретним профілем через зовнішній ключ owner_id, що утворює зв’язок типу «один-до-багатьох» – один користувач може мати кілька карток одночасно, наприклад основну та резервну. Поле uid зберігає унікальний ідентифікатор RFID-мітки у текстовому форматі з роздільниками у вигляді двокрапок, що відповідає формату, який генерує функція getCardUID на стороні мікроконтролера. Поле label дозволяє іменувати картки для зручності адміністрування в інтерфейсі застосунку. Поле created_at фіксує момент реєстрації картки в системі та слугує для хронологічного аудиту.

Основним з точки зору оперативного реагування на інциденти є булеве поле is_active. У разі втрати або крадіжки картки адміністратор одним керуючим запитом через мобільний застосунок встановлює значення is_active = false для відповідного запису. При наступній спробі авторизації система зчитає UID піднесеної картки, знайде відповідний запис у таблиці та відхилить запит, незважаючи на збіг UID – оскільки картка деактивована. При цьому профіль користувача та вся історія подій, пов'язаних із цією карткою в таблиці logs, повністю зберігаються, що забезпечує можливість проведення ретроспективного розслідування інциденту.

Таблиця logs забезпечує повну прозорість та аудит безпеки системи. Кожна подія в життєвому циклі сейфа – успішна або невдала спроба входу, спрацювання акселерометра, надання або відхилення доступу, активація гостьового режиму, зміна PIN-коду, скидання тривоги – фіксується із міткою часу у форматі timestamptz та прив'язкою до ініціатора дії через поле author_id. Використання цілочисельного первинного ключа типу int8 замість UUID обумовлене характером таблиці: записи журналу лише додаються і ніколи не редагуються та не видаляються, що виключає потребу в глобальній унікальності ідентифікатора поза межами однієї бази даних і водночас забезпечує максимальну швидкість вставки нових записів. Поле action зберігає текстовий опис події, що дозволяє фільтрувати журнал за типами подій та будувати аналітичні звіти для оцінки стану безпеки об'єкта за довільний проміжок часу.

Логічна модель (рисунок 3.4), відображає два явні зв'язки між сутностями, які реалізовано у фізичній схемі бази даних, зображеній на рисунку 3.5.

Перший зв'язок – між таблицями profiles та rfid_cards через поле owner_id – обов'язковий зі сторони картки: обмеження NOT NULL на зовнішньому ключі гарантує, що жодна картка не може існувати без прив'язки до конкретного власника. У фізичній реалізації це забезпечується каскадною поведінкою: видалення профілю автоматично деактивує або видаляє всі пов'язані картки, що унеможливлює появу осиротілих записів із недійсним owner_id. Водночас

профіль може існувати без жодної прив'язаної картки – наприклад, адміністраторський обліковий запис, що керує системою виключно через мобільний застосунок.

Другий зв'язок – між таблицями profiles та logs через поле author_id – є навмисно необов'язковим, що відображено відсутністю обмеження NOT NULL на цьому полі у фізичній схемі. Це технічне рішення дозволяє фіксувати системні події, ініційовані автоматично – спрацювання акселерометра без попередньої ідентифікації користувача, таймаут очікування підтвердження, автоматичне блокування після закриття дверцят – із порожнім значенням поля author_id. Така модель забезпечує консистентність журналу навіть у позаштатних ситуаціях, коли ініціатора події неможливо однозначно визначити.

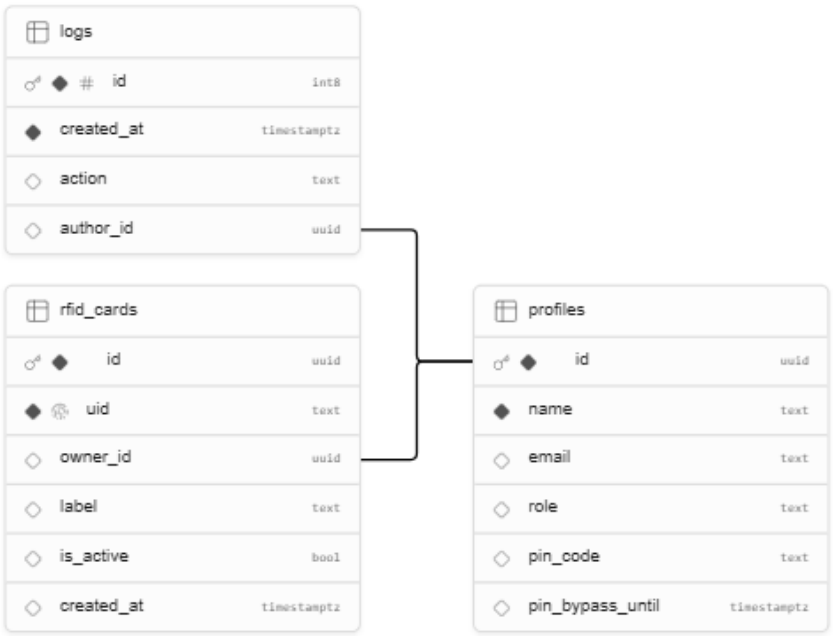


Рисунок 3.5 – Фізична схема бази даних у Supabase

Поряд із централізованою базою даних, граничний вузол використовує підсистему Non-Volatile Storage мікроконтролера ESP32, реалізовану через бібліотеку Preferences. Ця підсистема забезпечує збереження даних у виділеній області флеш-пам'яті мікроконтролера, що зберігається при відключенні

живлення та апаратному перезавантаженні. У постійній пам’яті зберігається мінімально необхідний набір даних: PIN-код у вигляді рядка під ключем «pin» у просторі імен «safe-config». Відповідно до принципу мінімальної достатності, жодні інші конфіденційні дані локально не кешуються – UID авторизованих карток не зберігаються на пристрої, що унеможливлює їх зчитування у разі несанкціонованого фізичного доступу до мікроконтролера.

Команда new_pin, отримана через топік vovk/safe/control, проходить первинну валідацію на стороні пристрою – перевіряється мінімальна довжина нового коду у чотири символи. Лише після успішної перевірки новий PIN записується одночасно в оперативну змінну masterCode та у флеш-пам’ять через виклик preferences.putString(). Це гарантує, що зміна набирає чинності негайно для поточного сеансу і зберігається після наступного перезавантаження без необхідності повторного отримання команди.

Така архітектура локального сховища визначає чітку поведінку системи в умовах відсутності мережевого з’єднання. Перший фактор автентифікації – перевірка PIN-коду – виконується повністю локально і не залежить від доступності MQTT-брокера чи будь-якої зовнішньої інфраструктури. Однак другий і третій фактори вимагають мережевої взаємодії: верифікація UID картки в таблиці rfid_cards та отримання підтвердження від мобільного застосунку здійснюються через хмарну інфраструктуру. Якщо з’єднання відсутнє на момент піднесення картки, функція reconnect блокує виконання до відновлення з’єднання, або спрацьовує таймаут у 30 секунд, після чого FSM повертається до стану LOCKED. Це свідоме проєктне рішення, адже у системі фізичного захисту матеріальних цінностей відмова у доступі є принципово безпечнішою поведінкою, ніж надання доступу в умовах деградації мережевої інфраструктури.

3.4 Розробка мобільного застосунку для віддаленого керування та моніторингу

Завершальним етапом створення цілісної екосистеми безпеки є розробка клієнтського мобільного застосунку, який виступає основним інтерфейсом взаємодії користувача з апаратною частиною сейфа. Застосунок не лише виконує роль пульта дистанційного керування, а й є інтелектуальним центром прийняття рішень третього фактора автентифікації – саме тут відбувається фінальна верифікація особи, яка намагається отримати доступ, та ведеться повний аудит усіх подій системи в реальному часі. На відміну від традиційних систем фізичної безпеки, де рішення про надання доступу приймається автоматично на основі збігу ідентифікатора, у даній системі людина залишається фінальною ланкою у ланцюжку авторизації, що принципово підвищує стійкість до атак із використанням скопійованих або вкрадених носіїв доступу.

Для реалізації мобільного клієнта обрано фреймворк React Native, що забезпечує єдину кодову базу для платформ iOS та Android із нативною рендеризацією елементів інтерфейсу. Це рішення дозволяє уникнути розробки та підтримки двох паралельних версій застосунку, зберігаючи при цьому якість користувацького досвіду, характерну для нативних додатків. Компонентна архітектура React забезпечує чіткий поділ між логікою відображення та бізнес-логікою: кожен екран або функціональний блок інкапсульований у самостійний компонент із власними відповідальностями.

Навігаційна структура застосунку є умовною та визначається станом авторизаційної сесії. При запуску система асинхронно перевіряє наявність активної сесії у захищеному сховищі пристрою та підписується на зміни стану автентифікації. Залежно від результату перевірки користувачеві відображається або стек екранів авторизації, або основний стек застосунку. Такий підхід унеможливлює доступ до функцій керування без підтвердженої ідентифікації – навіть якщо користувач спробує перейти на захищений екран напряму, система

автоматично перенаправить його на екран входу (рисунок 3.6). При виході з облікового запису сесія знищується, MQTT-з'єднання розривається, а весь локальний стан скидається до початкових значень.

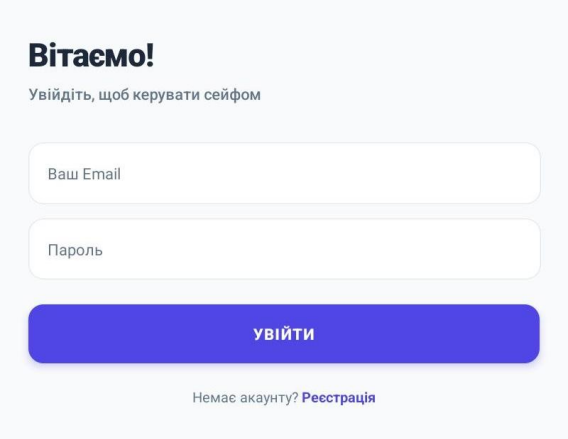


Рисунок 3.6 – Екран входу до застосунку

Управління станом застосунку реалізовано через єдине централізоване сховище, побудоване на основі бібліотеки Zustand. Сховище координує всі аспекти роботи застосунку: сесію та профіль користувача, стан підключення до MQTT-брокера, поточний стан сейфа, активність тривоги, прапорець очікування підтвердження та ідентифікатор останньої зчитаної картки. Вибір саме цієї бібліотеки обумовлений можливістю виконання асинхронних операцій безпосередньо всередині сховища – запити до бази даних та публікація MQTT-повідомлень відбуваються в одному місці, що спрощує відстеження потоку даних та усуває необхідність передачі зворотних викликів між компонентами.

MQTT-з'єднання ініціалізується одразу після успішної авторизації користувача і підтримується протягом усього часу роботи застосунку. Для підключення використовується протокол WebSocket Secure з підтримкою шифрування TLS, що дозволяє встановлювати захищене з'єднання безпосередньо з мобільного застосунку без необхідності проміжного серверного шару. Відсутність власного бекенду означає, що команди керування передаються

напряму між застосунком та граничним вузлом через хмарний брокер, що скорочує затримку та усуває єдину точку відмови у вигляді власного сервера.

Після встановлення з'єднання застосунок підписується на весь простір топіків сейфа за допомогою символу підстановки. Це забезпечує отримання всіх повідомлень від граничного вузла в межах одного підписання і спрощує додавання нових топіків у майбутньому без необхідності зміни коду клієнта. Індикатор стану підключення відображається на головному екрані та оновлюється в реальному часі – при втраті з'єднання користувач негайно отримує візуальне попередження і не може надіслати команди керування до відновлення зв'язку, що запобігає відправці команд у невизначений стан.

Обробка вхідних повідомлень відбувається асинхронно у фоновому режимі залежно від топіку. Повідомлення про зміну стану сейфа оновлюють інтерфейс відповідно – замок відображається відкритим або закритим, прапорець очікування скидається. Повідомлення про тривогу активують відповідний індикатор у вигляді пріоритетного банера на головному екрані та автоматично записують подію в журнал без участі користувача. Повідомлення з ідентифікатором картки запускають найскладнішу логіку верифікації, описану нижче. Публікація команд керування здійснюється із рівнем якості доставки QoS 1, що гарантує отримання повідомлення брокером щонайменше один раз. Ця властивість використовується для команд відкриття сейфа та скидання тривоги, втрата яких є неприпустимою.

При отриманні повідомлення з ідентифікатором зчитаної картки застосунок виконує запит до хмарної бази даних, намагаючись знайти активний запис із відповідним UID. Цей запит є асинхронним і виконується у фоновому режимі, не блокуючи інтерфейс користувача. Залежно від результату запиту та поточної ролі користувача можливі три принципово різні сценарії.

У першому сценарії картка знайдена в базі даних та має активний статус. Застосунок переводить інтерфейс у режим очікування підтвердження: на головному екрані з'являється ім'я власника картки та кнопка підтвердження, яка

стає центральним елементом взаємодії. Власник сейфа бачить хто саме намагається отримати доступ і приймає усвідомлене рішення – підтвердити або відхилити запит. Це принципово відрізняє систему від автоматичних рішень: людина завжди знає хто і коли звертався до сейфа.

У другому сценарії картка не знайдена в базі даних або деактивована, а поточний авторизований користувач є власником сейфа. У цьому випадку застосунок не відправляє жодних автоматичних команд на пристрій, натомість зберігаючи невідомий ідентифікатор та відображаючи власнику пропозицію зареєструвати нову картку. Це забезпечує зручний сценарій введення нових носіїв доступу без необхідності ручного введення шістнадцяткових символів.

У третьому сценарії картка невідома, а поточний авторизований користувач не є власником. Застосунок автоматично публікує команду відхилення без участі людини та записує подію в журнал із відповідною позначкою. Такий підхід запобігає ситуації, коли зловмисник міг би скористатися відсутністю власника для авторизації невідомої картки через застосунок неуважного користувача.

Система рольового доступу реалізована на рівні головного екрана через динамічне обчислення прав на основі ролі поточного користувача та стану часу делегування доступу. Кожна роль отримує власний кольоровий бейдж на головному екрані, що забезпечує миттєву візуальну ідентифікацію рівня доступу без необхідності заходити в налаштування.

Власник має повний доступ до всіх функцій системи без обмежень (рисуглк 3.7): керування сейфом, перегляд повного журналу подій усіх користувачів, адміністрування облікових записів, зміна PIN-коду пристрою та реєстрація нових RFID-карток. Саме власник є єдиним, хто може скидати тривогу та надавати або відкликати права доступу іншим користувачам.

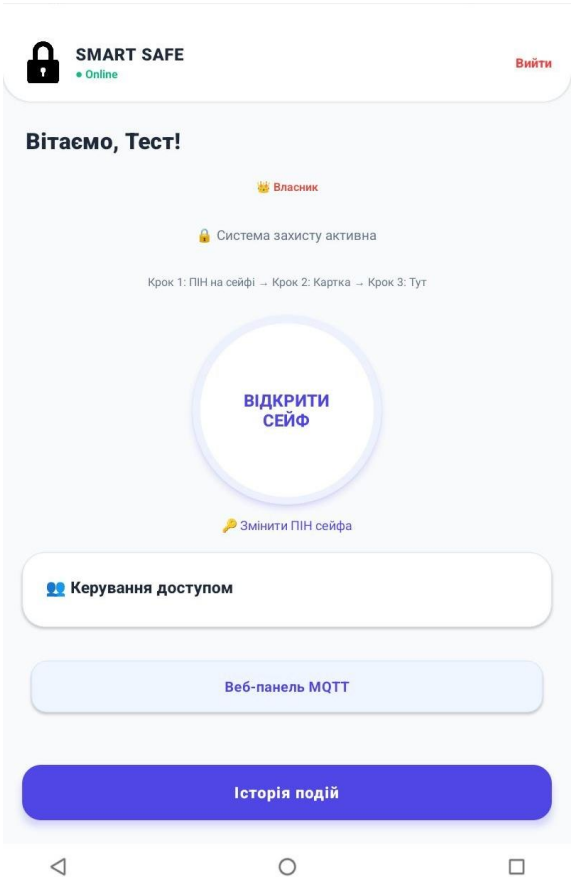


Рисунок 3.7 – Головний екран застосунку, вигляд для ролі «Власник»

Користувач може керувати сейфом у повному обсязі – вводити PIN на пристрої, підносити картку та підтверджувати доступ через застосунок – а також переглядати власну історію подій. Однак він позбавлений адміністративних функцій: не може змінювати ролі інших, реєструвати картки або змінювати PIN пристрою. Журнал подій для цієї ролі автоматично фільтрується – відображаються лише власні дії, що забезпечує конфіденційність між різними користувачами одного сейфа.

Наглядач є принципово пасивною роллю, призначеною для моніторингу без права втручання. Користувач із цією роллю бачить повний журнал усіх подій у реальному часі, включаючи тривоги та спроби несанкціонованого доступу, але позбавлений будь-якої можливості впливати на стан сейфа. Ця роль є корисною для сценарію охоронця або відповідального за безпеку, який повинен відстежувати активність, але не мати права самостійно відкривати сейф.

Гість отримує мінімальний рівень доступу. За відсутності активного дозволу доступу без автентифікації він бачить лише інформаційне повідомлення з пропозицією звернутися до власника. За наявності дозволу гість може відкрити сейф, підтвердивши знання PIN-коду.

Функція тимчасового делегування реалізує сценарій надання короткострокового доступу без передачі фізичної картки або постійної зміни налаштувань системи. Власник через екран адміністрування одним натисканням встановлює для обраного користувача часову мітку закінчення дозволу доступу – на 15 хвилин від поточного моменту. Протягом цього часу вказаний користувач може відкрити сейф, ввівши лише PIN-код, без піднесення картки та без очікування підтвердження в застосунку.

Перевірка активності дозволу відбувається щоразу при відкритті головного екрана через оновлення профілю з бази даних. Така архітектура дозволяє власнику відкликати дозвіл достроково – досить встановити в базі минулу дату або порожнє значення – і зміна набуде чинності при наступному відкритті застосунку користувачем без необхідності будь-якої взаємодії з його боку. Після закінчення терміну дозволу система автоматично повертається до стандартного трифакторного режиму, оскільки умова порівняння поточного часу з часовою міткою перестає виконуватись. Кожне надання тимчасового доступу фіксується в журналі подій із зазначенням імені користувача та тривалості.

Модуль додавання фізичних носіїв доступу (рисунок 3.8) є ексклюзивним інструментом власника і доступний виключно з екрана адміністрування. Додавання нових карток підтримує два рівноцінні режими. Перший – ручне введення ідентифікатора – призначений для випадків, коли картка вже відома системі або адміністратор має її UID із зовнішнього джерела. Другий – автоматичне перехоплення – власник просить користувача піднести картку до сейфа, після чого ідентифікатор автоматично з’являється в застосунку власника і модальне вікно реєстрації відкривається з попередньо заповненим полем. Власнику залишається лише вказати назву картки та прив’язати її до профілю

користувача зі списку. Після збереження нової картки система автоматично надсилає команду підтвердження на сейф, відкриваючи його без повторного піднесення картки.

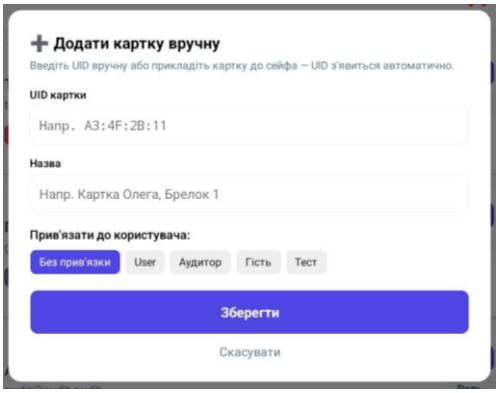


Рисунок 3.8 – Модуль додавання RFID-карток

Модуль журналу забезпечує повну прозорість усіх операцій із сейфом і реалізує двошаровий механізм отримання даних. При відкритті екрана виконується початкове завантаження повної історії подій (рисунок 3.9) у зворотньому хронологічному порядку – із приєднанням профілів авторів для відображення імен замість технічних ідентифікаторів.

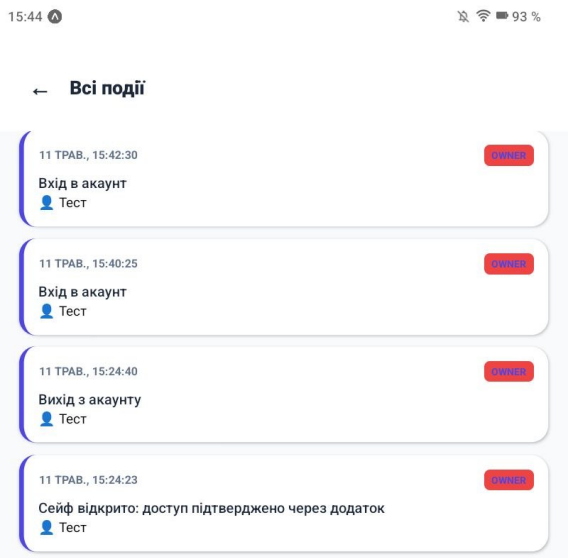


Рисунок 3.9 – Модуль відображення історії подій (журналу)

Після початкового завантаження активується підписка на оновлення в реальному часі. При появі нового запису в базі даних застосунок миттєво отримує сповіщення, асинхронно підвантажує профіль автора події та додає запис на початок списку без повного перезавантаження даних. Це забезпечує миттєве відображення тривожних сповіщень від акселерометра – банер тривоги з'являється на екрані власника (рисунок 3.10) протягом секунди після спрацювання датчика, забезпечуючи можливість оперативного реагування на інциденти. Кожен запис журналу містить точний час події з локалізацією для українського часового поясу, кольоровий бейдж ролі автора та текстовий опис дії. Підписка на оновлення скасовується при закритті екрана, що запобігає накопиченню фонових процесів та дублюванню подій при повторному відкритті журналу.

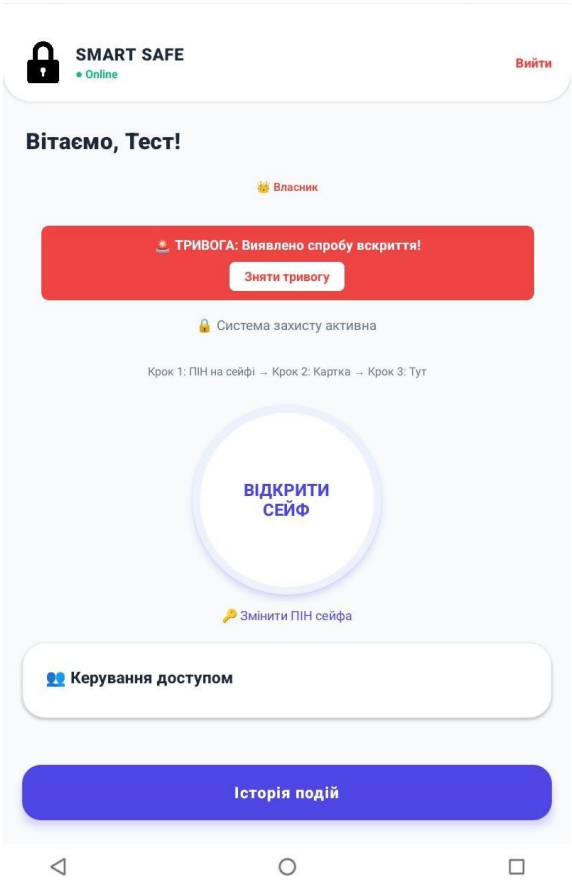


Рисунок 3.10 – Головний екран застосунку у стані тривоги

3.5 Тестування апаратно-програмного комплексу

Метою етапу тестування є перевірка відповідності розробленого апаратно-програмного комплексу визначеним функціональним та безпековим вимогам. Верифікація здійснювалася шляхом проведення серії експериментальних випробувань, що охоплювали перевірку локальної логіки граничного вузла, стабільність мережевого обміну та коректність роботи мобільного застосунку.

Для комплексного оцінювання системи було застосовано метод «чорної скриньки» (black-box testing) для перевірки користувацьких сценаріїв та стрес-тестування для аналізу поведінки системи в умовах апаратних чи програмних збоїв (втрата живлення чи Wi-Fi з’єднання).

Основним етапом тестування стала перевірка алгоритму трифакторної автентифікації. Результати тестування основних функціональних вузлів наведено у таблиці 3.2.

Таблиця 3.2 – Результати функціонального тестування системи

Опис тестової дії	Очікуваний результат	Статус
Введення PIN-коду на матричній клавіатурі	Зміна стану з LOCKED на WAITING_RFID	Пройдено
Прикладання незареєстрованої RFID-мітки	Відмова у доступі, логування UID	Пройдено
Прикладання авторизованої RFID-мітки	Відправка JSON-пакета в топік vovk/safe/auth, стан WAITING_APP	Пройдено
Підтвердження доступу через мобільний додаток	Розблокування сейфа (UNLOCKED), зелена індикація	Пройдено

Особлива увага була приділена точності спрацювання акселерометра MPU-6050. В ході тестування було встановлено, що вибраний програмний поріг чутливості дозволяє ігнорувати незначні вібрації фонового характеру, але впевнено фіксує підняття, нахил або удар по корпусу макета.

При фіксації аномального прискорення система миттєво переходила в стан ALARM. Під час тестування підтверджено, що в режимі тривоги локальні пристрої введення (клавіатура) повністю блокуються, а деактивація звукового сигналу можлива виключно через команду віддаленого адміністратора, що підтверджує стійкість архітектури до спроб фізичного саботажу.

У ході дослідної експлуатації розробленого комплексу було проведено якісну оцінку стабільності програмного забезпечення та надійності мережевої взаємодії. Оскільки в основі граничного вузла лежить принцип неблокуючого опитування периферії, під час тестування не було виявлено затримок інтерфейсу користувача або пропусків подій від акселерометра MPU-6050, що підтверджує коректність обраної алгоритмічної моделі.

3.6 Висновки до третього розділу

У третьому розділі було реалізовано повний цикл програмно-апаратної розробки комплексу, що охоплює рівні граничного пристрою, хмарної інфраструктури та клієнтського інтерфейсу. Обґрунтування вибору технологічного стеку дозволило інтегрувати високоефективні засоби розробки, зокрема екосистему PlatformIO для мікроконтролера ESP32 та фреймворк React Native для мобільного застосунку, що забезпечило кросплатформеність та стабільність зв’язку через протокол MQTT.

Основою мікропрограмної логіки пристрою виступає розроблений алгоритм на базі скінченного автомата станів, який детермінує сувору послідовність проходження етапів трифакторної автентифікації та виключає стани невизначеності при обробці зовнішніх подій. Впроваджена гібридна

стратегія збереження даних забезпечила необхідний баланс між автономністю пристрою завдяки використанню енергонезалежної пам’яті NVS та централізованим аудитом подій у реляційній базі даних. У межах розділу розроблено та оптимізовано безпекову логіку комплексу, що включає створення системи рольового контролю доступу та алгоритму тимчасового делегування доступу.

Завершальний етап верифікації та тестування методом «чорної скриньки» підтвердив відповідність розробленого комплексу технічним вимогам. Експериментальні випробування довели надійність роботи інерціального моніторингу та стабільність взаємодії між апаратною частиною і мобільним клієнтом, що свідчить про готовність системи до практичного застосування у сфері захисту матеріальних цінностей.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень було спроєктовано, розроблено та протестовано макет апаратно-програмного комплексу «Розумний сейф», який базується на концепції розподілених кіберфізичних систем і забезпечує багаторівневий захист матеріальних цінностей від фізичного та логічного втручання.

У першому розділі проведено аналіз еволюції засобів безпеки та існуючих рішень у сфері Інтернету речей. Обґрунтовано необхідність переходу до парадигми нульової довіри (zero-trust) та застосування трифакторної моделі автентифікації (PIN-код, RFID-картка, підтвердження через мобільний застосунок) для усунення вразливостей локального доступу. Визначено, що для надійного просторового моніторингу доцільно використовувати інерціальні MEMS-датчики, а для обміну сповіщеннями в реальному часі оптимальним є асинхронний протокол MQTT.

У другому розділі спроєктовано трирівневу архітектури комплексу, що складається з граничного вузла, хмарного брокера та клієнтського термінала. Обґрунтовано вибір апаратної бази з центральним мікроконтролером ESP32, розроблено електричну принципову схему з розподілом периферії на шинах I2C та SPI. Сформовано математичний апарат для визначення аномальних просторових прискорень (на базі датчика MPU-6050) та розроблено неблокуючу алгоритмічну модель управління на основі скінченного автомата станів, що гарантує високу реактивність системи.

У третьому розділі здійснено повну програмно-апаратну реалізацію комплексу. Написано вбудоване програмне забезпечення для мікроконтролера в середовищі PlatformIO мовою C++ та розроблено кросплатформений мобільний застосунок на базі фреймворку React Native. Спроєктовано реляційну базу даних для ведення безперервного журналу аудиту та управління рольовим доступом. Проведене тестування прототипу підтвердило стабільність MQTT-комунікації,

надійність енергонезалежного збереження PIN-коду, високу точність інерціального моніторингу та загальну ефективність реалізованої системи віддаленого контролю.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. A cross-layer secure and energy-efficient framework for the Internet of things: A comprehensive survey / R. Mustafa et al. *Sensors*. 2024. Vol. 24, no. 22. P. 7209. DOI: https://doi.org/10.3390/s24227209 (дата звернення: 17.02.2026).

2. A survey on edge computing (EC) security challenges: Classification, threats, and mitigation strategies / A. M. Sheikh et al. *Future Internet*. 2025. Vol. 17, no. 4. P. 175. DOI: https://doi.org/10.3390/fi17040175 (дата звернення: 17.02.2026).

3. Ajax Systems: офіційний сайт. URL: https://ajax.systems/ (дата звернення: 19.02.2026).

4. Alluri K., Gopikrishnan S. Enhancing IoT security: A review of multi-factor authentication protocols and their effectiveness. *Smart Innovation, Systems and Technologies*. Singapore, 2025. P. 619–630. DOI: https://doi.org/10.1007/978-981-96-2182-8_46 (дата звернення: 17.02.2026).

5. Alshehri H. Developing multi-factor authentication and biometric verification protocols for enhancing data security in IoT healthcare devices. *2025 17th International Conference on Computer and Automation Engineering (ICCAE)*, Perth, Australia, 20–22 March 2025. 2025. P. 367–373. DOI: https://doi.org/10.1109/iccae64891.2025.10980555 (дата звернення: 17.02.2026).

6. An approach to cloud user access control using behavioral biometric-based authentication and continuous monitoring. *International Journal of Advanced Technology and Engineering Exploration*. 2024. Vol. 11, no. 119. DOI: https://doi.org/10.19101/ijatee.2024.111100516 (дата звернення: 19.02.2026).

7. Ayeswarya S., John Singh K. A comprehensive review on secure biometric-based continuous authentication and user profiling. *IEEE Access*. 2024. P. 1. DOI: https://doi.org/10.1109/access.2024.3411783 (дата звернення: 15.02.2026).

8. Bengheni A. Relay node selection scheme and deep sleep period for power management in energy harvesting wireless sensor networks. *International Journal of*

Communication Systems. 2024. DOI: https://doi.org/10.1002/dac.5742 (дата звернення: 20.02.2026).

9. Borse Pradnya Balasaheb. IoT-driven smart cities: Enhancing attack detection via cloud-based analytics and multifactor authentication. *International Journal of Applied Mathematics*. 2025. Vol. 38, no. 2s. P. 966–984. DOI: https://doi.org/10.12732/ijam.v38i2s.125 (дата звернення: 18.02.2026).

10. Cetintav I., Sandikkaya M. T. A review of lightweight IoT authentication protocols from the perspective of security requirements, computation, communication, and hardware costs. *IEEE Access*. 2025. P. 1. DOI: https://doi.org/10.1109/access.2025.3546147 (дата звернення: 20.02.2026).

11. Deep learning techniques for biometric security: A systematic review of presentation attack detection systems / K. Shaheed et al. Engineering Applications of *Artificial Intelligence*. 2024. Vol. 129. P. 107569. DOI: https://doi.org/10.1016/j.engappai.2023.107569 (дата звернення: 17.02.2026).

12. El-Shafai W., Azar A. T., Ahmed S. AI-driven ensemble classifier for jamming attack detection in VANETs to enhance security in smart cities. *IEEE Access*. 2025. P. 1. DOI: https://doi.org/10.1109/access.2025.3552544 (дата звернення: 25.02.2026).

13. Enhancing the security of firmware over-the-air updates in automotive cyber-physical system / R. Y. Patil et al. *Cyber Physical System 2.0*. Boca Raton, 2024. P. 282–301. DOI: https://doi.org/10.1201/9781003559993-12 (дата звернення: 27.02.2026).

14. Evaluating transport layer security 1.3 optimization strategies for 5G cross-border roaming: A comprehensive security and performance analysis / J. K. Lastre et al. *Sensors*. 2025. Vol. 25, no. 19. P. 6144. DOI: https://doi.org/10.3390/s25196144 (дата звернення: 25.02.2026).

15. Fuzzy vault revisited - enabling privacy-preserving multi-modal biometric authentication / B. Zhang et al. *IEEE Transactions on Dependable and Secure*

Computing. 2025. P. 1–15. DOI: https://doi.org/10.1109/tdsc.2025.3627218 (дата звернення: 17.02.2026).

16. Gutierrez del Arroyo J. A., Borghetti B. J., Temple M. A. Fingerprint extraction through distortion reconstruction (FEDR): A CNN-based approach to RF fingerprinting. *IEEE Transactions on Information Forensics and Security*. 2024. P. 1. DOI: https://doi.org/10.1109/tifs.2024.3463528 (дата звернення: 20.02.2026).

17. Hardware security for Internet of things identity assurance / A. Cirne et al. *IEEE Communications Surveys & Tutorials*. 2024. P. 1. DOI: https://doi.org/10.1109/comst.2024.3355168 (дата звернення: 19.02.2026).

18. Hasan S., Amundson I., Hardin D. Zero-trust design and assurance patterns for cyber-physical systems. *Journal of Systems Architecture*. 2024. Vol. 155. P. 103261. DOI: https://doi.org/10.1016/j.sysarc.2024.103261 (дата звернення: 20.02.2026).

19. IoT-enabled biometric security: enhancing smart car safety with depth-based head pose estimation / C. Bisogni et al. *ACM Transactions on Multimedia Computing, Communications, and Applications*. 2024. DOI: https://doi.org/10.1145/3639367 (дата звернення: 20.02.2026).

20. Jeffrey N., Tan Q., Villar J. R. A hybrid methodology for anomaly detection in cyber-physical systems. *Neurocomputing*. 2024. Vol. 568. P. 127068. DOI: https://doi.org/10.1016/j.neucom.2023.127068 (дата звернення: 17.02.2026).

21. Kondabala R., Balaji S., Anuraghav S. S. Multi-tiered authentication framework for enhanced security in cloud and IoT systems with blockchain. *Service Oriented Computing and Applications*. 2025. DOI: https://doi.org/10.1007/s11761-025-00483-6 (дата звернення: 17.02.2026).

22. Let’s get cyber-physical: Validation of safety-critical cyber-physical systems / L. Novais et al. *IEEE Access*. 2024. P. 1. DOI: https://doi.org/10.1109/access.2024.3470216 (дата звернення: 17.02.2026).

23. Lightweight embedded IoT gateway for smart homes based on an ESP32 microcontroller / F. Serepas et al. *Computers*. 2025. Vol. 14, no. 9. P. 391. DOI: https://doi.org/10.3390/computers14090391 (дата звернення: 18.02.2026).

24. Machine learning-based intelligent security framework for secure cloud key management / S. Ahmad et al. *Cluster Computing*. 2024. DOI: https://doi.org/10.1007/s10586-024-04288-8 (дата звернення: 18.02.2026).

25. Murthy C. K., Taneja A., Singh S. A comprehensive security risk assessment of wireless fidelity protocol with respect to smart homes. *Artificial Intelligence, Computational Intelligence and Inclusive Technologies*. London, 2026. P. 637–643. DOI: https://doi.org/10.1201/9781003740100-79 (дата звернення: 21.02.2026).

26. Park C.-Y., Lee S.-J., Lee I.-G. Secure and lightweight firmware over-the-air update mechanism for Internet of things. *Electronics*. 2025. Vol. 14, no. 8. P. 1583. DOI: https://doi.org/10.3390/electronics14081583 (дата звернення: 20.02.2026).

27. Park J.-H., Park S.-C., Youm H.-Y. A Proposal for a zero-trust-based multi-level security model and its security controls. *Applied Sciences*. 2025. Vol. 15, no. 2. P. 785. DOI: https://doi.org/10.3390/app15020785 (дата звернення: 23.02.2026).

28. Post-quantum security framework for resource-constrained systems: Emerging trends, challenges, sustainability, and future directions / N. Shirisha et al. *Discover Computing*. 2026. Vol. 29, no. 1. DOI: https://doi.org/10.1007/s10791-026-09982-2 (дата звернення: 25.02.2026).

29. Pramana A. L., Ulumuddin M. F., Farizki D. A. Design of an edge computing-based security system with multimodal face and voice biometric authentication. *Journal of Informatics, Game Engineering, and Networking*. 2026. Vol. 1, no. 1. P. 42–54. DOI: https://doi.org/10.55732/4ywpes88 (дата звернення: 20.03.2026).

30. Quantum key distribution networks - key management: A survey / E. Dervisevic et al. *ACM Computing Surveys*. 2025. DOI: https://doi.org/10.1145/3730575 (дата звернення: 17.02.2026).

31. Real-time anomaly detection and threat mitigation in IoT networks using convolutional neural networks (CNNs) for enhanced security / J. Singh et al. 2024 *Second International Conference on Advanced Computing & Communication Technologies (ICACCTech)*, Sonipat, India, 16–17 November 2024. 2024. P. 619-624. DOI: https://doi.org/10.1109/icacctech65084.2024.00104 (дата звернення: 17.02.2026).

32. Reis M. J. C. S., Serdio C. Edge AI for real-time anomaly detection in smart homes. *Future Internet*. 2025. Vol. 17, no. 4. P. 179. DOI: https://doi.org/10.3390/fi17040179 (дата звернення: 10.03.2026).

33. Research on smart-locks cybersecurity and vulnerabilities / C. Caballero-Gil et al. *Wireless Networks*. 2023. DOI: https://doi.org/10.1007/s11276-023-03376-8 (дата звернення: 10.03.2026).

34. Sasikumar K., Nagarajan S. Enhancing cloud security: A multi-factor authentication and adaptive cryptography approach using machine learning techniques. *IEEE Open Journal of the Computer Society*. 2025. P. 1–12. DOI: https://doi.org/10.1109/ojcs.2025.3538557 (дата звернення: 10.03.2026).

35. Securing access: a multi-modal biometric door lock system with arduino and three-factor authentication / Y. D. Patil et al. *Intelligent Computing and Communication Techniques*. London, 2025. P. 419–424. DOI: https://doi.org/10.1201/9781003530190-59 (дата звернення: 12.03.2026).

36. Securing the future: exploring post-quantum cryptography for authentication and user privacy in IoT devices / K. Mansoor et al. *Cluster Computing*. 2024. Vol. 28, no. 2. DOI: https://doi.org/10.1007/s10586-024-04799-4 (дата звернення: 13.03.2026).

37. Securing UAV networks: A lightweight chaotic-frequency hopping approach to counter jamming attacks / C. Atheeq et al. *IEEE Access*. 2024. P. 1. DOI: https://doi.org/10.1109/access.2024.3375343 (дата звернення: 13.03.2026).

38. SLS: a novel RFID based smart locking system / S. Sarangi et al. *2024 6th International Conference on Computational Intelligence and Networks (CINE)*,

Bhubaneswar, India, 19–21 December 2024. 2024. P. 1–6. DOI: https://doi.org/10.1109/cine63708.2024.10881250 (дата звернення: 20.03.2026).

39. Smart home security: an efficient multi-factor authentication protocol / G. Sarbishaei et al. *IEEE Access*. 2024. P. 1. DOI: https://doi.org/10.1109/access.2024.3437294 (дата звернення: 20.03.2026).

40. Smart key protection using LIS3DH accelerometer / R. Beegam S et al. 2025 *Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*, Ernakulam, India, 18–19 July 2025. 2025. P. 1–6. DOI: https://doi.org/10.1109/accthpa65749.2025.11168524 (дата звернення: 15.03.2026).

41. Soni L., Chandra H., Gupta D. S. LB-RFID: Provably secure post-quantum authentication protocol for RFID devices in resource-constrained IoT environment. *Wireless Personal Communications*. 2025. DOI: https://doi.org/10.1007/s11277-025-11847-8 (дата звернення: 16.03.2026).

42. The evolution of biometric authentication: A deep dive into multi-modal facial recognition: A review case study / M. Abdul-Al et al. *IEEE Access*. 2024. P. 1. DOI: https://doi.org/10.1109/access.2024.3486552 (дата звернення: 18.03.2026).

43. Tsai K.-Y., Wei Y.-L., Chi P.-S. Lightweight privacy-protection RFID protocol for IoT environment. *Internet of Things*. 2025. P. 101490. DOI: https://doi.org/10.1016/j.iot.2025.101490 (дата звернення: 18.03.2026).

44. Паритет-К: офіційний сайт. URL: https://paritet-k.com.ua/ (дата звернення: 03.04.2026).

45. Інтернет-магазин ROZETKA: офіційний сайт. URL: https://rozetka.com.ua/ (дата звернення: 08.04.2026).

46. Інтернет-магазин Arduino.ua: офіційний сайт. URL: https://arduino.ua/ (дата звернення: 08.04.2026).

47. Bamashmos S., Chilamkurti N., Shahraki A.S. Two-layered multi-gactor authentication using decentralized blockchain in an IoT environment. *Sensors*. 2024. Vol. 24. P. 3575. DOI: https://doi.org/10.3390/s24113575 (дата звернення: 27.04.2026).

48. Alotaibi A., Aldawghan H., Aljughaiman A. A review of the authentication techniques for internet of things devices in smart cities: opportunities, challenges, and future directions. *Sensors*. 2025. Vol. 25, no. 6. P. 1649. DOI: https://doi.org/10.3390/s25061649 (дата звернення: 27.04.2026).

49. Dhanushkodi K., Thejas S. AI enabled threat detection: Leveraging artificial intelligence for advanced security and cyber threat mitigation. *IEEE Access*. 2024. Vol. 12. P. 173127–173136. DOI: https://doi.org/10.1109/ACCESS.2024.3493957 (дата звернення: 29.04.2026).

50. Permana K. A. K., Piarsa I. N., Wiranatha A. A. K. A. C. IoT-based smart Door Lock System with Fingerprint and Keypad Access. *Journal of Information Systems and Informatics*. 2024. Vol. 6, no. 3. P. 2086–2098. DOI: https://doi.org/10.51519/journalisi.v6i3.844 (дата звернення: 29.04.2026).

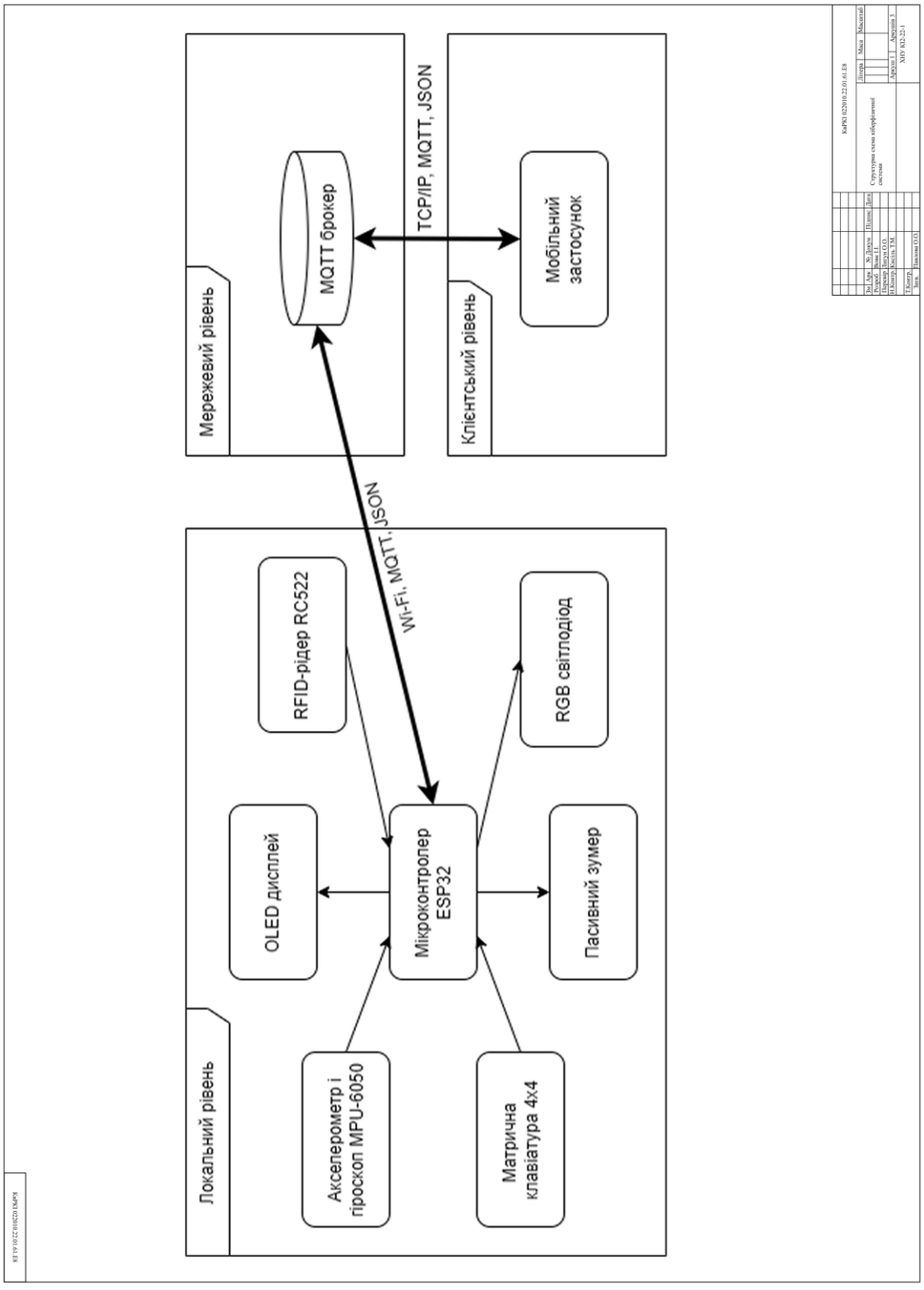
51. Інтернет-магазин Замов двері: офіційний сайт. URL: https://zamovdveri.com.ua/ (дата звернення: 04.05.2026).

52. Інтернет-магазин Імперія сейфів: офіційний сайт. URL: https://safe.com.ua/ua/ (дата звернення: 04.05.2026).

ДОДАТОК А

(обов’язковий)

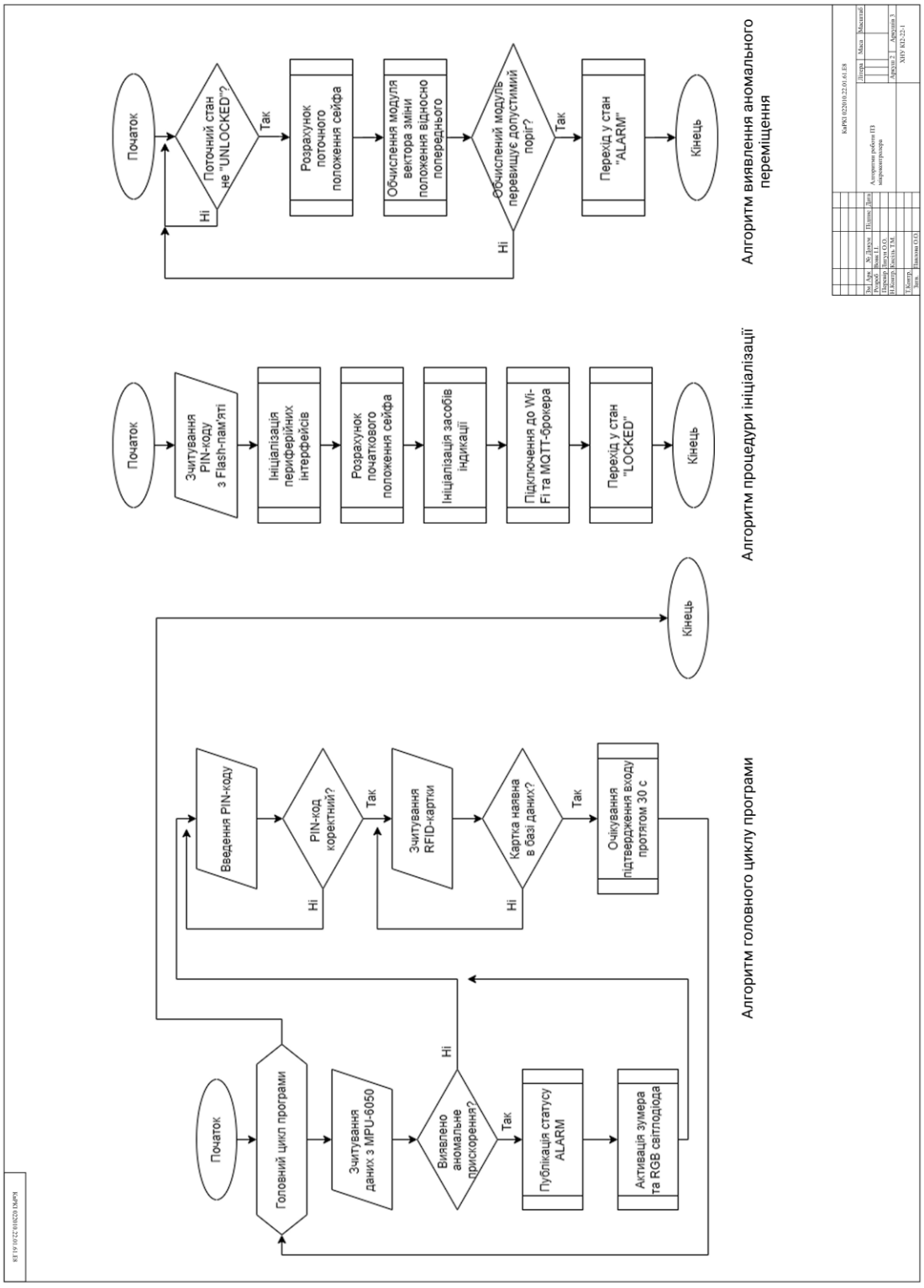
Копія креслення «Структурна схема кіберфізичної системи»



ДОДАТОК Б

(обов’язковий)

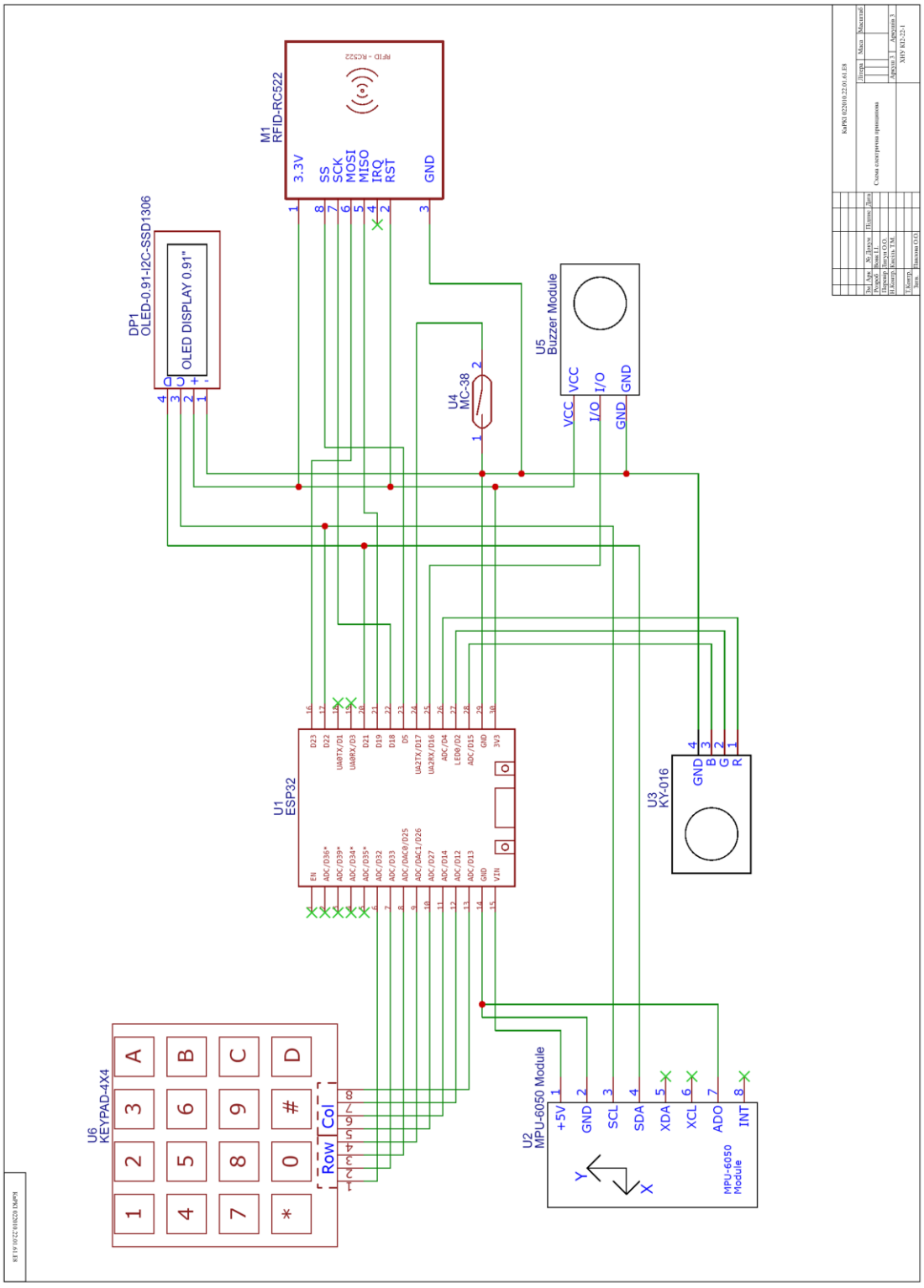
Копія креслення «Алгоритми роботи ПЗ мікроконтролера»



ДОДАТОК В

(обов’язковий)

Копія креслення «Схема електрична принципова»



ДОДАТОК Г

(обов’язковий)

Лістинг коду програмного забезпечення мікроконтролера

```
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Keypad.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <Preferences.h>
#include <math.h>

enum SafeState { LOCKED, WAITING_RFID, WAITING_APP, UNLOCKED };
SafeState currentState = LOCKED;
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
Adafruit_MPU6050 mpu;
MFRC522 rfid(5, -1);
Preferences preferences;
const int PIN_BUZZER = 16, PIN_REED = 17;
const int PIN_RGB_R = 4, PIN_RGB_G = 2, PIN_RGB_B = 15;
byte rowPins[4] = {32, 33, 25, 26};
byte colPins[4] = {27, 14, 12, 13};
char keys[4][4] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, 4, 4);

const char* ssid = "Verizon-MiFi6620L-DDC1";
const char* password = "378f00ba";
const char* mqtt_server = "broker.mqttdashboard.com";
WiFiClient espClient;
PubSubClient client(espClient);

String masterCode;
String inputCode = "";
bool alarmActive = false;
unsigned long lastAlarmToggle = 0;
float lastX = 0, lastY = 0, lastZ = 0;
float lastPitch = 0, lastRoll = 0;
```

```
const float angleThreshold = 5.0;
const float sensitivity = 1.8;
unsigned long lastSampleTime = 0;
unsigned long waitingAppSince = 0;
const unsigned long APP_CONFIRM_TIMEOUT = 30000;

void updateUI(String title, String content);
void setRGB(int r, int g, int b);
void triggerAlarm();
void setup_wifi();
void reconnect();
void beep(int duration);

String getCardUID() {
  String uid = "";
  for (byte i = 0; i < rfid.uid.size; i++) {
    if (rfid.uid.uidByte[i] < 0x10) uid += "0";
    uid += String(rfid.uid.uidByte[i], HEX);
    if (i < rfid.uid.size - 1) uid += ":";
  }
  uid.toUpperCase();
  return uid;
}

void callback(char* topic, byte* payload, unsigned int length) {
  StaticJsonDocument<300> doc;
  deserializeJson(doc, payload, length);
  const char* command = doc["command"];

  if (String(command) == "app_confirm" && currentState == WAITING_APP)
{
    currentState = UNLOCKED;
    updateUI("3-STEP OK", "WELCOME");
    setRGB(0, 255, 0);
    beep(100); delay(100); beep(100);
    client.publish("vovk/safe/status", "UNLOCKED");
  }

  if (String(command) == "app_deny") {
    currentState = LOCKED;
    updateUI("DENIED", "LOCKED");
    setRGB(255, 0, 0);
    beep(500);
    delay(2000);
    setRGB(0, 0, 255);
    updateUI("LOCKED", "ENTER PIN");
  }

  if (String(command) == "guest_unlock") {
    currentState = UNLOCKED;
    updateUI("GUEST ACCESS", "OPENED");
    setRGB(0, 255, 0);
```

```
    beep(300);
    client.publish("vovk/safe/status", "UNLOCKED");
  }

  if (String(command) == "alarm_off") {
    alarmActive = false;
    digitalWrite(PIN_BUZZER, HIGH);
    setRGB(0, 0, 255);
    updateUI("ALARM OFF", "SECURE");
    client.publish("vovk/safe/status", "ALARM_STOPPED");
  }

  if (String(command) == "new_pin") {
    const char* p = doc["value"];
    if (p && strlen(p) >= 4) {
      masterCode = String(p);
      preferences.begin("safe-config", false);
      preferences.putString("pin", masterCode);
      preferences.end();
      beep(200);
    }
  }
}

void setup() {
  Serial.begin(115200);

  preferences.begin("safe-config", false);
  masterCode = preferences.getString("pin", "1234");
  preferences.end();

  Wire.begin(21, 22);
  SPI.begin();
  rfid.PCD_Init();

  if (mpu.begin(0x68)) {
    sensors_event_t a, g, temp;
    mpu.getEvent(&a, &g, &temp);
    lastX = a.acceleration.x; lastY = a.acceleration.y; lastZ =
a.acceleration.z;
  }

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  pinMode(PIN_RGB_R,     OUTPUT);     pinMode(PIN_RGB_G,     OUTPUT);
pinMode(PIN_RGB_B, OUTPUT);
  pinMode(PIN_BUZZER, OUTPUT); digitalWrite(PIN_BUZZER, HIGH);
  pinMode(PIN_REED, INPUT_PULLUP);

  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
```

```
  updateUI("Smart Safe", "READY");
  setRGB(0, 0, 255);
}

void loop() {
  if (!client.connected()) reconnect();
  client.loop();

  char key = keypad.getKey();

  if (currentState != UNLOCKED && millis() > 10000) {
    if (millis() - lastSampleTime > 200) {
      sensors_event_t a, g, temp;
      mpu.getEvent(&a, &g, &temp);

      float A = sqrt(sq(a.acceleration.x) + sq(a.acceleration.y) +
sq(a.acceleration.z));

      float pitch = atan2(a.acceleration.y, a.acceleration.z) *
RAD_TO_DEG;
      float roll = atan2(-a.acceleration.x, sqrt(sq(a.acceleration.y)
+ sq(a.acceleration.z))) * RAD_TO_DEG;

      bool isTampered = false;

      if (abs(A - 9.81) > sensitivity) {
        isTampered = true;
      }

      if (abs(pitch - lastPitch) > angleThreshold || abs(roll -
lastRoll) > angleThreshold) {
        isTampered = true;
      }

      if (isTampered && !alarmActive) {
        alarmActive = true;
        client.publish("vovk/safe/alarm", "TAMPER DETECTED");
      }

      lastPitch = pitch;
      lastRoll = roll;
      lastSampleTime = millis();
    }
  }

  if (alarmActive) triggerAlarm();

  if (currentState == WAITING_APP && (millis() - waitingAppSince >
APP_CONFIRM_TIMEOUT)) {
    currentState = LOCKED;
    setRGB(0, 0, 255);
    updateUI("TIMEOUT", "ENTER PIN");
```

```
  client.publish("vovk/safe/status", "LOCKED");
}

if (key) {
  if (!alarmActive) beep(40);
  if (key == '*') {
    currentState = LOCKED; inputCode = "";
    setRGB(0, 0, 255); updateUI("LOCKED", "ENTER PIN");
  }
  else if (key == '#') {
    if (inputCode == masterCode) {
      currentState = WAITING_RFID;
      setRGB(255, 165, 0); updateUI("PIN OK", "SCAN CARD");
    } else {
      updateUI("WRONG PIN", "ERROR");
      setRGB(255, 0, 0); delay(1000); setRGB(0, 0, 255);
      updateUI("LOCKED", "ENTER PIN");
    }
    inputCode = "";
  } else {
    if (inputCode.length() < 8) inputCode += key;
    String maskedCode = "";
    for (int i = 0; i < inputCode.length(); i++) maskedCode += "*";
    updateUI("PIN:", maskedCode);
  }
}

if (currentState == WAITING_RFID) {
  if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
    beep(100);
    String uid = getCardUID();
    currentState = WAITING_APP;
    waitingAppSince = millis();
    updateUI("CARD OK", "CHECK APP...");
    setRGB(255, 255, 255);

    StaticJsonDocument<100> doc;
    doc["uid"] = uid;
    char buf[100];
    serializeJson(doc, buf);
    client.publish("vovk/safe/auth", buf);
    rfid.PICC_HaltA();
  }
}

if (currentState == UNLOCKED && digitalRead(PIN_REED) == LOW) {
  delay(1000);
  if (digitalRead(PIN_REED) == LOW) {
    currentState = LOCKED;
    setRGB(0, 0, 255);
    updateUI("CLOSED", "LOCKED");
    client.publish("vovk/safe/status", "LOCKED");
```

```
    }
  }
}

void setup_wifi() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi Connected");
}

void reconnect() {
  while (!client.connected()) {
    if (client.connect("ESP32Safe_Vovk_Client")) {
      client.subscribe("vovk/safe/control");
    } else {
      delay(3000);
    }
  }
}

void updateUI(String title, String content) {
  display.clearDisplay();
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0,          0);          display.setTextSize(1);
display.println(title);
  display.setCursor(0,         15);          display.setTextSize(2);
display.println(content);
  display.display();
}

void setRGB(int r, int g, int b) {
  analogWrite(PIN_RGB_R,     r);     analogWrite(PIN_RGB_G,     g);
analogWrite(PIN_RGB_B, b);
}

void triggerAlarm() {
  if (millis() - lastAlarmToggle > 250) {
    lastAlarmToggle = millis();
    digitalWrite(PIN_BUZZER, !digitalRead(PIN_BUZZER));
    setRGB(255, 0, 0);
    updateUI("!!!ALARM!!!", "DANGER");
  }
}

void beep(int duration) {
  digitalWrite(PIN_BUZZER, LOW);
  delay(duration);
  digitalWrite(PIN_BUZZER, HIGH);
}
```

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: Іван ВОВК

Співавтор:

Назва: Апаратно-програмний комплекс «Розумний сейф» з багатофакторною автентифікацією та захистом від несанкціонованого переміщення

Експерт: Олексій ЛИГУН

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:2.79%

Коефіцієнт подібності 2:0.73%

Мікропробіли: 3

Заміна букв: 1

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-05-21 11:16:41.0

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:

2026-05-21 Доцент Андрій Нічепорук

Дата експерт

Anti-Plagiarism (http://ap.km.ua) v-15.701

Максимальне співпадіння з одним документом 5.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 16%**

	Документ		Сумарний збіг по Базі Даних	
ID: 271888 Назва: БКР Апаратно-програмний комплекс «Розумний сейф» з багатофакторною автентифікацією та захистом від несанкціонованого переміщення Додано в БД: 2026-05-21 Автора: Іван ВОВК Керівники: Олексій ЛИГУН Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	91730	648	5901 (6%)	72 (11%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Вовк Іван Іванович

Тема: Апаратно-програмний комплекс «Розумний сейф» з багатофакторною автентифікацією та захистом від несанкціонованого переміщення

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 62

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є проєктування апаратно-програмного комплексу «Розумний сейф» з багатофакторною автентифікацією та захистом від несанкціонованого переміщення.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз еволюції засобів безпеки та існуючих рішень у сфері Інтернету речей. Обґрунтовано необхідність переходу до парадигми нульової довіри та застосування трифакторної моделі автентифікації для усунення вразливостей локального доступу. У другому розділі спроєктовано трирівневу архітектуру комплексу, що складається з граничного вузла, хмарного брокера та клієнтського термінала. Обґрунтовано вибір апаратної бази з центральним мікроконтролером ESP32, розроблено електричну принципову схему з розподілом периферії на шинах I2C та SPI. Сформовано математичний апарат для визначення аномальних просторових прискорень та розроблено неблокуючу алгоритмічну модель управління на основі скінченного автомата станів, що гарантує високу реактивність системи. У третьому розділі здійснено повну програмно-апаратну реалізацію комплексу. Написано вбудоване програмне забезпечення для мікроконтролера в середовищі PlatformIO мовою C++ та розроблено кросплатформений мобільний застосунок на базі фреймворку React Native, а також спроєктовано реляційну базу даних для ведення безперервного

журналу аудиту та управління рольовим доступом. Проведено тестування прототипу, яке підтвердило стабільність, надійність та загальну ефективність реалізованої системи.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: -.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

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

8. Інші зауваження: _____

9. Оцінка дипломної роботи: відмінно (A / 93).

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Мартинюк Валерій Валерійович, д.т.н., проф.; професор кафедри АКІТеР

"28" травня 2026 р. _____ (підпис)

Зав. кафедри КІС
д-р. філософії Ользі ПАВЛОВІЙ

Іван ВОВК
ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

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

1 травня 2026 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Апаратно-програмний комплекс «Розумний сейф» з багатофакторною автентифікацією та захистом від несанкціонованого переміщення
Автор Іван ВОВК
Освітня програма Комп'ютерна інженерія та програмування
Рівень вищої освіти перший (бакалаврський)
Спеціальність 123 Комп'ютерна інженерія
Науковий керівник: Олексій ЛИГУН

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	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

2) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 3,52%; та системою Anti-Plagiarism складає 5%, що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

01.06.2026

Завідувач кафедри _____ Підпис Ольга ПАВЛОВА Ім'я, ПРІЗВИЩЕ

Гарант освітньої програми _____ Підпис Андрій НІЧЕПОРУК Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи _____ Підпис Олексій ЛИГУН Ім'я, ПРІЗВИЩЕ