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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Система автоматичного контролю доступу до приміщень за допомогою біометричних даних на базі мікроконтролеру Arduino ATMega328
Назва теми

КвРКІ 210248.21.02.09 ПЗ
Шифр

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

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

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

Виконав: студент IV курсу, група КІ2-22-1 _____ Артем САХНО
Підпис Ініціали, прізвище

Керівник _____ Олег САВЕНКО
Підпис, дата Ініціали, прізвище

Нормоконтролер _____ Тетяна КИСІЛЬ
Підпис, дата Ініціали, прізвище

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

«05» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень бакалавр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

" 10 " 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Артему САХНУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система автоматичного контролю доступу до приміщень за допомогою біометричних даних на базі мікроконтролера Arduino ATMega328

Керівник проекту (роботи) Олег САВЕНКО, д.т.н., проф.

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

Затверджена наказом ректора університету від 07.02.2025 р. № 23

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

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

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

Аналіз предметної області та формулювання задачі дослідження

Вибір та обгрунтування компонентів системи контролю доступу

Проєктування та програмна реалізація системи контролю доступу

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

Схема електрична принципова

Алгоритмічне забезпечення системи

Алгоритм ідентифікації користувача

6. Консультанти розділів дипломного проекту (роботи)

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

7. Дата видачі завдання «_10_»_____01_____2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Приміт
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконан
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконан
3	Робота над розділом 1 – аналіз предметної області та формулювання задачі дослідження	01.03.2025	виконан
4	Робота над розділом 2 – вибір та обгрунтування компонентів системи контролю доступу	01.04.2025	виконан
5	Робота над розділом 3 – проєктування та програмна реалізація системи контролю доступу	29.04.2025	виконан
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконан
7	Попередній захист ВКР	26.05.2025	виконан
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент _____ Артем САХНО
Підпис Ініціали, прізвище

Керівник роботи _____ Олег САВЕНКО
Підпис Ініціали, прізвище

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 210248.21.02.09 ПЗ	Пояснювальна записка	83		
2		КвРКІ 210248.21.02.09 Е8	Схема електрична принципова	1		
3		КвРКІ 210248.21.02.09 Е8	Алгоритмічне забезпечення системи	1		
4		КвРКІ 210248.21.02.09 Е8	Алгоритм ідентифікації користувача			

Зм	Арк	№ докум	Підпис	Дата	КвРКІ 210248.21.02.09 ПЗ		
Розробив		Сахно		06.06.25	Система автоматичного контролю доступу до приміщень за допомогою біометричних даних на базі мікроконтролеру Arduino ATMega328	Літера	Аркуш	Аркушів
Перевір.		Савенко		06.06.25		У	1	83
Н. контр.		Кисіль		06.06.25		ХНУ, КІ2-21-2		
Затв.		Павлова		06.06.25				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система автоматичного контролю доступу до приміщень за допомогою біометричних даних на базі мікроконтролеру Arduino ATMega328».

Автор роботи: Сахно Артем Євгенович.

Керівник роботи: Савенко Олег Станіславович.

Пояснювальна записка: 83 с., 30 рис., 2 табл., 3 дод., 78 джерел.

Графічна частина: 3 креслення.

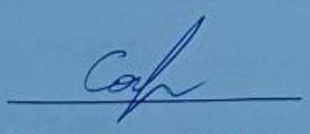
Система контролю доступу, біометрична ідентифікація, відбитки пальців, мікроконтролер Arduino, клієнт-серверна архітектура, PosgreSQL.

Метою кваліфікаційної роботи є розробка та експериментальне дослідження функціональних характеристик системи автоматичного контролю доступу до приміщень, що базується на мікроконтролері Arduino ATMega328 та використовує біометричні дані відбитків пальців для ідентифікації користувачів.

Об'єктом дослідження є система автоматичного контролю та управління доступом до приміщень, що реалізована на базі мікроконтролера Arduino ATMega328.

Предметом дослідження є архітектурні рішення та програмно-апаратна реалізація системи біометричного контролю доступу, що використовує мікроконтролер Arduino ATMega328 та серверний компонент для обробки даних.

Під час проведення даного дослідження застосовувався комплекс методів, що включав аналіз науково-технічних джерел для формування теоретичної бази та огляду існуючих рішень у сфері біометричного контролю доступу.



Підпис студента

30.05.2025

Дата

ЗМІСТ

ВСТУП 3

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМУЛЮВАННЯ ЗАДАЧІ ДОСЛІДЖЕННЯ 4

1.1 Аналіз сучасних систем контролю доступу 4

1.2 Біометричні технології у системах безпеки 10

1.3 Постановка задачі дослідження 18

1.4 Висновок 24

2 ВИБІР ТА ОБГРУНТУВАННЯ КОМПОНЕНТІВ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ 27

2.1 Аналіз та вибір компонентів мікроконтролерної системи 27

2.2 Вибір та обгрунтування бази даних для зберігання біометричних даних 35

2.3 Архітектура та принцип дії системи контролю доступу 44

2.4 Висновок 49

3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ 51

3.1 Електрична принципова схема та архітектура програмно-апаратного комплексу 51

3.2 Розробка алгоритму роботи системи 57

3.3 Програмна реалізація системи на базі мікроконтролера Arduino ATMega328 73

3.4 Висновок 80

ВИСНОВОК 82

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 84

ДОДАТОК А 93

ДОДАТОК Б 94

ДОДАТОК В 95

ДОДАТОК Г 96

ВСТУП

У сучасному світі безпека інформаційних та фізичних об’єктів є одним із найважливіших аспектів технологічного розвитку. Автоматизовані системи контролю доступу (СКД) набувають все більшого поширення, оскільки забезпечують надійний захист приміщень від несанкціонованого проникнення. Одним із найефективніших методів ідентифікації користувачів є біометрична автентифікація, що грунтується на унікальних фізіологічних характеристиках людини, таких як відбитки пальців, зображення обличчя або райдужної оболонки ока.

Використання мікроконтролера Arduino ATMega328 у якості апаратної основи забезпечує гнучкість та доступність рішення, що сприяє його широкому застосуванню в малих та середніх об’єктах безпеки. Інтеграція біометричних датчиків із програмними алгоритмами обробки дозволяє реалізувати високоточний механізм розпізнавання користувачів.

Актуальність даної роботи полягає у необхідності створення ефективної, надійної та економічно доцільної системи контролю доступу, яка використовує біометричні дані для ідентифікації осіб. Дослідження спрямоване на розробку та аналіз програмно-апаратного комплексу, що грунтується на мікроконтролері Arduino ATMega328, з метою підвищення рівня безпеки приміщень та мінімізації ризиків несанкціонованого доступу. У ході роботи буде розглянуто принципи функціонування біометричних технологій, особливості інтеграції апаратних компонентів та оптимізацію алгоритмів розпізнавання.

Таким чином, дана кваліфікаційна робота має на меті дослідження можливостей застосування мікроконтролерних платформ у сфері безпеки, а також розробку ефективного рішення для автоматизованого контролю доступу до приміщень на основі біометричних даних. Отримані результати можуть бути корисними для подальших досліджень та практичного впровадження в системи безпеки різного рівня складності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМУЛЮВАННЯ ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз сучасних систем контролю доступу

Сучасні системи контролю та управління доступом (СКУД) – невід'ємний елемент безпеки більшості об'єктів. Вони забезпечують контрольований доступ до фізичних територій чи інформаційних ресурсів шляхом ідентифікації та автентифікації суб'єкта, приймаючи рішення про надання або обмеження доступу.

Реалізація такого контролю є критично важливою для захисту матеріальних та інформаційних активів, забезпечення безпеки персоналу та відвідувачів, а також відповідності встановленим нормативним вимогам об'єкта. Це формує фундаментальну основу системи безпеки будь-якого підприємства чи установи.

Ринок СКУД технологічно різноманітний, пропонуючи рішення на різних принципах ідентифікації та автентифікації. Кожна технологія має свої переваги та недоліки. Вибір оптимальної системи вимагає розуміння цих відмінностей та аналізу потреб об'єкта.

Еволюція систем контролю доступу (СКУД) йшла від простих механічних та електромеханічних замків до систем на магнітних картках та RFID-технологіях, що підвищували зручність та гнучкість, але мали вразливості до копіювання [2]. Ця еволюція є безперервним процесом, що рухається прагненням підвищити рівень безпеки, забезпечити більшу зручність використання та керованість, а також адаптуватися до появи нових загроз та технологічних можливостей. Кожен етап розвитку систем відповідав актуальним викликам свого часу.

Якісно новий етап розпочався з біометричних технологій (відбитки пальців, обличчя, райдужка ока), які ідентифікують за унікальними фізіологічними ознаками [1]. Це значно підвищило надійність, оскільки біометричні дані складніше підробити чи викрасти порівняно з традиційними ідентифікаторами [2].

Сучасні СКУД рухаються до багатофакторної автентифікації та інтелектуальних систем з ШІ та хмарними технологіями [1].

Основні біометричні методи: сканування відбитків пальців [3, 4], розпізнавання обличчя [5, 6] та сканування райдужки ока [5, 7].

Попри переваги, біометричні СКУД мають недоліки: висока вартість (особливо для точних методів) та вимоги до обладнання [7]. Критичним питанням залишається конфіденційність та захист незмінних біометричних даних відповідно до законодавства [1, 8].

Платформи на базі мікроконтролерів, зокрема Arduino на мікроконтролері ATMega328, стали важливим інструментом для розробки систем контролю доступу. Це зумовлено рядом факторів, які роблять їх особливо привабливими для прототипування, навчання та розробки бюджетних рішень. Оскільки технології на платформі Arduino є доступними та широко підтримуються спільнотою, вони дозволяють швидко створювати прототипи й реалізовувати концепції в СКУД. Використання таких платформ значно знижує бар'єри для експериментів та розробки, дозволяючи швидко тестувати нові ідеї та реалізовувати спеціалізовані рішення для контролю доступу. Це сприяє інноваціям та поширенню знань у галузі вбудованих систем безпеки.

По-перше, економічна доступність платформи Arduino є однією з основних причин, чому вона широко використовується в дослідницьких цілях та для створення рішень з обмеженим бюджетом. Плата Arduino коштує недорого, а також існує великий вибір сумісних модулів та сенсорів, що дозволяє значно знизити витрати на апаратну частину системи [3, 6].

По-друге, величезна підтримка спільноти сприяє розвитку проєктів. Завдяки великій кількості онлайн-ресурсів, готових бібліотек, прикладів коду та форумів, навіть початківці можуть швидко освоїти роботу з платформою. Це прискорює процес розробки та допомагає уникнути багатьох проблем, з якими часто стикаються новачки в програмуванні і проєктуванні [9, 10].

Платформа Arduino є надзвичайно гнучкою, а середовище розробки Arduino IDE підтримує мову програмування C/C++, що дозволяє програмістам швидко писати код для різноманітних сенсорів та виконавчих механізмів. Простота

програмування знижує поріг входу для нових розробників, тому навіть студенти можуть розпочати роботу з Arduino без необхідності в глибоких знаннях в галузі мікроконтролерів [6].

Мікроконтролер ATMega328 має низку вбудованих інтерфейсів, таких як SPI, I2C, UART, що дозволяє підключати різноманітні периферійні пристрої. Це включає біометричні сенсори, RFID-модулі, дисплеї, релейні модулі для керування замками та інші компоненти, необхідні для створення повноцінних систем контролю доступу. Завдяки достатній кількості портів вводу/виводу (GPIO), система може бути легко адаптована до різних вимог [3, 9].

В контексті створення СКУД з Arduino часто використовуються біометричні сенсори (наприклад, сканери відбитків пальців) та RFID-модулі для безконтактної ідентифікації. Це дозволяє інтегрувати різні методи автентифікації в одну систему, що підвищує її ефективність та зручність використання. Сканери відбитків пальців, як правило, мають хороше співвідношення ціни та точності для багатьох практичних застосувань, а RFID дає можливість швидко і безконтактно здійснювати автентифікацію за допомогою карт або брелоків [6, 9].

Незважаючи на всі переваги, використання Arduino має певні обмеження. Одним із основних є обмежена пам’ять (як оперативна, так і флеш-пам’ять), що ставить обмеження на складність алгоритмів обробки біометричних даних та розмір бази даних. Для великих баз даних або складних обчислень необхідно застосовувати оптимізовані алгоритми або використовувати зовнішні пристрої пам’яті, такі як карти SD або EEPROM [9, 10].

Таблиця 1.1 – Порівняння Біометричних/RFID Сенсорів для Інтеграції з Arduino

Тип сенсора	Орієнтована вартість	Інтерфейс	Ключові переваги	Ключові обмеження	Приклади моделей
Оптичний сканер відбитка	Низька - Середня	UART	Доступність, простота інтеграції,	Чутливість до забруднень,	Fingerprint Sensor FPM10A

Кінець таблиці 1.1

Ємнісний сканер відбитка	Середня	UART, SPI	Більш стійкий до обману, компактність	Вища вартість, чутливість до статичної електрики	Деякі моделі від Waveshare, SparkFun
RFID Зчитувач (MFRC522)	Дуже Низька	SPI, I2C	Дуже низька вартість, стандарт Mifare Classic	Обмежена безпека Mifare Classic, лише HF (13.56 MHz)	RC522 модулі
RFID Зчитувач (EM4100)	Дуже Низька	UART, Wiegand	Простота, низька вартість	Лише читання ID, низька безпека (легко копіюється), LF (125 kHz)	RDM6300, прості зчитувачі

Додатково, обмежена обчислювальна потужність може призвести до затримок у виконанні операцій ідентифікації, що є важливим аспектом для деяких застосувань, де потрібна швидка реакція. У таких випадках часто застосовують спеціалізовані алгоритми для попередньої обробки даних на біометричних сенсорах або переносять частину обчислень на потужніші сервери [3, 6].

Для того щоб забезпечити ефективну роботу навіть при обмежених ресурсах, розробники часто використовують зовнішні пристрої пам’яті, оптимізують алгоритми обробки даних, а також застосовують методи попередньої обробки даних на сенсорах. Це дозволяє значно зменшити навантаження на мікроконтролер і підвищити ефективність системи [9].

Сучасні системи контролю та управління доступом (СКУД) вийшли за межі базового функціоналу, еволюціонуючи до складних інтелектуальних та інтегрованих рішень із фокусом на підвищення рівня безпеки.

Одним із ключових напрямків посилення безпеки є багатофакторна автентифікація (MFA). Вона передбачає комбінування двох або більше різних

ідентифікаційних факторів (наприклад, біометричні дані, фізичний ідентифікатор, PIN-код), що значно ускладнює несанкціонований доступ [1, 2]. Впровадження MFA, однак, пов’язане зі зростанням складності та вартості.

Розвиток у напрямку інтелектуальних та хмарних СКУД базується на використанні можливостей штучного інтелекту та машинного навчання для аналізу поведінкових факторів та виявлення аномалій [3, 4]. Хмарні технології забезпечують централізоване адміністрування, масштабованість та гнучкість [5, 6], проте розгортання таких систем потребує значних інвестицій та висококваліфікованих фахівців.

Перспективним підходом є контекстно-залежний контроль доступу, який приймає динамічні рішення на основі сукупності параметрів поточної ситуації, включаючи час, місцезнаходження, роль користувача та стан пристрою доступу [7, 8]. Зростає популярність мобільного доступу із використанням смартфонів через технології BLE або NFC для зручнішої та більш гігієнічної ідентифікації [9, 10]. Також інтегруються новітні технології, такі як блокчейн для забезпечення цілісності даних про доступ [11, 12] та Інтернет речей (IoT) для взаємодії з іншими системами будівлі [11, 12].

Важливим аспектом сучасних СКУД є їхня інтеграція з іншими системами безпеки та управління будівлею (наприклад, відеоспостереженням, пожежною сигналізацією, BMS) [13]. Це забезпечує розширення функціональних можливостей та підвищення загальної ефективності (візуальна верифікація, автоматична евакуація), але водночас збільшує потенційну поверхню атаки, вимагаючи посиленої уваги до кібербезпеки інтегрованої інфраструктури [13].

Сучасні СКУД, інтегруючись у мережі та використовуючи хмарні і мобільні технології, стають значно складнішими, що виводить питання кібербезпеки на перший план. Потенційні загрози охоплюють несанкціонований доступ до чутливих даних (включаючи біометричні), перехоплення інформації під час передачі та атаки на програмне чи апаратне забезпечення системи з метою отримання контролю або порушення її роботи [14, 15, 16].

Відповідно, розширення функціональних можливостей та інтеграція сучасних СКУД зумовлюють необхідність імплементації надійних заходів кіберзахисту на всіх рівнях архітектури системи для забезпечення її стійкості до актуальних загроз. Питання безпеки вже не обмежується лише фізичним периметром, а поширюється на цифрову інфраструктуру СКУД.

Для протидії цим загрозам необхідний комплексний підхід, що включає забезпечення мережевої безпеки (брандмауери, IDS/IPS) [17], захист кінцевих точок (антивірус, своєчасні оновлення) [16, 17], криптографічний захист даних під час передачі та зберігання, а також управління ключами [15, 16, 18, 19]. Критичною є безпека на апаратному рівні, особливо для вбудованих систем, що передбачає захист від фізичного вилучення даних та модифікації прошивки [15, 18].

Забезпечення належного рівня безпеки вимагає регулярних аудитів та тестування на проникнення протягом життєвого циклу системи [17, 19]. Важливим аспектом є суворе дотримання законодавчих вимог щодо захисту персональних даних (як-от GDPR, Закон України), особливо при обробці біометрії. Це накладає зобов'язання щодо отримання згоди, забезпечення конфіденційності даних та прав суб'єктів, а також впровадження технічних та організаційних заходів захисту [14, 19]. Кібербезпека та нормативна відповідність є фундаментальними передумовами легітимного та надійного функціонування СКУД.

Вибір та впровадження СКУД є стратегічним рішенням без універсального підходу, що залежить від специфічних потреб, ризиків та ресурсів організації [20]. Ключовими критеріями вибору є необхідний рівень безпеки, що відповідає цінності активів [20, 21], загальна вартість володіння (TCO) [22, 23], зручність використання для всіх категорій користувачів [20, 21], масштабованість для адаптації до майбутніх потреб [21], а також можливість інтеграції з іншими системами безпеки та IT-інфраструктурою [22, 23]. Особливо вагомим серед них є аспект зручності. Недостатня увага до зручності може призвести до помилок операторів, а у випадку кінцевих користувачів, то до спроб обходу встановлених процедур, що прямо знижує ефективний рівень безпеки системи [20, 21].

При оцінці системи важливо враховувати не лише початкові витрати, а й ТСО, включаючи обслуговування, оновлення та потенційну модернізацію [23]. Часто недооцінюється складність інтеграції з існуючою IT-інфраструктурою та корпоративними системами (AD, BMS, ERP), що може спричинити труднощі під час впровадження та обмежити функціональність рішення [20, 21].

1.2 Біометричні технології у системах безпеки

Біометрія використовує унікальні фізіологічні або поведінкові характеристики людини для її ідентифікації та автентифікації, що робить її важливим компонентом сучасних систем безпеки [20]. Ці технології здійснюють вимірювання та статистичний аналіз індивідуальних рис з метою надання доступу до систем, пристроїв або даних, а також для виявлення осіб, що перебувають під наглядом [21]. Основна ідея полягає в тому, що кожна людина має неповторні біологічні або поведінкові ознаки, які можна використовувати для автоматичної ідентифікації [20]. Біометричні системи виконують зіставлення отриманих даних з шаблонами, збереженими у базі даних [21]. Їх популярність зростає через потребу у надійних способах підтвердження особи в умовах цифровізації суспільства [22]. У сфері інформаційної безпеки біометрія виступає як один з основних методів автентифікації поряд з паролями та токенами, забезпечуючи більш високий рівень достовірності [21]. Її ключова перевага це здатність встановити особу саме як унікального індивіда, що робить цей підхід більш захищеним у порівнянні з традиційними методами [20].

Еволюція біометричних систем від застосування у високозахищених об’єктах та правоохоронних структурах до масового впровадження у споживчих пристроях демонструє тісний зв’язок між технологічним розвитком (зниження собівартості, мініатюризація, підвищення точності завдяки використанню штучного інтелекту та машинного навчання) та зростаючим попитом на зручність і безпеку [20]. Така динаміка вказує на те, що подальше розширення біометричних рішень значною

мірою залежатиме від здатності розробників досягти оптимального балансу між економічною доцільністю, ефективністю та соціальною прийнятністю [21].

Біометрична ідентифікація грунтується на аналізі унікальних фізіологічних або поведінкових характеристик людини, що дозволяє здійснювати точну ідентифікацію особи. До найпоширеніших методів біометричної аутентифікації належать розпізнавання за відбитками пальців, рисами обличчя, райдужною оболонкою ока, сітківкою, а також геометрією руки. Кожен із зазначених підходів має власні технічні особливості, переваги й обмеження, які впливають на вибір того чи іншого методу в залежності від контексту застосування.

Одним із найдавніших і найпоширеніших методів є розпізнавання за відбитками пальців, яке передбачає сканування та подальший аналіз унікального візерунка папілярних ліній. Цей підхід характеризується високою точністю, оперативністю обробки даних та відносно низькою вартістю реалізації, що сприяє його широкому впровадженню у побутові пристрої, системи контролю доступу та банківські сервіси [23]. Водночас метод є чутливим до механічних пошкоджень шкіри, забруднень, а також потенційно вразливим до атак за допомогою штучних копій.

У свою чергу, розпізнавання обличчя грунтується на обробці зображення та аналізі взаємного розташування ключових рис, а саме очей, носа, рота, контуру обличчя тощо. В сучасних реалізаціях активно використовуються методи глибокого навчання, що дозволяє досягати прийнятного рівня точності навіть за умов варіативного освітлення чи кута огляду [24]. Основною перевагою цього методу є безконтактність та зручність використання, особливо у випадках масового сканування. Проте на ефективність розпізнавання можуть негативно впливати зовнішні фактори, такі як освітлення, наявність окулярів або масок, а також високий рівень подібності рис у деяких індивідів.

Більш високий рівень безпеки забезпечують методи, що використовують анатомічні особливості очей. Зокрема, ідентифікація за райдужною оболонкою базується на унікальності візерунків, які залишаються незмінними протягом життя.

Для зчитування застосовуються інфрачервоні камери, що дозволяє здійснювати розпізнавання за низького рівня освітлення [25]. Такий підхід забезпечує дуже низький рівень хибнопозитивних результатів, однак потребує спеціалізованого обладнання, яке суттєво підвищує загальну вартість системи.

Ще більш складним, але водночас точним є розпізнавання за сітківкою ока. Цей метод базується на аналізі капілярної структури сітківки, що є надзвичайно індивідуальною навіть у однояйцевих близнюків [26]. Сканування передбачає точне позиціонування ока відносно джерела інфрачервоного випромінювання, що знижує зручність використання, проте гарантує найвищий рівень захищеності. Через високу складність і вартість така технологія здебільшого використовується у високобезпечних об’єктах, наприклад, в оборонній або урядовій сферах.

Окрему нішу займає метод розпізнавання на основі геометрії руки, який використовує вимірювання морфологічних параметрів, таких як довжина пальців, ширина долоні, товщина кисті тощо. Хоча ця технологія поступається іншим методам у точності, вона залишається стійкою до незначних зовнішніх змін, наприклад, дрібних ушкоджень шкіри [27]. Такий метод зазвичай застосовується у поєднанні з іншими системами аутентифікації, наприклад, у системах контролю доступу в офісах або на виробничих об’єктах.

Оцінка ефективності біометричних систем є ключовим аспектом при їх виборі та впровадженні. Точність є першочерговим показником, який кількісно визначається за допомогою двох основних метрик помилок. Коефіцієнт помилкового допуску (False Acceptance Rate, FAR), що також відомий як помилка другого роду (Type II error), вимірює ймовірність того, що система помилково ідентифікує неавторизованого користувача як авторизованого. Зменшення значення FAR свідчить про вищий рівень безпеки системи, оскільки це знижує ризик несанкціонованого доступу [26].

Коефіцієнт помилкової відмови (False Rejection Rate, FRR), відомий як помилка першого роду (Type I error), вимірює ймовірність того, що система помилково відхиляє запит від авторизованого користувача. Зменшення значення

FRR підвищує зручність використання системи, оскільки знижує кількість випадків, коли легітимний користувач не може отримати доступ [27].

Таблиця 1.2 – Порівняльна таблиця біометричних методів

Біометричний метод	Переваги	Недоліки	Сфери застосування
Відбитки пальців	Висока точність, низька вартість	Можливі проблеми із забрудненням сенсора	Ідентифікація в системах доступу
Розпізнавання обличчя	Швидкість, безконтактність	Висока чутливість до умов освітлення та кута	Безпека смартфонів, контроль доступу
Райдужна оболонка ока	Висока точність, важко підробити	Висока вартість, складність у реалізації	Високий рівень безпеки
Геометрія руки	Підвищена безпека в умовах забруднення	Потребує спеціалізованих сенсорів	Використання в середовищах з високими вимогами до безпеки

Існує обернена залежність між FAR та FRR, що створює компроміс між безпекою та зручністю. Підвищення порогу чутливості системи (score threshold) знижує FAR, що підвищує безпеку, але водночас збільшує FRR, що зменшує зручність. Оптимальний вибір між безпекою і зручністю залежить від конкретних потреб застосування [28].

Коефіцієнт рівної помилки (Equal Error Rate, EER) відображає точку, де FAR дорівнює FRR. Чим менше значення EER, тим точніше працює система загалом. Для повної оцінки системи важливе використання кривих DET та ROC, що відображають залежність помилок від параметрів системи, що дозволяє зробити більш точні висновки про її ефективність при різних налаштуваннях [26, 27].

Оцінка загроз та вразливостей біометричних систем є важливим аспектом, що визначає рівень їхньої безпеки. Біометричні дані мають унікальні властивості, які відрізняють їх від традиційних методів автентифікації, таких як паролі чи токени, і водночас створюють специфічні ризики. Одним із головних недоліків є відсутність секретності, оскільки багато біометричних ознак, таких як відбитки пальців, обличчя чи голос, можуть бути отримані без відома особи. Це підвищує ймовірність несанкціонованого доступу, оскільки дані можуть бути скомпрометовані без попереднього інформування власника [29].

Іншим важливим аспектом є незмінність та невідкличність біометричних шаблонів. У разі компрометації біометричних даних, наприклад, через викрадення з бази даних, ці шаблони неможливо змінити чи відкликати, як це відбувається з паролями. Така невідкличність робить витоки біометричних даних особливо небезпечними, адже наслідки компрометації можуть бути постійними і мати триваліший вплив на безпеку користувачів [30].

Додатково, використання однакових біометричних ознак у різних системах створює можливість перехресного зіставлення даних. У разі компрометації шаблону в одній системі, всі інші системи, де використовується та сама біометрична ознака, можуть опинитися під загрозою. Це підкреслює необхідність ретельного контролю доступу до біометричних даних і використання засобів захисту, що мінімізують ризик їхнього витоку чи несанкціонованого використання [29].

Стандартизація та законодавче регулювання є невід'ємними складовими забезпечення надійності, сумісності та відповідального використання біометричних технологій. Міжнародні органи, зокрема ISO/IEC, мають важливе значення для формулювання та підтримки стандартів, що стосуються біометрії. Так, Об'єднаний

технічний комітет 1 (JTC 1), через свої підкомітети, зокрема SC 37, займається розробкою базових біометричних стандартів, що включають формати обміну біометричними даними та технічні інтерфейси. Ці стандарти сприяють забезпеченню сумісності та безпеки при використанні біометричних систем у різних сферах. Зокрема, SC 17, що працює над стандартами для біометрії на ідентифікаційних картах, та SC 27, який розробляє стандарти безпеки, мають критичне значення для контролю за використанням біометричних технологій в умовах зростаючих загроз інформаційної безпеки [31].

На національному рівні важливу роль відіграє Національний інститут стандартів і технологій США (NIST), який активно співпрацює з міжнародними організаціями, такими як ISO/IEC, для розробки та вдосконалення біометричних стандартів. NIST також займається створенням інструментів для тестування відповідності стандартам і підтримує інформаційні ресурси, які сприяють впровадженню цих стандартів на практиці [32]. Роль NIST у розробці таких стандартів має вирішальне значення для забезпечення надійності та ефективності біометричних систем в різних застосуваннях, від національної безпеки до комерційних рішень.

Законодавчі норми, зокрема Загальний регламент про захист даних (GDPR), мають важливе значення для забезпечення безпеки біометричних даних, оскільки цей регламент класифікує біометричні дані як "спеціальну категорію" персональних даних, що потребує посиленого захисту. Встановлення чітких умов для обробки біометричних даних та визначення правових підстав для їх використання є важливим кроком у запобіганні зловживанням та порушенням прав особи. Окрім того, GDPR вимагає впровадження технічних і організаційних заходів безпеки, таких як шифрування та контроль доступу до біометричних даних, що сприяє збереженню конфіденційності та захисту інформації [33].

В Україні також спостерігається процес гармонізації національного законодавства зі стандартами ЄС, зокрема через прийняття проєкту Закону України "Про захист персональних даних". Однак, як зазначається у юридичному висновку

Ради Європи, існують деякі недоліки, зокрема щодо визначення біометричних даних та умов їх обробки, що потребує подальших уточнень для відповідності європейським стандартам. Такі питання, як точне визначення біометричних даних та забезпечення юридичної ясності, вимагають ретельної правової та технічної експертизи для забезпечення відповідності з міжнародними вимогами [34].

Біометричні технології набули широкого застосування в різних галузях завдяки їх здатності забезпечувати високий рівень безпеки та зручності. Одним з основних напрямків їх використання є контроль доступу, як фізичний, так і логічний, що охоплює не лише контроль доступу до будівель і приміщень, але й доступ до комп'ютерних систем, мереж та даних [35]. Також важливою сферою є облік робочого часу, де біометрія допомагає забезпечити точне відстеження часу приходу та відходу співробітників, запобігаючи шахрайству, такому як "buddy punching". Для цього часто застосовуються методи на основі відбитків пальців або геометрії руки [36].

У правоохоронній діяльності біометрія активно використовується для ідентифікації підозрюваних, порівняння доказів, таких як відбитки пальців і ДНК, а також для аналізу фото- та відеоматеріалів. У прикордонному контролі та імміграції біометричні технології сприяють підвищенню безпеки та швидкості перетину кордону, де часто використовуються методи розпізнавання обличчя, відбитків пальців і райдужки [37].

Банківська та фінансова сфера також активно використовує біометрію для автентифікації клієнтів при доступі до рахунків, авторизації транзакцій та запобігання шахрайству. Використовуються біометричні методи для забезпечення процедури "Знай свого клієнта" (KYC), що є критично важливим для фінансових установ [38]. В охороні здоров'я біометрія допомагає ідентифікувати пацієнтів, що дозволяє уникнути медичних помилок та захистити доступ до медичних записів і приміщень.

Біометричні технології також широко використовуються в споживчій електроніці, зокрема для розблокування смартфонів, планшетів і ноутбуків, а також

для автентифікації при мобільних платежах і доступі до додатків. У державному секторі біометрія застосовується для національних програм ідентифікації, таких як ID-карти та паспорти, реєстрації виборців і контролю доступу до урядових будівель та систем [39].

Різноманіття сфер застосування біометричних технологій демонструє їх перетворення з нішевого інструменту безпеки на універсальну технологію ідентифікації та автентифікації. Однак вибір конкретного біометричного методу часто залежить від специфічних вимог кожної ситуації, таких як рівень безпеки, умови експлуатації, наявна інфраструктура, вартість та прийнятність для користувачів. Наприклад, для військових об’єктів може використовуватися розпізнавання сітківки ока, а для побутових застосувань, таких як облік робочого часу, відбитки пальців або геометрія руки можуть бути більш зручними та економічними [40].

Розвиток біометричних технологій визначається кількома ключовими тенденціями, які суттєво впливають на їх майбутнє. Однією з таких тенденцій є впровадження мультимодальних систем, що комбінують кілька біометричних ознак, таких як обличчя та відбиток пальця, для підвищення точності і надійності ідентифікації. Це дозволяє компенсувати недоліки окремих методів і підвищити стійкість до спроб підробки [41]. Водночас, штучний інтелект і машинне навчання все більше використовуються для вдосконалення алгоритмів розпізнавання, що дозволяє покращити точність і швидкість ідентифікації навіть у складних умовах, таких як низька освітленість чи часткове закриття [42].

Безконтактне сканування також набуває популярності завдяки своїй гігієнічності та зручності, особливо в умовах пандемій. Одночасно з цим, мініатюризація та інтеграція біометричних сенсорів дає змогу вбудовувати ці технології в широкий спектр пристроїв, включаючи смартфони, носимі гаджети та елементи Інтернету речей (IoT) [43]. У зв’язку з посиленим захистом даних, важливу роль відіграють новітні методи шифрування, такі як гомоморфне шифрування та блокчейн, що дозволяють забезпечити високий рівень приватності користувачів [44].

У майбутньому конвергенція технологій, таких як ШІ, мультимодальність, безконтактне сканування та мініатюризація, може привести до безперервної біометричної автентифікації, вбудованої в повсякденні пристрої, що дозволить здійснювати фонове управління ідентичністю без участі користувача. Однак, це також створює нові вразливості, такі як можливість створення реалістичних підробок за допомогою ШІ, що вимагає ретельного тестування та захисту від атак [45].

1.3 Постановка задачі дослідження

У контексті безперервного зростання вимог до інформаційної та фізичної безпеки приміщень у різноманітних сферах, від приватного сектору до державних установ, розробка ефективних та надійних систем контролю доступу набуває особливої актуальності. Існуючі традиційні методи ідентифікації, що базуються на використанні фізичних ідентифікаторів, таких як картки доступу, або на інформаційних ідентифікаторах, таких як паролі, виявляють суттєві вразливості. Ці вразливості пов’язані з ризиками втрати, крадіжки або несанкціонованого копіювання карток, а також із ризиками забування, компрометації або підбору паролів [1, 2]. Такі недоліки традиційних систем створюють потенційні загрози для безпеки об’єктів та інформації, що зберігається. Це, в свою чергу, зумовлює нагальну потребу в пошуку та інтеграції більш досконалих та надійних механізмів автентифікації, здатних ефективно протистояти сучасним викликам. У цьому контексті, впровадження біометричних технологій ідентифікації та автентифікації користувачів є перспективним напрямком, оскільки вони базуються на унікальних фізіологічних або поведінкових характеристиках особистості, які важко підробити або передати [3, 4].

Сучасні загрози, пов’язані з кібербезпекою, такі як фішинг, соціальна інженерія та атаки на системи аутентифікації, роблять традиційні методи недостатньо надійними для захисту високочутливих даних і критичної

інфраструктури. Зокрема, компрометація паролів є однією з найбільш поширених форм атак, що підкреслює необхідність у більш стійких методах ідентифікації, таких як біометрія, яка базується на унікальних фізіологічних ознаках, які неможливо змінити чи забути [6, 7].

Вибір ідентифікації за відбитками пальців як основної біометричної технології для даної роботи зумовлений кількома ключовими факторами, які роблять її придатною для широкого застосування. По-перше, папілярний візерунок відбитка пальця є унікальним для кожної людини і залишається відносно стабільним протягом життя, що забезпечує високий ступінь надійності ідентифікації [7]. По-друге, технологія сканування відбитків пальців є відносно зрілою, добре дослідженою та економічно доступною [4, 5, 7], що дозволяє розробляти ефективні за показниками безпеки та водночас економічно доцільні системи контролю доступу. По-третє, використання відбитків пальців забезпечує високий рівень зручності для кінцевих користувачів, оскільки не вимагає запам’ятовування складних паролів або постійного носіння фізичних ідентифікаторів [4]. Зростаюча потреба у захисті інформації та фізичних приміщень, зумовлена збільшенням кількості кіберзлочинів та фізичних вторгнень, підкреслює важливість розробки ефективних та надійних систем контролю доступу, що здатні забезпечити належний рівень безпеки та захист цінностей.

Системи на основі відбитків пальців можуть використовувати додаткові технології, такі як нейронні мережі для більш точного аналізу ідентифікаційних характеристик або мультимодальність для підвищення надійності і точності. Комбінування кількох біометричних ознак, таких як відбитки пальців і розпізнавання обличчя, дозволяє створювати більш стійкі до фальсифікацій та підробок системи, що знижує ймовірність помилкових відмов і прийомів [10, 11].

У сучасному світі, що стрімко розвивається та ставить нові виклики у сфері безпеки, забезпечення ефективного захисту приміщень та інформації є першочерговим завданням. Постійний розвиток технологій безпеки вимагає від дослідників та інженерів безперервного пошуку інноваційних підходів та

вдосконалення існуючих рішень. Розробка систем, здатних адекватно реагувати на сучасні загрози та інтегрувати передові методи ідентифікації, є ключовим напрямком розвитку галузі. Впровадження новітніх технологій та їх адаптація для практичного застосування в системах контролю доступу вимагає глибокого наукового аналізу, ретельного проєктування та експериментальної перевірки розроблених рішень для підтвердження їхньої ефективності та надійності.

На основі проведеного аналізу існуючих систем контролю доступу та біометричних технологій стає очевидним, що, попри розвиток передових рішень, існує актуальна потреба у розробці доступних, гнучких та водночас надійних систем, здатних ефективно протистояти сучасним загрозам, пов’язаним із компрометацією традиційних ідентифікаторів. Незважаючи на високий потенціал біометричних технологій, питання їхньої вартості, складності впровадження та відповідності ресурсним обмеженням певних апаратних платформ залишаються актуальними викликами, що вимагають подальших досліджень та інженерних рішень.

Вибір мікроконтролерної платформи Arduino ATMega328 для реалізації такої системи є значущим у контексті створення доступних та дослідницьких рішень. Це дозволяє дослідити можливості та виклики впровадження ресурсомістких завдань, як обробка та зберігання біометричних даних, на апаратних засобах з обмеженими обчислювальними можливостями та пам’яттю, які є широко доступними та економічно ефективними. Такий підхід відкриває шляхи для створення масштабованих та адаптованих рішень, що можуть бути використані в освітніх цілях, прототипуванні або у складі бюджетних систем безпеки, демонструючи потенціал вбудованих систем у сфері безпеки.

Розробка сучасних інженерних систем, зокрема у такій критично важливій сфері, як безпека, часто вимагає інтеграції знань та підходів із різних технологічних галузей. Успішна реалізація проєктів на перетині апаратного забезпечення, програмного забезпечення, спеціалізованих технологій ідентифікації, як біометрія, та аспектів кібербезпеки є свідченням комплексного підходу до вирішення

поставлених задач та відображає багатопрофільний характер сучасних досліджень у галузі захисту.

Процес створення та дослідження подібних систем контролю доступу включає послідовні етапи, починаючи від глибокого аналізу предметної області та існуючих рішень. Далі слідують етапи проєктування, практичної реалізації апаратної та програмної частин системи, а також ретельної перевірки функціональності, ефективності та надійності розробленого рішення шляхом експериментальних випробувань. Такий підхід дозволяє не лише створити робочу систему, але й об'єктивно оцінити її характеристики, виявити потенційні обмеження та визначити напрями для подальшого вдосконалення, що є важливим для внеску у розвиток галузі.

Реалізація наукових досліджень у галузі інженерії безпеки є ключовим механізмом для трансформації теоретичних знань та лабораторних розробок у функціональні системи, здатні відповідати реальним потребам користувачів та протистояти актуальним загрозам. Зокрема, дослідження можливостей використання доступних апаратних платформ для вирішення складних завдань, таких як біометрична ідентифікація, сприяє розширенню сфер застосування сучасних технологій безпеки та робить їх більш доступними для широкого кола споживачів та організацій. Отримані в ході подібних робіт результати мають цінність як для подальших наукових досліджень, так і для практичного впровадження у складі різноманітних систем захисту та автоматизації.

В умовах всебічної інтеграції технологій у повсякденне життя та виробничі процеси, питання надійності та безпеки використовуваних систем набувають особливої гостроти. Розвиток мікроконтролерних систем та технологій вбудованих обчислень відкриває нові можливості для створення інтелектуальних рішень безпосередньо на місцях їх застосування, зокрема у сфері фізичної безпеки та контролю доступу. Такі рішення вимагають глибокого розуміння взаємодії апаратних та програмних компонентів, а також врахування потенційних безпекових загроз.

Основною метою даної роботи є розробка та експериментальне дослідження функціональних характеристик системи автоматичного контролю доступу до приміщень на базі мікроконтролера Arduino ATMega328, що використовує біометричні дані відбитків пальців для ідентифікації користувачів. Досягнення цієї мети передбачає створення функціонального та надійного прототипу системи, здатного реалізувати процеси реєстрації та ідентифікації користувачів із застосуванням обраної біометричної технології, забезпечуючи при цьому належний рівень безпеки та контроль доступу. Розроблена система матиме потенціал для широкого застосування в різноманітних сферах, включаючи житлові будинки, офісні приміщення, лабораторії та державні установи, де необхідний високий рівень безпеки та контроль доступу з використанням доступних апаратних рішень.

У рамках цього дослідження ставиться комплекс взаємопов'язаних завдань, спрямованих на створення ефективної системи контролю доступу з біометричною ідентифікацією на мікроконтролерній платформі. Першим завданням є проведення аналізу сучасних систем контролю доступу, зокрема біометричних технологій, з метою виявлення їхніх сильних і слабких сторін та визначення місця запропонованого підходу серед існуючих рішень [1, 2, 3, 4, 5]. Другим завданням є розробка або адаптація ефективного алгоритму ідентифікації користувачів, що включає етапи отримання та обробки зображення відбитка пальця, виділення характерних ознак (мінуцій) та їх подальшого порівняння із збереженими шаблонами. Виконання цих операцій на мікроконтролері Arduino ATMega328 вимагає розробки рішень, що оптимально відповідають технічним характеристикам та обмеженим обчислювальним ресурсам даної платформи [9]. Враховуючи обмежені обчислювальні ресурси мікроконтролера Arduino ATMega328, актуальним є дослідження та оптимізація алгоритмів обробки біометричних даних відбитків пальців для забезпечення прийнятної швидкодії та точності ідентифікації в умовах обмеженої пам'яті (SRAM та Flash) та обчислювальної потужності. Необхідно дослідити та обрати ефективний метод вилучення характерних ознак та їх компактного представлення для подальшого порівняння, а також дослідити

можливості застосування спрощених метрик відстані для порівняння біометричних шаблонів з метою зниження обчислювального навантаження. Третім завданням є розробка стратегії зберігання біометричних шаблонів користувачів в енергонезалежній пам'яті. З урахуванням обмеженого обсягу внутрішньої Flash-пам'яті мікроконтролера, яка також використовується для зберігання програмного коду [9], актуальним є дослідження та вибір методів використання зовнішніх запам'ятовуючих пристроїв, таких як карти пам'яті SD або мікросхеми EEPROM, для зберігання бази даних, забезпечуючи при цьому необхідний рівень конфіденційності даних користувачів. Четвертим завданням є проєктування та створення апаратно-програмного комплексу системи, що включає мікроконтролер Arduino ATMega328, біометричний сенсор відбитків пальців та виконавчі пристрої, такі як електрозамок та засоби індикації (наприклад, LCD дисплей), з урахуванням їхньої інтеграції та підключення через відповідні інтерфейси мікроконтролера (UART, цифрові/аналогові виходи). П'ятим завданням є розробка відповідного програмного забезпечення, яке забезпечує коректну взаємодію між усіма компонентами, реалізує алгоритми обробки біометричних даних та логіку управління системою контролю доступу. Шостим завданням є проведення експериментальних випробувань розробленої системи для об'єктивної оцінки її функціональних характеристик, таких як швидкість та точність ідентифікації (FAR, FRR), загальна надійність та стабільність роботи в різних умовах. Сьомим завданням є дослідження та аналіз енергоспоживання розробленої системи з метою оптимізації та визначення можливостей автономного застосування. Восьмим завданням є аналіз та мінімізація ключових ризиків функціонування системи, пов'язаних із безпекою біометричних даних (підробка, компрометація, приватність) та загальними загрозами систем безпеки (відмова обладнання, втрата даних, атаки на ПЗ), з дослідженням відповідних заходів протидії (шифрування, виявлення підробки, резервування, кіберзахист). Дев'ятим завданням є проведення порівняльного аналізу розробленої системи з існуючими аналогами для оцінки її переваг та недоліків. Десятим завданням є формулювання висновків щодо практичної застосовності

системи та розробка рекомендацій щодо її впровадження та подальшого вдосконалення на основі отриманих результатів дослідження.

Дослідження проводилося з використанням комплексу теоретичних та експериментальних методів. Теоретичні дослідження включали аналіз наукової та технічної літератури, порівняльний аналіз існуючих рішень та біометричних технологій, а також розробку та дослідження алгоритмів обробки, ідентифікації та безпечного зберігання біометричних даних, враховуючи специфіку обраної апаратної платформи. Експериментальні дослідження передбачали проєктування та створення апаратно-програмного комплексу системи, розробку відповідного програмного забезпечення та проведення випробувань для оцінки функціональних характеристик розробленої системи (швидкості, точності, надійності) та її енергоспоживання в умовах, наближених до реальної експлуатації.

1.4 Висновок

У першому розділі дипломної роботи проведено грунтовне дослідження теоретичних та практичних аспектів побудови систем автоматичного контролю доступу із використанням біометричних даних. Особливу увагу приділено сучасному стану розвитку систем ідентифікації та верифікації особи, зокрема на основі біометричних ознак, таких як відбитки пальців. Проведено детальний аналіз технологій, які лежать в основі біометричних систем безпеки, а також виявлено ключові технічні та експлуатаційні характеристики, що впливають на ефективність, надійність і захищеність таких систем.

У ході дослідження були вивчені апаратні і програмні засоби, що застосовуються в сучасних біометричних системах, включаючи сенсори, алгоритми обробки даних, методи зберігання та захисту персональної інформації, а також типові архітектурні підходи до побудови систем контролю доступу. Визначено основні вимоги до точності, швидкодії, стабільності та рівня безпеки, які повинна забезпечувати система, орієнтована на використання в умовах обмежених ресурсів.

Окрему увагу приділено порівнянню біометричних методів з традиційними способами ідентифікації, зокрема з картковими та PIN-код системами, що дозволило підкреслити переваги біометрії у контексті підвищення рівня захисту та зменшення ризику несанкціонованого доступу.

З позиції інженерної реалізації було розглянуто доцільність застосування платформи Arduino на базі мікроконтролера ATMega328 для створення доступної, компактної та енергоефективної системи контролю доступу. Було проведено аналіз її технічних можливостей, сумісності з біометричними модулями, простоти інтеграції та програмування, що дозволило обгрунтувати вибір саме цієї апаратної бази для побудови прототипу системи. Також враховано фактори вартості, доступності компонентів, гнучкості в налаштуванні та можливості масштабування проєкту в майбутньому.

У межах теоретичної постановки задачі було чітко сформульовано мету розробки, а саме створення автоматизованої системи контролю доступу з використанням біометричних технологій на основі доступних апаратних засобів.

Сформульовано ключові завдання, які охоплюють проєктування архітектури системи, вибір оптимальних компонентів, розробку алгоритмів і програмного забезпечення, проведення тестування та оцінювання ефективності реалізованої моделі. Також визначено основні ризики, пов’язані з використанням біометричних даних, зокрема ризики компрометації особистої інформації, а також технічні обмеження, які можуть впливати на точність і надійність системи. У роботі наголошено на потенційних перевагах запропонованого підходу, таких як підвищена безпека, зручність у користуванні, можливість масштабування і впровадження в різних сферах, включаючи офіси, навчальні заклади, виробничі об’єкти та приватні приміщення.

Окрім цього, було окреслено комерційний потенціал системи в умовах сучасного зростаючого попиту на доступні й ефективні рішення в сфері інформаційної та фізичної безпеки. Проведено оцінку можливості інтеграції розробки в існуючі інфраструктури та зазначено перспективи для подальших

удосконалень і розширення функціоналу. Також визначено критерії, за якими буде проводитися порівняльний аналіз системи, зокрема показники точності, швидкодії, вартості та зручності використання.

Таким чином, у межах першого етапу дослідження було закладено міцну теоретичну та методологічну основу для розробки системи автоматичного контролю доступу, що поєднує переваги біометричних технологій із простотою і доступністю апаратних засобів на базі мікроконтролера Arduino ATMega328. Отримані результати є необхідною передумовою для практичної реалізації та подальшого вдосконалення даної системи.

2 ВИБІР ТА ОБГРУНТУВАННЯ КОМПОНЕНТІВ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ

2.1 Аналіз та вибір компонентів мікроконтролерної системи

У процесі створення мікроконтролерної системи контролю доступу надзвичайно важливим етапом є обгрунтований вибір апаратної платформи, яка забезпечуватиме належний рівень функціональності, продуктивності, масштабованості та енергоефективності. Серед доступних варіантів було проаналізовано три поширені рішення: Arduino UNO R3, STM32 та ESP32 кожне з яких має свої переваги та обмеження, що визначають доцільність їх застосування в тому чи іншому проєкті.

Arduino UNO R3 є однією з найвідоміших і найпопулярніших платформ серед початківців, розробників прототипів та освітніх установ. Її головна перевага полягає у простоті використання, яка реалізується завдяки інтуїтивно зрозумілому середовищу розробки Arduino IDE, великій кількості готових бібліотек, прикладів та широкій спільноті користувачів [45, 47]. Процесор ATmega328P із тактовою частотою 16 МГц має обмежену обчислювальну потужність, однак цього достатньо для реалізації базових функцій системи доступу на початковому етапі. Завдяки використанню логіки 5 В, UNO R3 також сумісна з багатьма простими периферійними модулями, що спрощує підключення компонентів без додаткових узгоджень напруги.

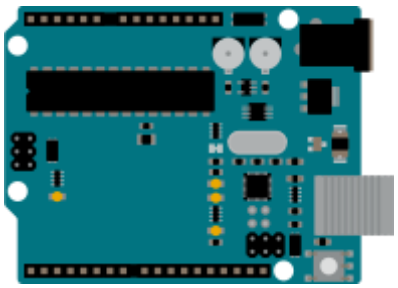


Рисунок 2.1 – Мікроконтролер Arduino Uno R3 [45]

Платформа STM32, яка базується на 32-бітових ARM Cortex-M ядрах, забезпечує вищий рівень продуктивності, розширені можливості роботи з периферією та точне управління в реальному часі. Залежно від серії, мікроконтролери STM32 можуть мати різну тактову частоту, обсяг пам’яті та спеціалізовані функціональні блоки, зокрема високоточні таймери, АЦП/ЦАП, шини зв’язку тощо [48]. Незважаючи на ці технічні переваги, вхідний поріг для роботи зі STM32 є вищим через складнішу екосистему, необхідність використання середовищ на кшталт STM32CubeIDE та знання бібліотек низького рівня таких як HAL, LL, може ускладнити швидкий старт проєкту.

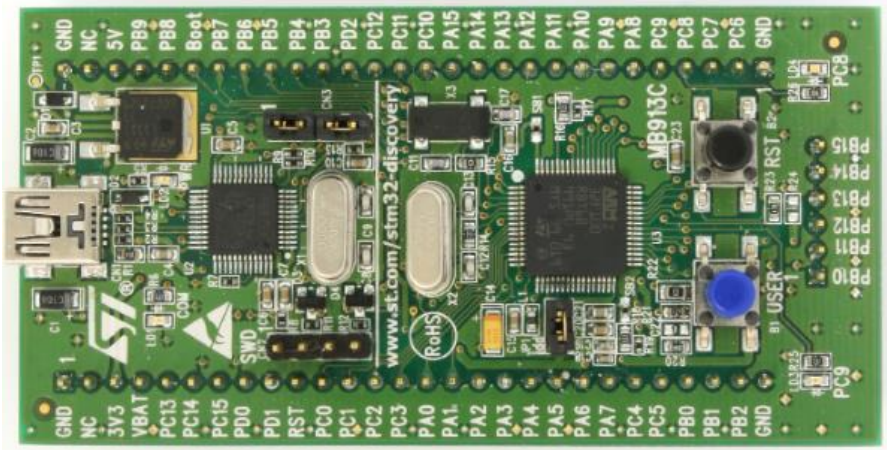


Рисунок 2.2 – Мікроконтролер STM32 [46]

Третій розглянутий варіант ESP32 поєднує високу обчислювальну потужність із підтримкою бездротових інтерфейсів Wi-Fi та Bluetooth (включно з BLE), що робить його ідеальним вибором для проєктів Інтернету речей (IoT) [46]. Двоядерний процесор із частотою до 240 МГц, велика кількість портів вводу/виводу, вбудовані криптографічні модулі та енергоефективні режими сну забезпечують широкі можливості для розробки складних систем. При цьому ESP32 підтримує програмування як через Arduino IDE, так і через більш гнучке середовище ESP-IDF, що дозволяє поступово переходити до розширених функцій у міру зростання вимог до проєкту [48]. Важливим нюансом є використання логіки 3.3 В, що вимагає узгодження рівнів при роботі з 5-вольтовими пристроями.

Рисунок 2.3 – Мікроконтролер ESP32 [46]

Порівняння зазначених платформ дає змогу оцінити їх з точки зору балансу між простотою розробки, апаратною потужністю та функціональністю. Arduino UNO R3, завдяки своїй простоті, залишається зручним вибором для початку розробки та швидкого створення прототипу. STM32 – це варіант для складніших рішень, де важливе управління в реальному часі та широкий доступ до периферії. ESP32 натомість надає потужні обчислювальні ресурси та вбудовані мережеві функції, що дозволяє ефективно реалізовувати сучасні IoT-системи. Таким чином, кожна з платформ має свої переваги, і їхній вибір визначається конкретними вимогами до проєкту та рівнем досвіду розробника [45-48].

У процесі створення системи контролю доступу важливо не лише обрати відповідну апаратну платформу, а й підібрати оптимальний дисплей для відображення інформації користувачеві. З-поміж численних варіантів, особливу увагу було приділено трьом основним типам дисплеїв: символьному LCD 1602, графічному LCD та OLED-дисплеям. Кожен з них має унікальні характеристики, що впливають на вибір залежно від вимог до функціональності, енергоспоживання, вартості та складності реалізації.

Символьний LCD 1602 є класичним рішенням для простих вбудованих систем. Його конструкція орієнтована на відображення фіксованого набору символів у заданій сітці (зазвичай 2 рядки по 16 символів). Завдяки вбудованому

контролеру HD44780 цей дисплей надзвичайно простий у використанні, має стандартизований інтерфейс та велику кількість бібліотек для платформ Arduino та подібних мікроконтролерів [49, 50]. Він не вимагає великої обчислювальної потужності або пам’яті для кадрів, що робить його ідеальним для систем з обмеженими ресурсами. Зазвичай дисплей підтримує підсвітку, яка може бути вимкнена або керована, зменшуючи енергоспоживання. Це робить символьний LCD одним із найбільш енергоефективних варіантів серед усіх доступних дисплеїв.

Рисунок 2.4 – LCD 1602 символьний дісплей 16x2 [50]

Графічні LCD-дисплеї, на відміну від символьних, дозволяють керувати окремими пікселями, що відкриває можливість відображення довільного тексту, шрифтів, графіків, іконок та навіть простих анімацій. Такі дисплеї використовуються у випадках, коли необхідна гнучкість у візуалізації інформації. Однак за цю гнучкість доводиться платити, тобто графічні LCD потребують значно більше пам’яті, оскільки весь вміст дисплея зберігається у буфері кадру, а також більшої обчислювальної потужності для обробки піксельних даних. Крім того, енергоспоживання цих дисплеїв вищі, оскільки необхідно підсвічувати всю площу екрана незалежно від вмісту, що є особливо суттєвим у портативних пристроях з автономним живленням [52].

Рисунок 2.5 – LCD 12864 графічний дисплей 128x64 [51]

OLED-дисплеї поєднують гнучкість графічного відображення з вражаючими оптичними характеристиками: глибокий справжній чорний колір (завдяки відсутності підсвітки), висока контрастність, яскравість, широкі кути огляду та швидкий відгук. Вони також підтримують керування окремими пікселями, однак, на відміну від LCD, кожен піксель сам випромінює світло. Це дозволяє значно зменшити споживання енергії для переважно темного інтерфейсу чорні пікселі фактично не споживають живлення [51]. З іншого боку, яскраві зображення з великою кількістю білих елементів можуть суттєво підвищити енергоспоживання. Крім того, OLED-дисплеї є дорожчими, ніж символьні або навіть графічні LCD, а також мають обмеження за терміном служби через явище “вигорання” пікселів при тривалому відображенні статичних елементів [51].

Рисунок 2.6 – 0.96-дюймовий OLED-дисплей [52]

З огляду на ці аспекти, вибір символьного LCD 1602 для проєкту системи контролю доступу є виваженим і доцільним рішенням на етапі розробки прототипу. Головним завданням дисплея є передача коротких текстових повідомлень користувачу, таких як “Доступ дозволено”, “Відмовлено” або “Прикладіть палець”. Для цього немає потреби у складній графіці чи високій контрастності, простого текстового інтерфейсу цілком достатньо. Використання LCD 1602 мінімізує апаратну та програмну складність, знижує вартість проєкту і зменшує споживання енергії, що є особливо важливим у випадку автономного живлення пристрою.

У майбутньому, коли система досягне стабільності та функціональної завершеності, можливе оновлення до OLED-дисплея або графічного LCD, якщо виникне потреба у покращенні візуального інтерфейсу або додаванні графічних елементів. Однак на поточному етапі символьний LCD 1602 є практичним рішенням, яке дозволяє сконцентруватися на реалізації основної логіки системи доступу без перевантаження розробки непотрібною складністю [49-52].

У сучасних біометричних системах контролю доступу технологія зчитування відбитків пальців відіграє ключову роль у забезпеченні автентифікації користувачів. На ринку представлені різні типи сенсорів, зокрема оптичні, ємнісні та ультразвукові, кожен з яких має свої переваги, недоліки та сценарії застосування. У межах цієї роботи використовується оптичний сенсор FPM10A, тому доцільно проаналізувати його в контексті порівняння з альтернативними технологіями, враховуючи сучасні тенденції.

Оптичні сенсори, такі як FPM10A, працюють за принципом зчитування 2D-зображення відбитка за допомогою камери та світлодіодного підсвічування. Світло відбивається від шкіри пальця, і камера фіксує візерунок ліній. Такий підхід є простим, добре вивченим і масово реалізованим у недорогих пристроях. Сенсор FPM10A має вбудований алгоритм порівняння, пам’ять для шаблонів та підтримку інтерфейсів UART, що робить його зручним для використання з мікроконтролерами типу Arduino [53, 54]. Головною перевагою оптичних сенсорів

є доступна ціна, що дозволяє використовувати їх у бюджетних прототипах і навчальних проєктах.

Рисунок 2.7 – Сенсор FPM10A [54]

Ємнісні сенсори зчитують відбиток пальця, вимірюючи зміни електричної ємності між сенсором і виступами чи западинами шкіри. Оскільки ці сенсори реагують не на світло, а на фізичні властивості шкіри, вони менш чутливі до умов зовнішнього середовища: вологи, бруду чи яскравого освітлення. Ємнісні сенсори зазвичай забезпечують вищу надійність і точність, ніж оптичні, і їх складніше обдурити, тобто для створення фальшивого відбитка потрібен матеріал, що має подібну електропровідність до живої шкіри. Крім того, сенсори такого типу зазвичай мають компактні розміри, що дозволяє інтегрувати їх у тонкі мобільні пристрої та платформи із жорсткими обмеженнями по простору.

Рисунок 2.8 – Ємнісні датчики SICK [55]

Ультразвукові сенсори – це найсучасніша технологія, яка використовує високочастотні звукові хвилі для формування 3D-зображення відбитка, включаючи підповерхневі шари шкіри. Такий підхід дозволяє досягти найвищого рівня точності, особливо в складних умовах, коли палець мокрий, забруднений або пошкоджений. Ультразвукові сенсори також найкраще протидіють спуфінгу, оскільки розпізнають не лише поверхневі контури, а й структури, недоступні для підробок. Проте ця технологія має найвищу вартість, а також може працювати повільніше через обробку великого обсягу даних. Незважаючи на це, вона активно впроваджується у преміальні смартфони та високобезпечні системи доступу.

Рисунок 2.9 – Ультразвуковий датчик відстані SICK UC30-21516A [55]

Таким чином, вибір сенсора FPM10A для реалізації прототипу системи контролю доступу є виваженим компромісом між вартістю, функціональністю та простотою інтеграції. Незважаючи на нижчий рівень захисту порівняно з ємнісними та ультразвуковими сенсорами, його використання дозволяє швидко протестувати базову логіку системи, не перевантажуючи проєкт апаратними чи програмними складнощами. Це типове рішення для етапу розробки MVP (Minimum Viable Product) або для систем, де необхідний базовий рівень автентифікації без підвищених вимог до безпеки. У випадках, коли система повинна протистояти підробкам або працювати в складних умовах, перехід до ємнісної або ультразвукової технології є логічним наступним кроком [53; 54].

Важливим аспектом при обгрунтуванні вибору апаратних компонентів для прототипу системи контролю доступу є також економічна доцільність, особливо на етапі первинної розробки та тестування концепції. Проведений аналіз вартості ключових елементів мікроконтролерної системи на українському ринку станом на поточний період (травень 2025 року) показує наступну структуру витрат. Базова платформа розробки, мікроконтролер Arduino Uno R3 на базі ATMega328P, характеризується вартістю у 252 гривні. Засіб візуального сповіщення користувача, а саме символьний дисплей LCD 1602 (16x2) з інтегрованим I2C модулем, що спрощує його підключення, оцінюється у 168 гривень. Біометричний модуль, оптичний сенсор відбитків пальців FPM10A, який є ключовим для реалізації функції ідентифікації, має вартість 504 гривні. Таким чином, сукупна вартість основних апаратних засобів, необхідних для побудови функціонального прототипу системи, становить 924 гривні. Цей рівень витрат підтверджує економічну обгрунтованість обраної компонентної бази для задач прототипування та розробки мінімально життєздатного продукту (MVP), дозволяючи зосередитися на програмній реалізації та відпрацюванні алгоритмів без значних початкових інвестицій в апаратне забезпечення.

2.2 Вибір та обгрунтування бази даних для зберігання біометричних даних

Біометричні дані, що охоплюють унікальні фізичні та поведінкові характеристики особи такі як відбитки пальців, риси обличчя, структура райдужної оболонки ока, голос або ДНК набули широкого поширення у сфері автентифікації та ідентифікації. Їхня унікальність і сталість роблять такі дані особливо цінними, але водночас і вразливими. На відміну від традиційних методів захисту, як-от паролі або PIN-коди, компрометація біометричної інформації має незворотний характер, оскільки ці характеристики не підлягають заміні чи скиданню. У разі витоку біометричних шаблонів користувач втрачає контроль над цими даними

назавжди, що створює потенційно серйозні ризики безпеки, зокрема повторне використання викрадених шаблонів у зловмисних цілях [59].

У зв’язку з цим критично важливим є ретельний вибір системи керування базами даних (СКБД), яка буде використовуватися для зберігання біометричних даних. Така СКБД має забезпечувати не лише ефективне збереження і швидкий доступ до біометричних записів, але й відповідати високим вимогам до захисту інформації. До ключових вимог належать підтримка шифрування даних, керування правами доступу, аудит змін та стійкість до несанкціонованих втручань [59].

Біометричні дані становлять собою вимірювані фізіологічні або поведінкові характеристики людини, які дозволяють ідентифікувати або автентифікувати особу. Серед найпоширеніших фізіологічних характеристик, що використовуються у біометричних системах, виокремлюють відбитки пальців, риси обличчя, структуру райдужної оболонки ока, геометрію руки та ДНК. Водночас поведінкові характеристики включають голос, динаміку підпису, натискання клавіш та ходу. Ці ознаки забезпечують високий рівень унікальності, що робить біометрію ефективним засобом безконтактної перевірки особистості в різних прикладних сферах, зокрема у смартфонах, системах контролю доступу та банківських застосунках [60; 61].

На практиці біометричні дані рідко зберігаються у вигляді первинних зображень або сигналів. Натомість після зчитування біометричного зразка (наприклад, сканування відбитка пальця або обличчя) здійснюється обробка із виділенням унікальних ознак, які перетворюються у математичний шаблон, який називається biometric template. Цей шаблон є цифровим представленням ознак, що не дозволяє відновити початкове зображення, а отже, забезпечує вищий рівень безпеки порівняно зі зберіганням сирих даних [62].

Існують різні підходи до зберігання біометричних шаблонів залежно від архітектури системи. До апаратних рішень належать спеціалізовані пристрої з локальним зберіганням, зокрема смарт-карти або USB-токени, які мінімізують ризики, пов’язані з передаванням даних мережею. У мобільних пристроях, таких

як смартфони, біометричні шаблони зазвичай зберігаються на захищених чіпах без можливості їх вилучення. Централізоване зберігання на біометричних серверах дозволяє автентифікацію з різних точок, однак створює додаткові виклики у сфері захисту даних. Альтернативним є гібридне, або розподілене, зберігання, коли частина шаблону розміщується на сервері, а інша на пристрої користувача, що знижує вірогідність повного доступу до даних у разі компрометації одного з вузлів [62].

Біометричні дані можуть використовуватись у двох основних операціях: автентифікації та ідентифікації. У разі автентифікації (1:1) здійснюється перевірка твердження про особу, що передбачає порівняння з єдиним шаблоном, асоційованим з користувачем, і застосовується, наприклад, при вході в систему з використанням облікового запису або персонального пристрою. Ідентифікація (1:N), навпаки, має на меті встановлення невідомої особи шляхом порівняння її біометричного зразка з усіма шаблонами у базі даних. Цей процес є складнішим і потребує значних обчислювальних ресурсів, особливо у масштабних системах, де якість зображень та ефективність алгоритмів відіграють вирішальну роль [63].

Вибір системи керування базами даних (СКБД) для зберігання біометричної інформації визначається специфічними вимогами до безпеки, масштабованості, продуктивності та цілісності даних, що зумовлено чутливим характером біометричних шаблонів та високими ризиками при їх компрометації. Серед першочергових вимог є захист від несанкціонованого доступу, що передбачає використання надійних моделей керування правами, зокрема рольової моделі доступу (RBAC), яка гарантує, що лише авторизовані користувачі мають можливість взаємодіяти з інформацією відповідно до своєї ролі у системі [63].

Забезпечення конфіденційності даних є критично важливим. Для цього застосовуються алгоритми шифрування даних як у стані спокою (at rest), так і під час передавання (in transit). Деякі сучасні СКБД, такі як MongoDB, підтримують запити до зашифрованих даних без необхідності їх попередньої розшифрації, що реалізується через технологію Queryable Encryption. Іншим інноваційним підходом

є гомоморфне шифрування, яке дає змогу обробляти зашифровані дані без доступу до їх відкритого вигляду [64]. Крім того, система має бути захищеною від атак типу SQL-ін’єкцій, DoS (відмова в обслуговуванні) та спроб підробки біометричних ознак (spoofing), зокрема з використанням deepfake-технологій. У зв’язку з цим набувають поширення методи перевірки “живості” (liveness detection), що дозволяють визначити, чи належить зразок реальній людині, а не штучному джерелу [65].

Масштабованість є ще одним визначальним чинником. У випадку зростання кількості користувачів або запитів система повинна зберігати стабільну продуктивність. Існують два основні типи масштабування: вертикальне (scale-up), що полягає в розширенні ресурсів одного вузла (процесор, оперативна пам’ять, сховище), та горизонтальне (scale-out), при якому дані розподіляються між кількома вузлами системи шляхом кластеризації або шардингу. Шардинг, зокрема, передбачає розбиття бази на незалежні частини, що обробляються паралельно, однак вимагає складної архітектури [66].

Питання продуктивності має особливе значення для біометричних систем, у яких ідентифікація 1:N передбачає порівняння з великою кількістю записів у базі даних. У таких випадках ключовими показниками є низька затримка (latency) під час обробки запитів та висока пропускна здатність системи, що дозволяє обробляти десятки запитів щосекунди. Для систем безперервної автентифікації, зокрема в режимі реального часу, ці характеристики є критичними [67].

Збереження цілісності біометричних даних включає забезпечення їх точності, узгодженості та повноти протягом усього життєвого циклу. У випадку реляційних СКБД важливою є підтримка транзакцій з дотриманням властивостей ACID (атомарність, узгодженість, ізольованість, довговічність), що гарантує надійність обробки інформації [68]. Особливої уваги потребує запобігання помилкам при шифруванні та дешифруванні, оскільки біометричні шаблони є унікальними і непідлягають відновленню у разі втрати або пошкодження.

Одним із перспективних рішень для зберігання та обробки біометричних даних є використання об’єктно-реляційної системи управління базами даних PostgreSQL, яка поєднує високу надійність, розширюваність і відповідність стандартам SQL. Серед головних переваг PostgreSQL варто виокремити її відповідність принципам ACID, що забезпечує цілісність і надійність транзакцій, що є критично важливим для роботи з чутливими біометричними шаблонами [69].

Система пропонує широкий спектр механізмів безпеки, зокрема реалізацію рольової моделі доступу (RBAC), контроль доступу на рівні рядків (RLS), підтримку SSL/TLS-з’єднань, а також можливість шифрування окремих стовпців за допомогою розширення pgcrypto. У певних випадках також використовується Transparent Data Encryption (TDE) для забезпечення повного шифрування даних на рівні файлової системи [70].

Особливої уваги заслуговує розширення pgvector, яке дозволяє зберігати векторні представлення біометричних шаблонів (так звані embeddings) та ефективно виконувати пошук за метриками подібності, такими як евклідова відстань, косинусна подібність чи внутрішній добуток. pgvector також підтримує використання індексів для оптимізації векторного пошуку, зокрема HNSW (Hierarchical Navigable Small World Graphs) та IVFFlat (Inverted File with Flat Quantization), що дає змогу зменшити час відповіді при пошуку подібних ознак [71].

Додатковими перевагами PostgreSQL є підтримка різноманітних типів даних, включаючи BYTEA для зберігання бінарних шаблонів, та JSONB для метаданих, а також широкі можливості індексації (B-tree, Hash, GiST, GIN), які сприяють підвищенню продуктивності запитів. Щодо масштабованості, PostgreSQL підтримує реплікацію, партиціонування та розширення для горизонтального масштабування, зокрема Citus, що дозволяє адаптувати систему до зростання обсягів біометричних записів. Важливою перевагою є й відсутність ліцензійних платежів, що робить PostgreSQL привабливим з точки зору загальної вартості володіння (TCO) у порівнянні з комерційними рішеннями [70].

Втім, використання PostgreSQL у контексті великих біометричних систем пов'язане з низкою викликів. Зокрема, pgvector, хоча й забезпечує високу ефективність на середніх наборах даних, може втрачати продуктивність або вимагати значних ресурсів при роботі з мільярдами векторів. Досвід компанії Heroku свідчить про складнощі використання pgvector у середовищах з великою кількістю користувачів, а також про необхідність ретельного тестування продуктивності на етапі проєктування системи [71]. Також варто враховувати, що налаштування горизонтального масштабування у PostgreSQL, хоча й можливе, зазвичай складніше, ніж у деяких NoSQL-рішеннях, спеціально створених для розподіленої архітектури. Додатковим обмеженням може стати тривалість процесу відновлення індексів pgvector після аварійного відновлення бази, що може впливати на доступність системи при роботі з великими обсягами даних [71].

MongoDB є однією з найпоширеніших NoSQL баз даних, орієнтованих на документоцентричний підхід, і використовує для зберігання даних формат BSON, який є бінарним аналогом JSON, який поєднує гнучкість структури з ефективністю зберігання та обробки [73]. Такий підхід забезпечує MongoDB особливою адаптивністю щодо роботи з різними типами біометричних даних. Відсутність жорстко визначеної схеми дозволяє зберігати як традиційні шаблони, так і супровідні метадані, не потребуючи попереднього визначення структури, що є перевагою при роботі з новими біометричними модальностями або умовами динамічної зміни даних [74].

MongoDB є системою, яка з самого початку проєктувалася з акцентом на горизонтальне масштабування. Її архітектура базується на механізмі шардингу, тобто розподілу даних між кількома вузлами, що дозволяє ефективно обробляти великі обсяги біометричної інформації та високі навантаження при одночасному збереженні продуктивності [74]. У межах хмарної платформи MongoDB Atlas реалізовано компонент Atlas Vector Search, що дозволяє здійснювати пошук за векторними представленнями біометричних шаблонів. Цей інструмент підтримує побудову індексів на основі алгоритмів HNSW (Hierarchical Navigable Small World)

та IVF (Inverted File), що надає можливість реалізувати високошвидкісний семантичний пошук у великих векторних просторах [75].

Водночас MongoDB має низку обмежень, які варто враховувати при використанні її у біометричних системах. Попри те що починаючи з версії 4.0 MongoDB підтримує ACID-транзакції для операцій з кількома документами, загальна модель узгодженості та транзакційної обробки все ще поступається традиційним реляційним СКБД за жорсткістю та передбачуваністю поведінки [74]. Також ускладнення викликає реалізація складних аналітичних запитів, зокрема з'єднання даних між різними колекціями (аналогами таблиць у реляційних СКБД), де MongoDB може демонструвати нижчу ефективність у порівнянні з SQL-орієнтованими рішеннями [74].

Ще одним викликом є перебудова індексів у рамках Atlas Search. Масштабування або оновлення інфраструктури в MongoDB Atlas може вимагати повної реконструкції індексів, що є тривалим процесом і може спричиняти навантаження на кластери. Окрім того, хоча MongoDB Community доступна безкоштовно, функціональні можливості платформи Atlas, включаючи Vector Search та Queryable Encryption, доступні лише в рамках платного хмарного середовища, що може підвищити загальну вартість володіння системою [76].

Oracle Database є однією з провідних комерційних реляційних систем керування базами даних, яка вирізняється високим рівнем надійності, масштабованості та широким спектром інструментів для забезпечення інформаційної безпеки. Завдяки багаторічному розвитку ця платформа посідає вагоме місце в інфраструктурах великих підприємств та організацій, зокрема в тих, що оперують чутливою інформацією, включаючи біометричні дані [77].

У контексті зберігання біометричних шаблонів Oracle Database пропонує розширені функціональні можливості захисту, серед яких Transparent Data Encryption (TDE) для шифрування даних на рівні таблиць і файлів, Oracle Database Vault для реалізації принципу розділення обов'язків і обмеження привілейованого доступу, Oracle Label Security для мандатного контролю доступу, Oracle Key Vault

як централізований менеджер ключів шифрування, та Oracle Audit Vault and Database Firewall для моніторингу, аудиту й запобігання вторгненням. Комплексність цих механізмів є надзвичайно актуальною для забезпечення конфіденційності та цілісності біометричної інформації [77].

Ще однією перевагою є здатність Oracle Database підтримувати високу масштабованість та доступність системи через використання технологій Oracle RAC (Real Application Clusters), що дозволяє кільком серверам працювати з єдиною базою даних, а також Oracle Sharding, який реалізує горизонтальний розподіл даних за принципом шардингу. Такі рішення є ефективними у випадках масових запитів до великих баз біометричних шаблонів [78].

Із виходом Oracle Database 23ai система отримала підтримку функціоналу AI Vector Search, який орієнтований на роботу з векторними вбудовуваннями, а саме цифровими представленнями біометричних зразків. Ця функція дозволяє виконувати семантичний пошук за допомогою індексів HNSW та IVF, а також використовувати апаратне прискорення за допомогою графічних процесорів NVIDIA, що відкриває можливості для обробки великих обсягів векторних даних у режимі реального часу [78].

Втім, використання Oracle Database пов’язане з певними викликами. Насамперед, це висока вартість володіння, яка включає ліцензування, технічну підтримку та додаткові опції безпеки. Такий рівень витрат може стати обмеженням для невеликих організацій чи проєктів. Крім того, адміністративне управління системою вимагає спеціалізованої підготовки, що також впливає на вартість експлуатації та складність розгортання [78].

Слід зазначити, що векторний пошук активно підтримується всіма трьома розглянутими платформами. У PostgreSQL функціонал реалізовано через модуль pgvector, який забезпечує інтеграцію векторного пошуку з реляційними запитами SQL. MongoDB пропонує інструмент Atlas Vector Search у своїй хмарній платформі, базований на Apache Lucene, який об’єднує можливості семантичного пошуку з повнотекстовим аналізом. Oracle, у свою чергу, запровадила AI Vector

Search, що поєднує ефективність кластерного пошуку з підтримкою GPU-прискорення, забезпечуючи високу швидкодію в системах, орієнтованих на біометричну ідентифікацію [71; 75; 78].

З огляду на проведене порівняльне дослідження сучасних систем керування базами даних, доцільним є вибір PostgreSQL як платформи для зберігання та обробки біометричних даних. Такий вибір зумовлений комплексом технічних і практичних чинників, серед яких ключову роль відіграють баланс між функціональністю, продуктивністю, безпекою, гнучкістю масштабування та загальною вартістю володіння.

PostgreSQL, як об’єктно-реляційна система з відкритим вихідним кодом, демонструє високу відповідність вимогам, що висуваються до систем, що працюють з чутливою біометричною інформацією. Однією з головних переваг PostgreSQL є її підтримка транзакційної моделі, що відповідає принципам ACID, а також розширені засоби безпеки, включаючи рольову модель доступу, шифрування з’єднань, контроль доступу на рівні рядків та можливість використання TDE або pgcrypto для шифрування даних [69; 70]. Це робить систему здатною гарантувати цілісність та конфіденційність біометричних шаблонів протягом усього циклу їх обробки.

Особливо значущою для сучасних біометричних систем є підтримка PostgreSQL розширення pgvector, що дозволяє здійснювати зберігання та векторний пошук шаблонів (embeddings) з використанням ефективних метрик подібності та індексації (HNSW, IVF). Це дозволяє інтегрувати семантичний пошук безпосередньо в середовищі реляційної СКБД без потреби в додаткових зовнішніх сервісах [71].

Порівняно з MongoDB, яка хоч і надає потужні можливості векторного пошуку через Atlas Vector Search, поступається PostgreSQL за рівнем транзакційної строгості та підтримкою складної логіки запитів, що є критичним для систем, де важливе узгоджене оброблення множини біометричних даних [74]. Крім того, MongoDB у хмарному варіанті (Atlas), що містить ключові функції, є платною, на

відміну від PostgreSQL, який є безкоштовним продуктом з відкритим кодом, що значно знижує вартість володіння [76].

Що стосується Oracle Database, попри її визнану надійність та широкий арсенал засобів безпеки, включаючи інструменти керування ключами та мандатного доступу, її основним обмеженням залишається висока вартість ліцензування та обслуговування, а також складність розгортання й адміністрування, що робить її менш привабливою для невеликих або експериментальних проєктів [77, 78].

Таким чином, PostgreSQL виступає оптимальним рішенням для зберігання біометричних даних у системах з обмеженим бюджетом, які водночас потребують високої безпеки, підтримки векторного пошуку, масштабованості та прозорого адміністрування.

2.3 Архітектура та принцип дії системи контролю доступу

Розроблена система автоматичного контролю доступу базується на архітектурі з розподіленою обробкою даних, що реалізована у форматі клієнт-серверної моделі з чітким розмежуванням функціональних обов’язків між апаратними та програмними компонентами. Такий підхід дозволяє досягти високої гнучкості, масштабованості, енергоефективності й надійності під час роботи в умовах реального середовища, не вимагаючи при цьому складної мережевої інфраструктури або постійного підключення до віддалених серверів для виконання базових операцій ідентифікації. Ця архітектурна парадигма забезпечує оптимальний розподіл обчислювальних ресурсів, де периферійні пристрої виконують збір та первинну обробку даних, тоді як центральний серверний компонент відповідає за складні операції порівняння та управління базою даних.

Центральною керуючою одиницею фізичного рівня, що відповідає за безпосередню взаємодію з користувачем та периферійними модулями, є мікроконтролер Arduino Uno R3, побудований на основі 8-бітного мікропроцесора

ATMega328P. Даний мікроконтролер забезпечує локальне детерміноване керування всіма апаратними компонентами точки доступу, включно з біометричним сенсором відбитків пальців, виконавчими пристроями та засобами візуального зворотного зв’язку. Вибір саме цього мікроконтролера обгрунтований його відкритою апаратною платформою, що сприяє легкості інтеграції та модифікації, широкою підтримкою з боку розробницької спільноти, що надає доступ до великої кількості бібліотек та прикладів коду, відносно невисоким енергоспоживанням, що є критичним для систем, які можуть живитися від обмежених джерел, та достатнім обсягом флеш-пам’яті (32 КБ) і оперативної пам’яті (2 КБ SRAM) для реалізації логіки керування, збору та передачі даних у режимі, близькому до реального часу [45; 47]. Мікроконтролер виконує роль інтерфейсного шлюзу між сенсорним модулем та серверною частиною системи.

Процес ідентифікації користувача ініціюється шляхом прикладання пальця до оптичного сенсора FPM10A. Цей модуль виконує зчитування папілярного візерунка пальця та його подальше оцифрування [53; 54]. Важливою особливістю модуля FPM10A є наявність вбудованого цифрового сигнального процесора (DSP), який дозволяє здійснювати попередню обробку біометричних зображень безпосередньо на сенсорі [53]. Ця обробка включає покращення якості зображення, виділення характерних ознак (мінуцій) та формування компактного цифрового шаблону (template) відбитка пальця. Таке апаратне рішення значно знижує обчислювальне навантаження на основний мікроконтролер. Отримані цифрові шаблони, що є математичним поданням унікальних характеристик відбитка, передаються з модуля FPM10A на мікроконтролер Arduino Uno R3 через асинхронний послідовний інтерфейс UART (Universal Asynchronous Receiver/Transmitter). Після отримання шаблону мікроконтролер формує запит на верифікацію, який інкапсулює отриманий біометричний шаблон. У деяких конфігураціях запит може також містити ідентифікаційні ознаки або ID користувача, якщо система працює в режимі 1:1 верифікації, однак у даній реалізації передбачається режим 1:N ідентифікації.

Сформований запит передається з мікроконтролера на персональний комп’ютер (ПК), який у контексті даної архітектури виступає як логічний сервер системи, відповідальний за зберігання та обробку біометричної бази даних. Передача даних між Arduino та ПК здійснюється через стандартний інтерфейс Universal Serial Bus (USB) [45]. Це з’єднання, крім функції двонаправленої комунікації, також виконує критично важливу роль єдиного джерела живлення для мікроконтролера Arduino та всіх підключених до нього периферійних модулів, таких як сенсор FPM10A та LCD-дисплей. На персональному комп’ютері функціонує спеціалізований програмний модуль-посередник (middleware). Цей програмний компонент відповідає за прийом та обробку вхідних запитів, що надходять від мікроконтролера через USB-порт, взаємодію з системою управління базами даних (СУБД) PostgreSQL, виконання операцій порівняння переданих біометричних шаблонів з наявними у базі даних та прийняття остаточного рішення про дозвіл або заборону доступу. Сама база даних PostgreSQL, розгорнута на ПК, зберігає шаблони відбитків пальців зареєстрованих користувачів у спеціалізованому векторизованому форматі. Це стало можливим завдяки використанню розширення pgvector для PostgreSQL, яке оптимізує зберігання та пошук векторних подань, що є надзвичайно ефективним для задач пошуку за подібністю в біометричних системах, забезпечуючи високу швидкість і точність ідентифікації навіть при великих обсягах даних.

Після завершення процесу пошуку та прийняття рішення на основі результатів порівняння (зазвичай, з використанням певного порогу схожості), програмне забезпечення на ПК формує структуровану відповідь. Ця відповідь включає результат аутентифікації (наприклад, кодовані повідомлення “доступ дозволено”, “користувача не знайдено в базі даних”, “відмовлено в доступі - низька схожість шаблонів”) та, за потреби, додаткову службову інформацію або статус помилки. Ця відповідь повертається на мікроконтролер Arduino через той самий USB-канал. Після отримання та декодування результату верифікації, мікроконтролер Arduino ініціює виведення відповідного текстового повідомлення

на рідкокристалічний дисплей LCD 1602. Цей дисплей, як правило, є символьним дисплеєм формату 2 рядки по 16 символів і підключений до мікроконтролера через двопровідний послідовний інтерфейс I2C (Inter-Integrated Circuit). Використання I2C дозволяє мінімізувати кількість задіяних цифрових виводів мікроконтролера, що є важливим при обмеженій їх кількості, та забезпечує достатню швидкість передачі даних для оновлення інформації на дисплеї [49; 50]. Це забезпечує швидкий та інтуїтивно зрозумілий візуальний зворотний зв'язок для користувача, мінімізуючи при цьому навантаження на апаратні ресурси мікроконтролера.

Важливою та елегантною особливістю запропонованої архітектури є використання персонального комп'ютера не лише як логічного хоста для ресурсоємної бази даних та обчислювального ядра системи верифікації, але й як централізованого фізичного джерела енергії для всієї периферійної частини системи контролю доступу. Стандартний USB-інтерфейс, через який мікроконтролер Arduino підключено до ПК, одночасно забезпечує стабільне живлення (зазвичай 5В) для самого мікроконтролера, оптичного сканера відбитків пальців FPM10A та рідкокристалічного дисплея LCD 1602. Таке рішення суттєво спрощує загальну електричну схему пристрою, зменшує кількість необхідних компонентів (наприклад, окремих блоків живлення, стабілізаторів напруги), знижує загальну вартість системи та мінімізує потенційні ризики, пов'язані з нестабільним або недостатнім енергозабезпеченням окремих модулів.

Дане архітектурне рішення, що детально відображене на структурній схемі (Рисунок 2.9), базується на фундаментальних принципах модульності, що дозволяє легко замінювати або модернізувати окремі компоненти системи без суттєвого впливу на інші. Також важливим є принцип мінімізації обчислювального навантаження на мікроконтролер шляхом делегування ресурсоємних задач (порівняння біометричних шаблонів, управління великими масивами даних) на потужнішу обчислювальну платформу, якою виступає персональний комп'ютер. Централізація складних операцій у межах ПК забезпечує високу швидкодію процесу ідентифікації, ефективну обробку значних обсягів біометричних даних та

надає гнучкість у подальшому масштабуванні системи. Наприклад, така архітектура дозволяє відносно легко розширити систему до мережі з кількох точок доступу, що керуються одним центральним сервером, або інтегрувати додаткові методи автентифікації. Це також спрощує обслуговування та оновлення програмного забезпечення системи. Усі апаратні та програмні модулі системи спроектовані для узгодженої роботи, формуючи єдиний інтегрований апаратно-програмний комплекс. Даний комплекс здатний ефективно виконувати функцію біометричної ідентифікації користувачів та здійснювати контроль доступу з дотриманням сучасних вимог до безпеки передачі та зберігання даних, доступності системи для користувачів та загальної ергономіки взаємодії.

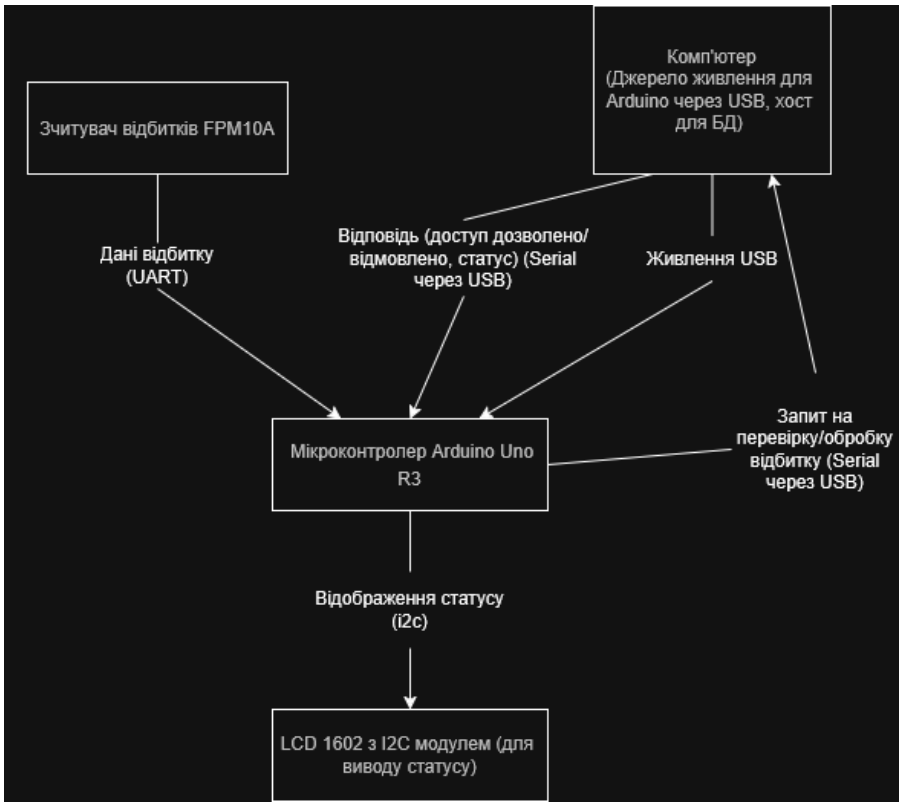


Рисунок 2.9 – Структурна схема системи контролю доступу

На рисунку 2.9 представлена структурна схема розробленої системи контролю доступу. Ця схема візуалізує ключові функціональні блоки: пристрій біометричної ідентифікації (що включає сенсор FPM10A, мікроконтролер Arduino Uno R3 та LCD 1602) та зовнішній обчислювальний вузол (ПК з базою даних

PostgreSQL та програмним посередником). Схема чітко ілюструє напрямки потоків даних: від зчитування відбитку сенсором FPM10A (UART до Arduino), передачі запиту на верифікацію (Serial через USB до ПК), обробки на ПК, передачі відповіді назад на мікроконтролер (Serial через USB) та відображення статусу на LCD (I2C). Також показано канал живлення через USB, що підкреслює інтегрований підхід до енергозабезпечення. Така візуалізація дозволяє наочно зрозуміти логіку взаємодії компонентів та загальний принцип функціонування системи.

2.4 Висновок

У межах другого розділу було розроблено та теоретично обгрунтовано архітектуру автоматизованої системи контролю доступу, що базується на біометричній ідентифікації користувачів за відбитками пальців.

Для забезпечення функціонування апаратної частини системи було здійснено ретельний аналіз та вибір ключових електронних компонентів. На основі порівняння характеристик поширених мікроконтролерних платформ, таких як Arduino UNO R3, STM32 та ESP32, було обрано Arduino Uno R3 як оптимальне рішення для прототипування завдяки його простоті використання, доступності та достатній функціональності для поставлених завдань. Аналогічно, після розгляду різних типів дисплеїв (символьний LCD, графічний LCD, OLED) та сенсорів відбитків пальців (оптичні, ємнісні, ультразвукові), було обгрунтовано вибір символьного дисплея LCD 1602 для виведення текстових повідомлень та оптичного сенсора FPM10A для зчитування біометричних даних, як компроміс між вартістю, функціональністю та простотою інтеграції на етапі розробки. Економічна доцільність обраної компонентної бази була підтверджена розрахунком сукупної вартості основних апаратних засобів.

Важливу увагу було приділено питанням безпечного та ефективного зберігання біометричних даних. Було досліджено специфіку біометричних шаблонів, сучасні підходи до їх зберігання та ключові вимоги до систем керування

базами даних (СКБД), що працюють з такою чутливою інформацією. На основі детального порівняльного аналізу можливостей, переваг та недоліків провідних СКБД, зокрема PostgreSQL, MongoDB та Oracle Database, з акцентом на їхні засоби безпеки, масштабованість, продуктивність та підтримку векторного пошуку, було обгрунтовано вибір об'єктно-реляційної системи PostgreSQL. Таке рішення забезпечує надійне зберігання векторних представлень біометричних шаблонів, ефективний пошук за подібністю, відповідність принципам ACID та високий рівень безпеки даних при оптимальній вартості володіння.

Спроєктована архітектура системи контролю доступу реалізована за клієнт-серверною моделлю з чітким розподілом функціональних обов'язків. Мікроконтролерний модуль на базі Arduino Uno R3 виступає в ролі клієнта, відповідального за збір біометричних даних з сенсора FPM10A, їх первинну обробку, формування запитів на верифікацію та відображення результатів на LCD-дисплеї. Персональний комп'ютер виконує функції сервера, на якому функціонує програмний модуль-посередник та розгорнута база даних PostgreSQL. Взаємодія між клієнтом та сервером здійснюється через USB-інтерфейс, який також забезпечує живлення для периферійних компонентів клієнтської частини. Було детально описано принцип дії системи, починаючи від прикладання пальця до сенсора, передачі даних через UART та USB, обробки запиту на сервері з використанням pgvector для порівняння шаблонів, і до виведення результату ідентифікації користувачеві. Розроблена архітектура характеризується модульністю, що спрощує її подальшу модифікацію, та раціональним розподілом обчислювального навантаження між мікроконтролером і потужнішою серверною платформою.

Таким чином, в результаті проведеної роботи було створено комплексну апаратно-програмну архітектуру системи біометричного контролю доступу. Ця архітектура забезпечує виконання ключових функцій ідентифікації користувачів за відбитками пальців і слугує надійною основою для подальшої практичної реалізації та потенційного вдосконалення системи.

3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ

3.1 Електрична принципова схема та архітектура програмно-апаратного комплексу

Розробка будь-якої електронної системи невіддільно пов'язана зі створенням її електричної принципової схеми. Ця схема є фундаментальним графічним документом, що детально відображає всі електричні з’єднання та взаємозв’язки між окремими компонентами пристрою. Вона слугує не лише візуальним посібником для розуміння архітектури системи, але й є критично важливою основою для коректного монтажу, тестування та подальшого налагодження розроблюваного обладнання. У контексті проєктування системи автоматичного контролю доступу, електрична схема наочно демонструє спосіб підключення та взаємодії ключових функціональних вузлів: мікроконтролерної платформи, модуля візуального виведення інформації та біометричного сенсора.

Центральним елементом розроблюваної системи, що виконує функції обробки даних, управління периферійними пристроями та реалізації логіки контролю доступу, є мікроконтролер Arduino Uno на базі мікросхеми ATmega328P. Ця платформа обрана через її доступність, широку підтримку спільнотою розробників та достатню продуктивність для вирішення поставлених задач.

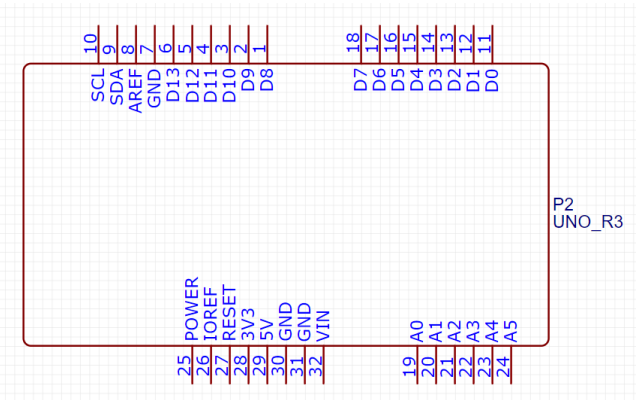


Рисунок 3.1 – Мікроконтролерна плата Arduino Uno R3

Для забезпечення інтеракції з користувачем та візуального відображення інформації про поточний стан системи (наприклад, запит на сканування пальця, результат ідентифікації, системні повідомлення) використовується рідкокристалічний дисплей LCD 16x2 з інтегрованим інтерфейсом I2C (Inter-Integrated Circuit). Даний тип дисплея дозволяє виводити два рядки по 16 символів. Використання I2C модуля є перевагою, оскільки значно спрощує підключення до мікроконтролера, вимагаючи лише дві лінії для передачі даних, на відміну від стандартного паралельного підключення, що потребує більшої кількості цифрових виводів.

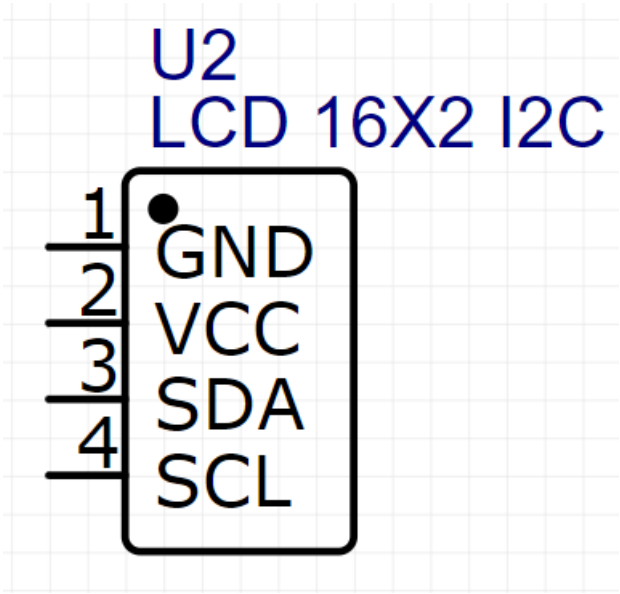


Рисунок 3.2 – Рідкокристальний дисплей LCD 16x2 з модулем I2C

Підключення LCD 16x2 I2C до мікроконтролера Arduino Uno здійснюється за допомогою чотирьох основних провідників. Лінія живлення дисплея, позначена як VCC (Voltage Common Collector), підключається до виводу +5V на платі Arduino, що забезпечує необхідну напругу для роботи дисплея. Лінія заземлення, позначена як GND (Ground), з'єднується з одним із відповідних виводів GND мікроконтролера, створюючи спільну точку відліку потенціалів. Передача даних та команд керування між дисплеєм та Arduino Uno відбувається по двопровідній шині I2C. Лінія даних SDA (Serial Data) дисплея підключається до аналогового виводу

A4 мікроконтролера, який на Arduino Uno також може функціонувати як лінія SDA шини I2C. Лінія тактування SCL (Serial Clock) дисплея, що синхронізує передачу даних, з'єднується з аналоговим виводом A5 мікроконтролера, який відповідно виконує функцію SCL. Таке підключення є стандартним для I2C пристроїв на платформі Arduino та дозволяє ефективно керувати дисплеєм, мінімізуючи кількість задіяних виводів мікроконтролера.

Біометрична ідентифікація користувачів, що є ключовою функцією системи, реалізується за допомогою оптичного зчитувача відбитків пальців FPM10A. Цей модуль здатен сканувати, обробляти, зберігати та порівнювати шаблони відбитків пальців. Взаємодія модуля FPM10A з мікроконтролером Arduino Uno відбувається через послідовний інтерфейс UART (Universal Asynchronous Receiver-Transmitter).

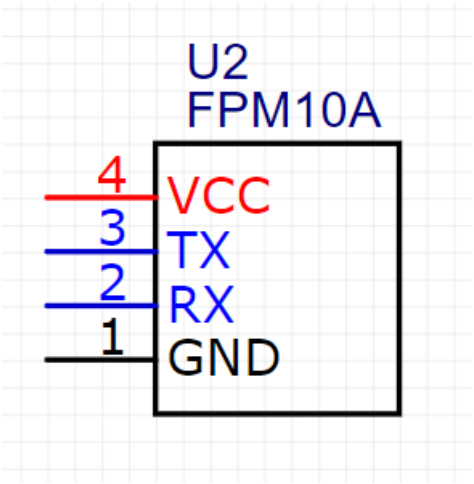


Рисунок 3.3 – Оптичний зчитувач відбитків пальців FPM10A

Для живлення модуля FPM10A його вивід VCC підключається до виводу +3.3V на платі Arduino Uno. Важливо зазначити, що цей модуль працює саме від напруги 3.3В, тому підключення до 5В може призвести до його пошкодження. Лінія заземлення GND модуля з’єднується з виводом GND мікроконтролера для забезпечення спільного нульового потенціалу. Обмін даними між модулем та мікроконтролером здійснюється по двох лініях. Вивід передавача даних модуля TX (Transmit) підключається до цифрового виводу A2 мікроконтролера Arduino Uno, який у програмному коді буде налаштований як приймач (RX) для програмної

реалізації послідовного порту (SoftwareSerial). Відповідно, вивід приймача даних модуля RX (Receive) з'єднується з цифровим виводом A3 мікроконтролера Arduino Uno, який буде налаштований як передавач (TX) для того ж програмного послідовного порту. Такий підхід (використання SoftwareSerial на виводах A2 та A3) дозволяє залишити апаратний UART (виводи 0-RX та 1-TX) вільним для програмування мікроконтролера або для комунікації з комп'ютером через USB. Дана схема з'єднання забезпечує надійний двосторонній обмін даними, необхідний для надсилання команд модулю (наприклад, команда сканування, реєстрації нового відбитка) та отримання відповідей (наприклад, результат порівняння, ідентифікатор користувача).

Загальна електрична принципова схема, що об’єднує всі вищезгадані компоненти та їхні з’єднання, є основою для апаратної реалізації системи контролю доступу. Вона забезпечує коректну передачу сигналів живлення та даних між мікроконтролером, дисплеєм та біометричним сенсором, що є необхідною умовою для стабільного та надійного функціонування всієї системи.

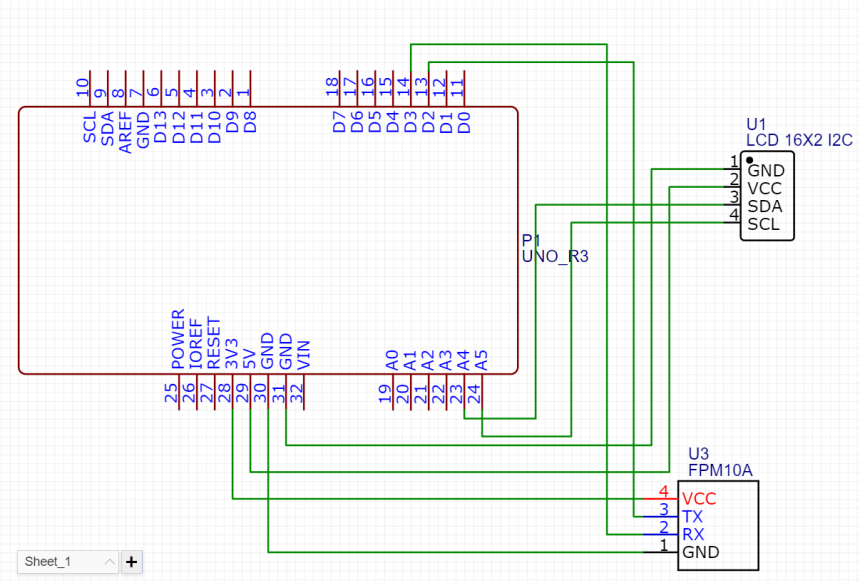


Рисунок 3.4 – Загальна електрична принципова схема системи контролю доступу

Представлена електрична схема з’єднання є базовою для функціонування розроблюваної системи контролю доступу, забезпечуючи коректну взаємодію всіх

її апаратних компонентів та створюючи передумови для подальшої програмної реалізації алгоритмів контролю та управління.

Взаємодія мікроконтролерної системи на базі Arduino з серверною частиною, що включає базу даних PostgreSQL з розширенням pgvector, при використанні прямого USB-підключення до персонального комп'ютера (ПК) та HTTP-протоколу для обміну даними, представляє собою специфічну архітектуру. У цій конфігурації ПК виконує роль як шлюзу для мікроконтролера, так і хост-системи для серверних компонентів.

Мікроконтролерний вузол, представлений платою Arduino, підключеною до біометричного сенсора FPM10A та пристрою виведення інформації (LCD-дисплей), відповідає за збір первинних біометричних даних. Після отримання даних від сенсора, Arduino здійснює їх передачу на ПК через інтерфейс USB, використовуючи протокол послідовної передачі даних (virtual COM port). Дані, що передаються, можуть бути сирими даними шаблону або попередньо обробленим ідентифікатором.

Код мікроконтролера, що реалізує зчитування біометричних шаблонів, взаємодію із сенсором FPM10A, передавання даних через USB, а також виведення результатів на LCD-дисплей, розробляється в середовищі Arduino IDE з використанням мови програмування C/C++. Arduino також обробляє вхідні команди (наприклад, статус результату ідентифікації), що надходять з ПК у відповідь, та завершує цикл взаємодії.

На персональному комп'ютері функціонує спеціалізований програмний компонент, а саме клієнтський застосунок, написаний на мові Python. Він виконує дві ключові функції. По-перше, постійно прослуховує визначений COM-порт для отримання даних, що надходять від Arduino. По-друге, після отримання та розбору даних, ініціює HTTP-запит (наприклад, POST-запит) до локального серверного API, реалізованого за допомогою FastAPI.

Серверний API, який також розгорнутий на цьому ж ПК (наприклад, як локальний веб-сервер, що слухає localhost на певному порту), реалізує бізнес-

логіку системи: валідує отримані дані, здійснює перетворення біометричного шаблону у векторне представлення (embedding), сумісне з pgvector, та формує відповідні SQL-запити до бази даних PostgreSQL.

Система керування базами даних PostgreSQL, з активованим розширенням pgvector, розміщена на ПК або віддаленому сервері. При додаванні нового користувача, API надсилає команду INSERT для збереження векторного представлення відбитка пальця у відповідній таблиці з типом даних vector. Під час ідентифікації API формує запит SELECT із застосуванням операторів подібності pgvector (наприклад, <-> для евклідової відстані) для порівняння векторів та знаходження найближчого збігу. Pgvector забезпечує ефективний пошук завдяки підтримці спеціалізованих індексів (HNSW, IVFFlat).

Після виконання операції базою даних, результат повертається до API. API, у свою чергу, формує HTTP-відповідь (наприклад, у форматі JSON), що містить статус операції або ідентифіковані дані, та відправляє її клієнтському застосунку. Клієнтський застосунок, отримавши відповідь, передає релевантну інформацію назад на Arduino через USB-серійний зв'язок. Arduino виводить результат на LCD-дисплей.

Загальна логіка взаємодії між компонентами системи представлена на блок-схемі нижче (рис. 3.5), яка ілюструє передачу даних між мікроконтролером Arduino, ПК із серверною логікою на Python/FastAPI та базою даних PostgreSQL з розширенням pgvector.

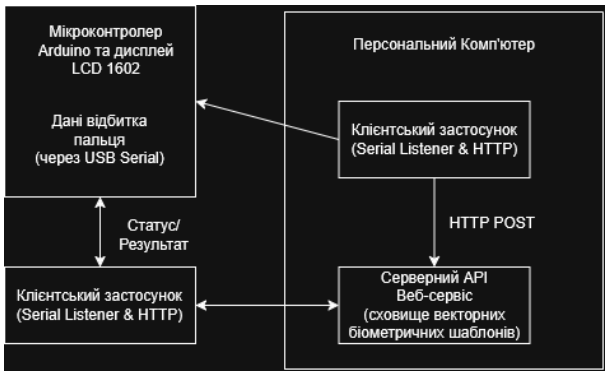


Рисунок 3.5 – Блок-схема інтеграції мікроконтролера Arduino з базою даних PosgreSQL

3.2 Розробка алгоритму роботи системи

Для забезпечення коректної та послідовної роботи розроблюваної системи автоматичного контролю доступу, ключовим етапом є формалізація її функціональної логіки у вигляді чіткого алгоритму. Блок-схема, що аналізується нижче, візуалізує саме таку логічну послідовність операцій, необхідних для ідентифікації користувача за біометричними даними відбитка пальця. Даний алгоритм розроблено з урахуванням специфіки реалізованої клієнт-серверної архітектури, в якій мікроконтролерний вузол виконує завдання збору первинних даних та взаємодії з користувачем, тоді як серверний компонент, що базується на персональному комп'ютері, відповідає за управління базою даних біометричних шаблонів та прийняття рішень щодо доступу.

Функціонування системи згідно з алгоритмом розпочинається зі стартового стану, що сигналізує про готовність до прийому запиту на ідентифікацію. Першою операцією є збір біометричних даних, представлений блоком введення даних, де оптичний сенсор FPM10A виконує сканування папілярного візерунка пальця та генерує його унікальний цифровий шаблон.

Рисунок 3.6 – Ініціалізація та сканування відбитка

Після успішного створення шаблону, мікроконтролер Arduino виконує операцію процесу, приймаючи ці дані від сенсорного модуля. Наступним кроком є ще одна операція процесу, під час якої мікроконтролер передає отриманий біометричний шаблон на персональний комп'ютер, що виконує роль сервера, для подальшої обробки.

Рисунок 3.7 – Обробка та передача шаблону мікроконтролером

На серверній стороні ініціюється операція взаємодії з базою даних. На цьому етапі відбувається зіставлення переданого шаблону з усіма біометричними шаблонами, що зберігаються в базі даних PostgreSQL, для виявлення потенційної відповідності. Результат цього порівняння слугує основою для прийняття рішення.

Рисунок 3.8 – Серверна верифікація відбитка

Якщо в результаті перевірки встановлено, що відбиток знайдено в базі даних і він є валідним, система переходить до операції процесу, що полягає в інформуванні користувача про успішну ідентифікацію та надання доступу. Після цього процес для даної сесії завершується, що позначено кінцевим станом. У протилежному випадку, якщо відбиток не знайдено або він не пройшов валідацію, виконується альтернативна операція процесу, інформуючи користувача про відмову в доступі, після чого алгоритм також переходить до кінцевого стану.

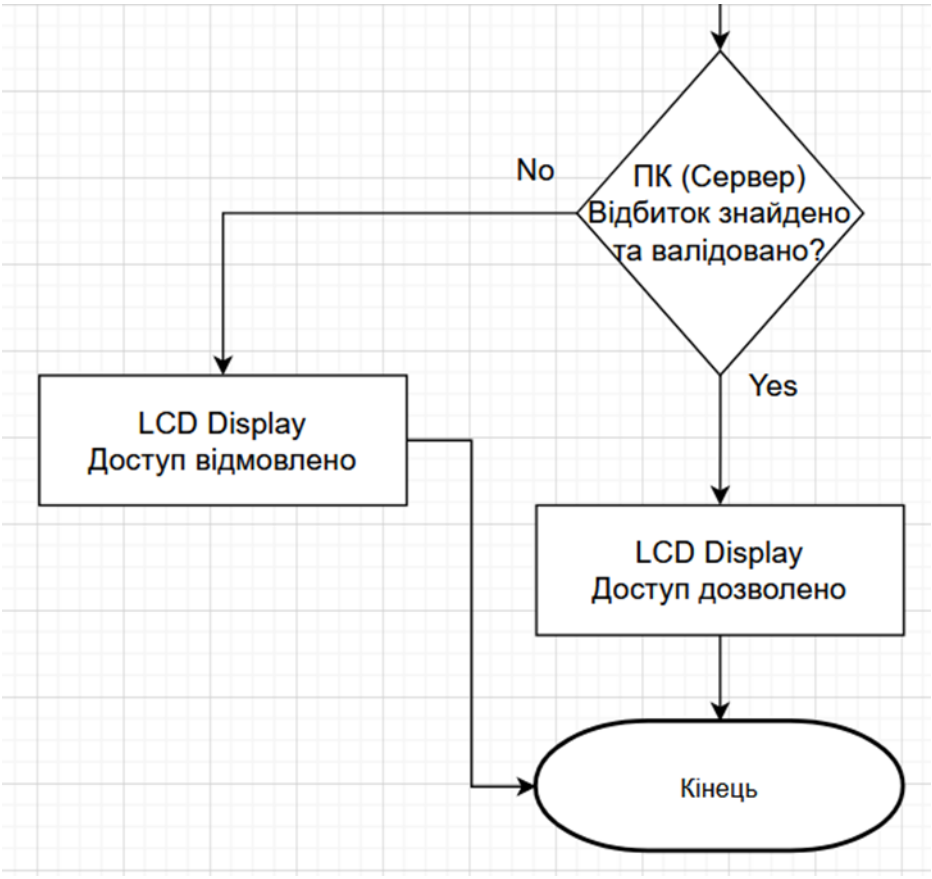


Рисунок 3.9 – Прийняття рішення та результати ідентифікації

Запропонований алгоритм, представлений у вигляді блок-схеми, починається з ініціалізації процесу шляхом сканування відбитка пальця за допомогою сенсора FPM10A, який також генерує цифровий шаблон. Цей шаблон передається через мікроконтролер Arduino на серверний компонент, реалізований на ПК. На сервері відбувається порівняння отриманого шаблону з базою даних PostgreSQL. Залежно

від результату порівняння чи знайдено валідний відповідник система приймає рішення про надання або відмову в доступі. Фінальний результат ідентифікації доводиться до відома користувача, після чого алгоритм завершує свою роботу для поточного запиту. Така послідовність операцій забезпечує логічно завершений цикл біометричної ідентифікації в рамках розробленої системи контролю доступу.

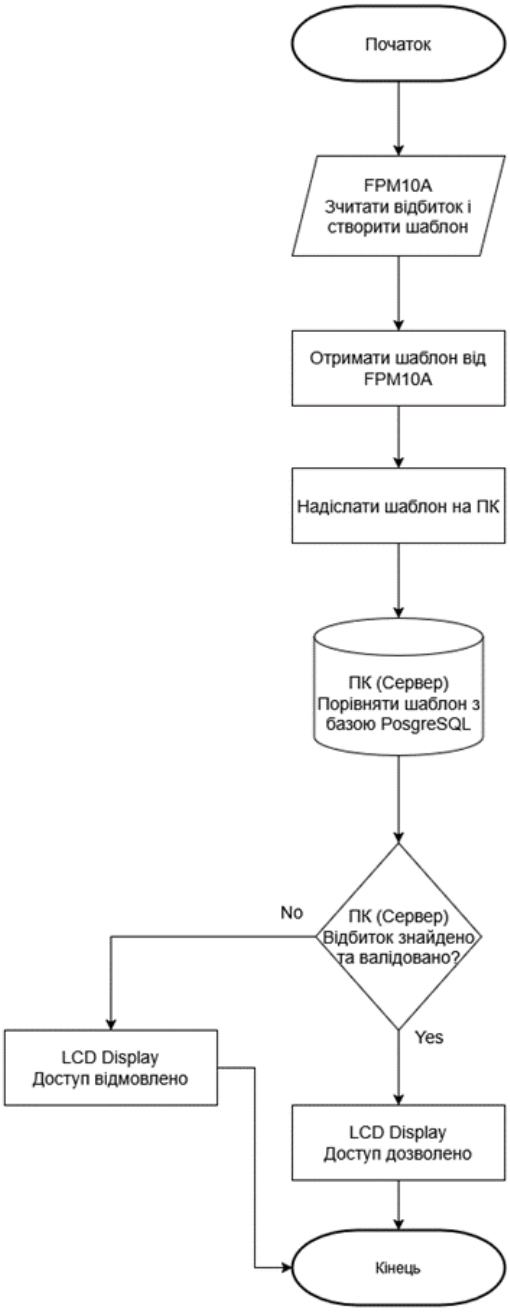


Рисунок 3.10 – Блок-схема алгоритму ідентифікації користувача в системі контролю доступу

Процедура зчитування та первинної обробки біометричного шаблону відбитка пальця є фундаментальним етапом у функціонуванні розроблюваної системи контролю доступу. Вона забезпечує отримання унікальних ідентифікаційних даних користувача безпосередньо з фізичного носія - його пальця. Цей алгоритм реалізується на мікроконтролерному вузлі Arduino Uno R3 у тісній взаємодії з оптичним сенсором FPM10A і включає послідовність команд керування, операцій збору даних та базової обробки помилок. Точність та надійність виконання цього алгоритму безпосередньо впливають на загальну ефективність системи ідентифікації.

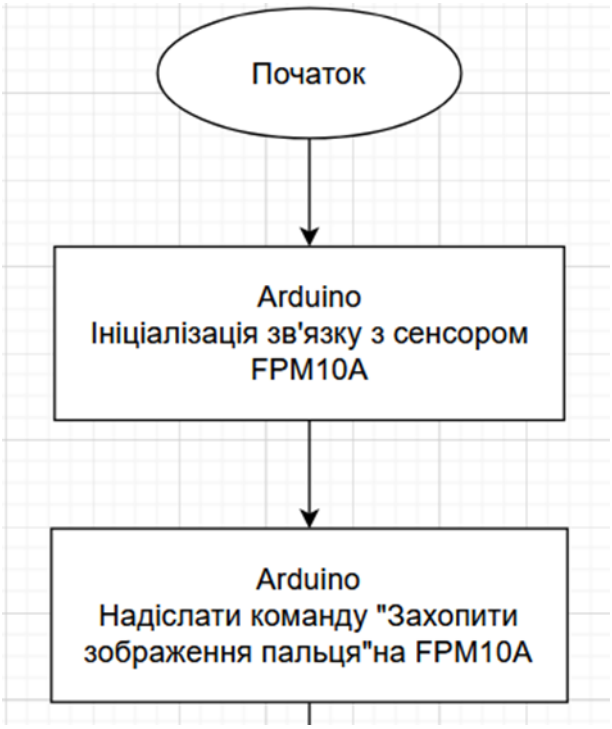


Рисунок 3.11 – Ініціалізація зв’язку та надсилання команди сенсору FPM10A

Логіка алгоритму зчитування активується стартовою подією, що переводить мікроконтролерний вузол у режим очікування або безпосереднього виконання процедури сканування. Мікроконтролер Arduino, виконуючи операцію процесу, спершу здійснює ініціалізацію програмного послідовного інтерфейсу для комунікації з модулем FPM10A. Це включає налаштування швидкості передачі даних та визначення цифрових виводів, що будуть використані для прийому (RX)

та передачі (TX) сигналів. Після успішної ініціалізації зв'язку, Arduino формує та надсилає на сенсор FPM10A специфічну команду процесу, яка інструктує сенсор активувати режим захоплення зображення. Ця команда передається у вигляді послідовності байтів, визначених протоколом комунікації сенсора. Важливим аспектом на цьому етапі є забезпечення коректної передачі команди та очікування можливого підтвердження від сенсора про її отримання, що може включати короткочасну затримку для стабілізації стану сенсора.

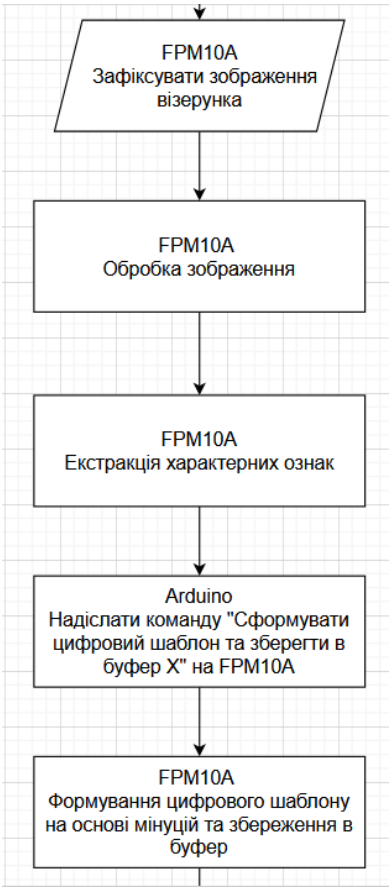


Рисунок 3.12 – Процес сканування, внутрішньої обробки та генерації біометричного шаблону сенсором FPM10A

У відповідь на отриману команду, оптичний сенсор FPM10A виконує операцію введення даних, що полягає в активації внутрішнього світлодіодного джерела для освітлення поверхні пальця, прикладеного до скануючої призми, та фіксації зображення папілярного візерунка за допомогою вбудованої КМОН-

камери. Отримане сире зображення зазнає низки перетворень безпосередньо на борту сенсора завдяки його інтегрованому цифровому сигнальному процесору (DSP). Ці операції процесу включають алгоритми цифрової обробки сигналів, такі як придушення шумів, покращення контрастності, вирівнювання гістограми та бінаризацію зображення для чіткого виділення папілярних ліній. Наступним критичним кроком є процес екстракції характерних ознак, або мінуцій, що є унікальними точками на відбитку, такими як закінчення та розгалуження ліній. Мікроконтролер Arduino, після ймовірної короткої паузи, необхідної сенсору для захоплення та первинної обробки зображення, надсилає наступну команду процесу, що інструктує сенсор FPM10A сформувати на основі виділених мінуцій цифровий біометричний шаблон та зберегти його в одному зі своїх внутрішніх буферів пам'яті (наприклад, CharBuffer1 або CharBuffer2). Цей процес генерації шаблону також виконується DSP сенсора, результатом чого є компактне математичне представлення унікальних рис відбитка.

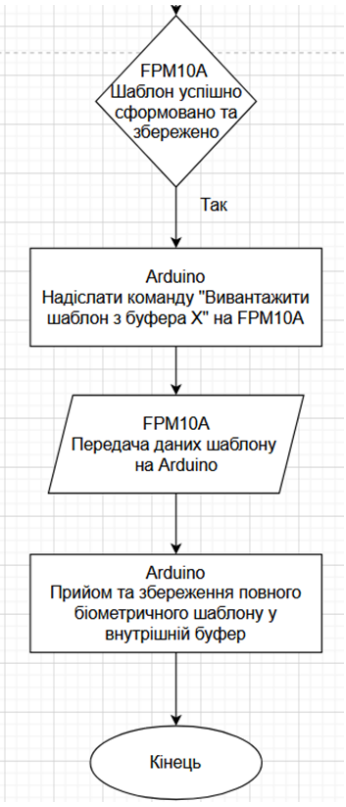


Рисунок 3.13 – Успішне формування та передача біометричного шаблону з FPM10A на мікроконтролер Arduino

Після команди на генерацію шаблону, сенсор FPM10A повертає статус виконання операції. Алгоритм на мікроконтролері Arduino переходить до блоку прийняття рішення, де аналізується цей статус для перевірки успішності формування та збереження шаблону сенсором. У випадку позитивного результату, що свідчить про коректне завершення попередніх етапів, мікроконтролер Arduino надсилає сенсору FPM10A наступну команду процесу, тобто команду на вивантаження (передачу) сформованого шаблону з вказаного буфера пам'яті сенсора. У відповідь, модуль FPM10A ініціює операцію виведення даних, передаючи послідовність байтів, що складають біометричний шаблон, на мікроконтролер Arduino через встановлений UART-зв'язок. Мікроконтролер Arduino, у свою чергу, виконує операцію процесу, приймаючи ці байти та акумулюючи їх у своєму внутрішньому масиві (буфері SRAM) до отримання повного шаблону. Розмір шаблону є фіксованим і відомим з документації на сенсор, що дозволяє контролювати повноту отриманих даних. Завершенням цієї гілки є кінцевий стан, що означає успішне отримання біометричного шаблону мікроконтролером.

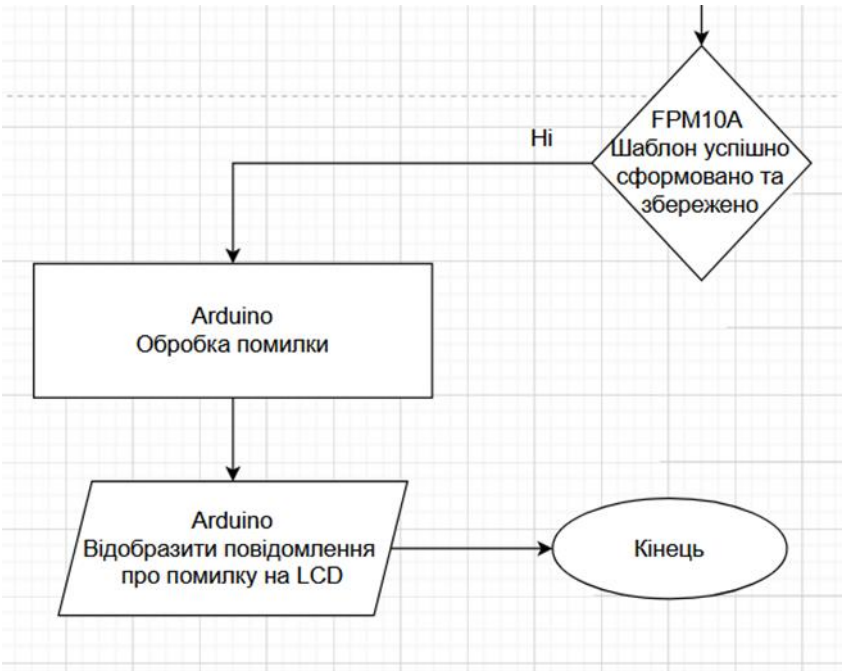


Рисунок 3.14 – Обробка помилок на етапі генерації та отримання шаблону відбитка

Якщо ж у блоці прийняття рішення статус від сенсора FPM10A вказує на помилку під час формування або збереження шаблону, система переходить до процедури обробки нештатної ситуації. Мікроконтролер Arduino виконує операцію процесу, яка може включати логування типу помилки або встановлення відповідного внутрішнього прапорця стану. Для інформування користувача про проблему, Arduino здійснює операцію виведення даних, надсилаючи команду на LCD-дисплей для відображення відповідного повідомлення, наприклад, “Помилка сканування” або “Спробуйте ще раз”. Після цього даний цикл зчитування також завершується, переходячи у кінцевий стан, сигналізуючи про неможливість отримання валідного біометричного шаблону на поточній ітерації.

Таким чином, деталізований алгоритм зчитування та первинної обробки біометричного шаблону мікроконтролерним вузлом охоплює повний цикл взаємодії з сенсором FPM10A, від ініціалізації сканування до отримання готового для передачі цифрового шаблону або обробки стану помилки. Ця послідовність операцій, що включає надсилання команд, аналіз відповідей сенсора, отримання та тимчасове збереження даних, є критично важливою для забезпечення системи якісними вхідними біометричними даними. Повна логічна структура цього підпроцесу, що об'єднує всі розглянуті етапи та розгалуження, представлена на відповідній узагальнюючій блок-схемі.

Після успішного завершення процедури зчитування та формування біометричного шаблону на мікроконтролерному вузлі, наступним фундаментальним етапом загального алгоритму ідентифікації є передача цих даних на серверний компонент та їх подальша обробка. Саме на сервері реалізується найбільш обчислювально складна частина, а саме порівняння отриманого шаблону з великою базою даних еталонних відбитків та прийняття остаточного рішення про надання або відмову в доступі. Детальний розгляд алгоритмічних аспектів функціонування серверної частини системи є необхідним для повного розуміння всього процесу ідентифікації.

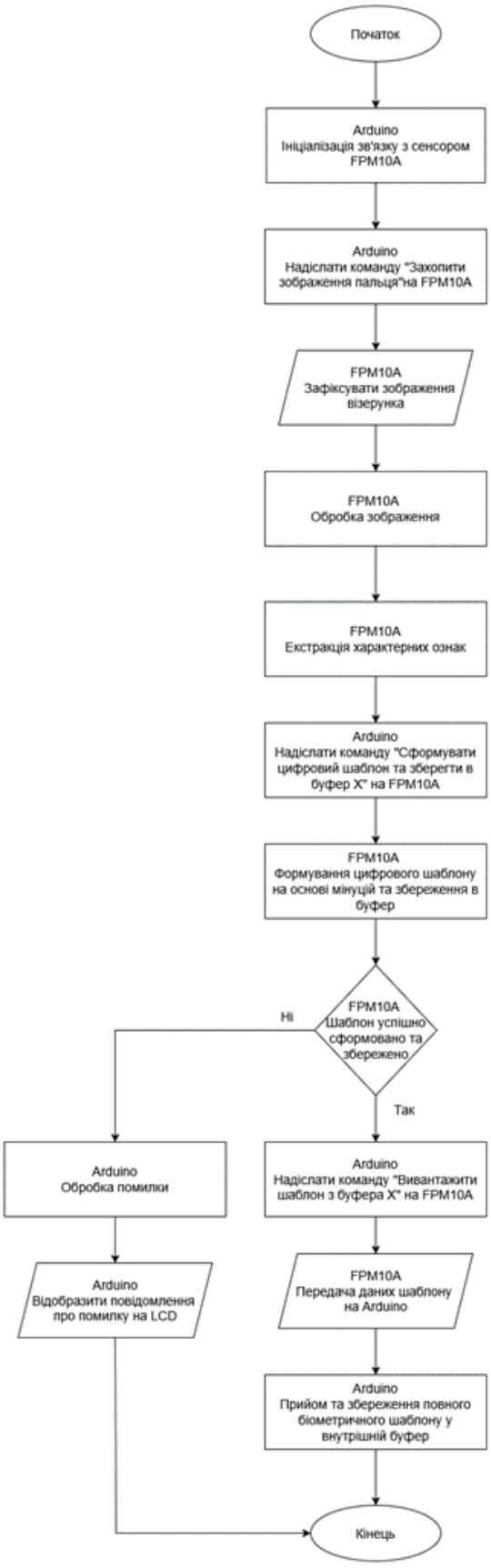


Рисунок 3.15 – Блок-схема алгоритму зчитування та первинної обробки біометричного шаблону відбитка пальця

Етап серверної ідентифікації є центральною обчислювальною частиною системи контролю доступу, де відбувається верифікація отриманого від мікроконтролерного вузла біометричного шаблону шляхом його порівняння з еталонними шаблонами, що зберігаються в захищеній базі даних. Цей алгоритм реалізується на персональному комп'ютері, що виконує роль сервера, і включає етапи прийому даних, їх перетворення, взаємодію з системою управління базами даних PostgreSQL, застосування спеціалізованих алгоритмів пошуку за подібністю (pgvector) та формування кінцевого рішення про надання чи відмову в доступі.

Рисунок 3.16 – Передача біометричного шаблону від мікроконтролера Arduino до серверного API через ПК-клієнт

Процес серверної ідентифікації активується стартовою подією, що відповідає моменту отримання біометричного шаблону від мікроконтролера Arduino на персональний комп'ютер. Спеціалізований клієнтський застосунок на ПК, написаний на Python, здійснює операцію введення даних, зчитуючи послідовність байтів шаблону з віртуального COM-порту, що емулюється через USB-з'єднання. Після отримання повного пакету даних, цей застосунок виконує операцію процесу, що включає первинний розбір отриманих даних для перевірки їх цілісності та

відповідності очікуваному формату, а також базову валідацію. Успішно розібраний та провалідований шаблон далі інкапсулюється у тіло HTTP POST-запиту, який Python-клієнт, виконуючи операцію процесу, надсилає на локально розгорнутий серверний API, реалізований за допомогою FastAPI. Цей запит адресується до конкретної кінцевої точки (endpoint) API, призначеної для обробки запитів на ідентифікацію.

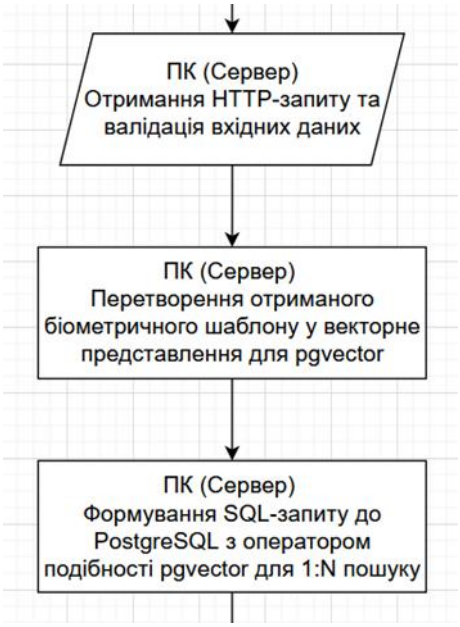


Рисунок 3.17 – Перетворення шаблону у векторне представлення та формування SQL-запиту сервером FastAPI

Серверний API, отримавши HTTP-запит, розпочинає його обробку з операції введення даних, що полягає в парсингу тіла запиту для вилучення біометричного шаблону та будь-яких супутніх метаданих, а також у повторній валідації вхідних параметрів на рівні API для забезпечення безпеки та коректності. Далі слідує критично важлива операція процесу, тобто перетворення отриманого біометричного шаблону, що може бути представлений у форматі, специфічному для сенсора FPM10A, у стандартизоване векторне представлення фіксованої довжини. Це векторне представлення є числовим вектором, що компактно кодує унікальні характеристики відбитка і є сумісним з можливостями розширення

pgvector для PostgreSQL. Для цього перетворення можуть використовуватися попередньо навчені моделі глибокого навчання або інші алгоритми векторизації біометричних ознак. Сформувавши векторне представлення, FastAPI сервер виконує наступну операцію процесу: динамічне конструювання SQL-запиту до бази даних PostgreSQL, який буде містити отриманий вектор та інструкції для пошуку за подібністю.

Критичним аспектом на цьому етапі є забезпечення повної консистентності методу векторизації: модель або алгоритм, що використовується для перетворення поточного шаблону відбитка під час ідентифікації, має бути ідентичним тому, що застосовувався при реєстрації та збереженні еталонних векторних представлень у базі даних. Будь-які розбіжності у процесі генерації ембедингів можуть призвести до суттєвого зниження точності ідентифікації. Окрім того, слід враховувати потенційну обчислювальну складність самого процесу векторизації, особливо якщо для цього залучаються складні нейромережеві моделі.

Рисунок 3.18 – Взаємодія з PosgreSQL для пошуку за подібністю та підготовка до прийняття рішення

Сформований SQL-запит передається на виконання системі управління базами даних, що є операцією взаємодії з базою даних. PostgreSQL, за допомогою розширення pgvector, здійснює ефективний пошук найближчих векторів у таблиці еталонних шаблонів, використовуючи метрики відстані та оптимізовані індексні структури (HNSW або IVFFlat) для прискорення процесу. Для забезпечення ефективного використання ресурсів та уникнення надлишкового навантаження на сервер бази даних, особливо при обробці множинних одночасних запитів, доцільно використовувати механізми пулінгу з'єднань (connection pooling). Це дозволяє повторно використовувати вже встановлені з'єднання з PostgreSQL, замість створення нового для кожного запиту. Хоча операція ідентифікації зазвичай є операцією читання (SELECT), при реалізації функціоналу реєстрації нових користувачів, що включає запис (INSERT) нового векторного шаблону та пов'язаних метаданих, важливо забезпечити атомарність цих операцій через використання транзакцій. Це гарантує, що всі зміни будуть або успішно застосовані, або повністю відхилені у випадку помилки, підтримуючи цілісність бази даних біометричних шаблонів.

Результатом виконання запиту, який повертається базою даних у вигляді операції виведення даних, зазвичай є набір кандидатів, що включає ідентифікатори потенційно відповідних користувачів та обчислені значення відстані/подібності їхніх еталонних шаблонів до запитаного вектора. Серверний API (FastAPI) отримує ці результати та виконує операцію процесу їх аналізу. Ключовим елементом цього аналізу є блок прийняття рішенн, де мінімальна обчислена відстань (або максимальна подібність) порівнюється з наперед встановленим пороговим значенням (threshold). Це порогове значення визначає чутливість системи та баланс між помилками першого та другого роду (FRR/FAR).

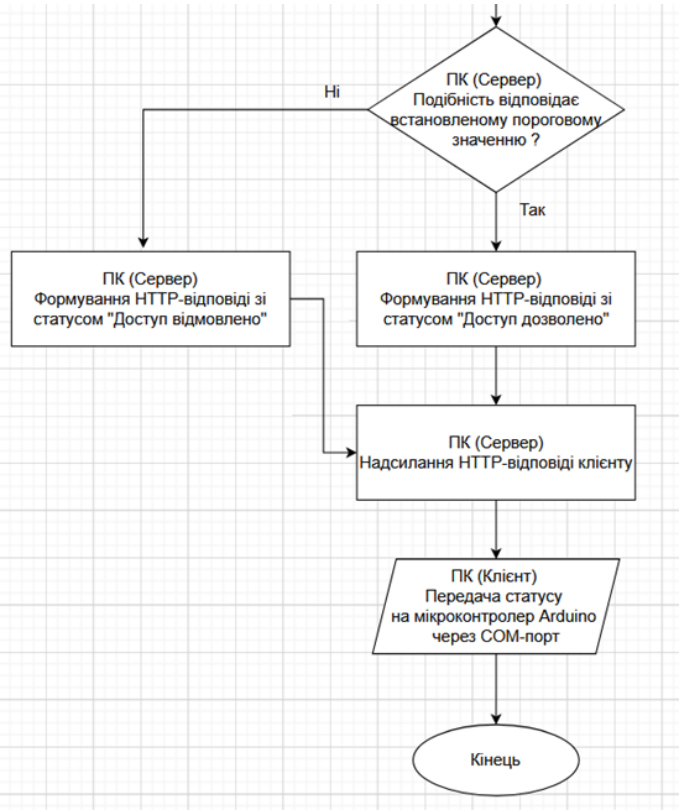


Рисунок 3.19 – Формування та передача результату ідентифікації на мікроконтролер Arduino

Якщо результат порівняння у блоці прийняття рішення свідчить про валідний збіг (гілка “Так”, наприклад, відстань менша за поріг, серверний API виконує операцію процесу, формуючи HTTP-відповідь зі статусом “Доступ дозволено” та, можливо, ідентифікатором розпізнаного користувача. Якщо ж валідного збігу не знайдено (гілка “Ні”), формується аналогічна операція процесу, але зі статусом “Доступ відмовлено”. Незалежно від результату, сформована HTTP-відповідь, зазвичай у форматі JSON, надсилається FastAPI сервером назад клієнту, що є операцією процесу. Клієнт, отримавши відповідь, вилучає з неї релевантну інформацію (статус доступу) і здійснює операцію процесу, тобто передачу відповідної команди або коду статусу на мікроконтролер Arduino через COM-порт. Після цього даний цикл серверної ідентифікації завершується, переходячи у кінцевий стан, а мікроконтролер вже відображає фінальний результат користувачеві.

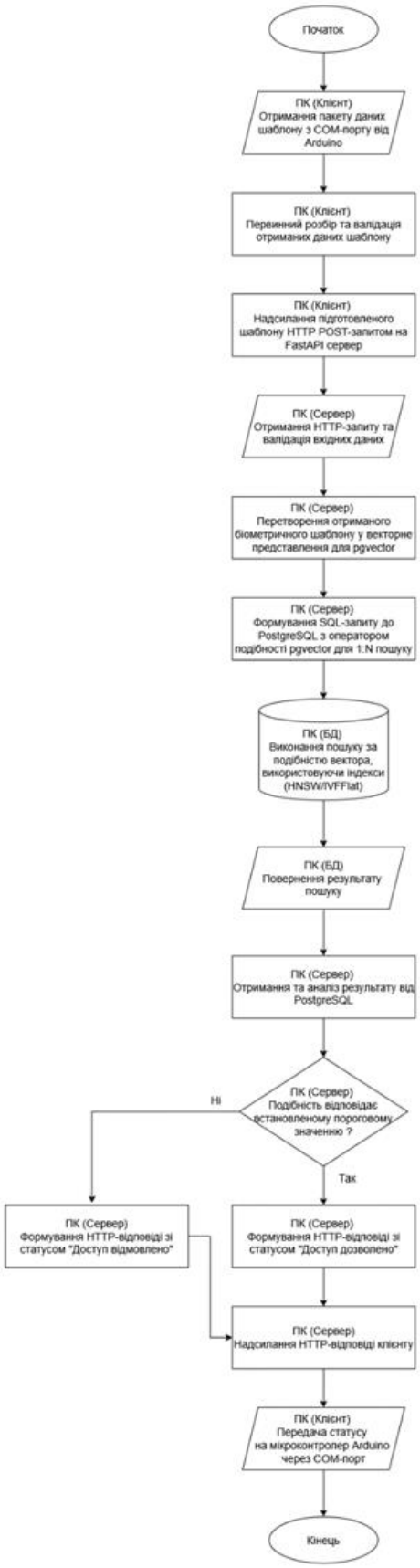


Рисунок 3.20 – Блок-схема алгоритму ідентифікації користувача на серверному компоненті системи

3.3 Програмна реалізація системи на базі мікроконтролера Arduino ATMega328

Перехід від теоретичного проєктування до практичного втілення системи автоматичного контролю доступу знаменується етапом програмної реалізації. Цей підрозділ присвячений докладному опису розробки програмного забезпечення для ключових компонентів системи: мікроконтролерного вузла на платформі Arduino ATMega328 та серверної частини, що функціонує на персональному комп'ютері. Для мікроконтролера Arduino використовується мова C++ в середовищі Arduino IDE, що дозволяє ефективно взаємодіяти з апаратною частиною. Серверна логіка, включаючи обробку запитів та взаємодію з базою даних PostgreSQL, реалізується засобами мови Python з використанням фреймворку FastAPI. Особлива увага приділяється забезпеченню коректної передачі даних між компонентами, реалізації процесів зчитування, збереження та порівняння біометричних шаблонів.

Програмне забезпечення мікроконтролера Arduino є фундаментом для взаємодії з користувачем та апаратними модулями на фізичному рівні. Воно відповідає за ініціалізацію периферії, управління процесом сканування відбитка пальця, обробку команд від сенсора, передачу біометричних даних на серверний компонент та відображення результатів ідентифікації на LCD-дисплеї, з’єднання з яким деталізовано в підрозділі 3.1.

Процес програмування починається з конфігурації середовища Arduino IDE та підключення необхідних бібліотек для роботи з сенсором FPM10A та I2C LCD-дисплеєм. У функції setup() відбувається ініціалізація послідовних портів: апаратного Serial для зв'язку з ПК та програмного SoftwareSerial для комунікації з модулем FPM10A. Також ініціалізується LCD-дисплей. Важливим кроком є перевірка зв'язку з сенсором відбитків пальців.

Основна логіка роботи мікроконтролера реалізується у функції loop(). Цей цикл починається з виведення на LCD-дисплей запрошення користувачеві

прикласти палець. Концептуально, очікування та зчитування відбитка можна представити таким псевдокодом:

```
ПРОЦЕДУРА ОтриматиЗображенняВідбитка():
  ДОВГО_ЧЕКАТИ_ПОКИ (FPM10A.отриматиЗображення() != КОД_УСПІХУ_FPM):
    ЗАТРИМКА(50)
  КІНЕЦЬ_ЦИКЛУ
  ЯКЩО результат_отримання_зображення == КОД_УСПІХУ_FPM ТОДІ
    ПОВІДОМИТИ_LCD("Зображення отримано")
    ПОВЕРНУТИ УСПІХ
  ІНАКШЕ
    ПОВІДОМИТИ_LCD("Помилка зчитування")
    ПОВЕРНУТИ ПОМИЛКА
  КІНЕЦЬ_ЯКЩО
КІНЕЦЬ_ПРОЦЕДУРИ
```

Після успішного отримання зображення, мікроконтролер надсилає команду сенсору FPM10A для його перетворення на цифровий шаблон (характерний файл) та збереження у внутрішньому буфері сенсора (наприклад, CharBuffer1). Цей процес, як зазначено в алгоритмах підрозділу 3.2, виконується власним DSP сенсора. Варто зазначити, що комунікація з модулем FPM10A зазвичай відбувається за визначеним протоколом, де кожна команда (наприклад, захоплення зображення, генерація характерного файлу, зчитування шаблону) надсилається у вигляді структурованого пакета байтів. Такий пакет, як правило, містить стартові байти, адресу сенсора, ідентифікатор команди, довжину пакета даних, самі дані (якщо є) та контрольну суму для перевірки цілісності передачі. У відповідь сенсор також надсилає пакет, що містить код підтвердження виконання команди або код помилки. Програмна реалізація на Arduino повинна коректно формувати ці командні пакети та ретельно аналізувати пакети-відповіді, обробляючи різноманітні сценарії, такі як відсутність пальця на сканері, неможливість зчитати

якісне зображення, помилка генерації або переповнення внутрішніх буферів сенсора. Наприклад, псевдокод для команди конвертації зображення в шаблон може концептуально включати очікування підтвердження від сенсора:

```
РЕЗУЛЬТАТ_КОНВЕРТАЦІЇ = FPM10A.конвертуватиЗображенняВШаблон(БУФЕР_1)
ЯКЩО РЕЗУЛЬТАТ_КОНВЕРТАЦІЇ == КОД_УСПІХУ_FPM ТОДІ
  ПОВІДОМИТИ_LCD("Шаблон створено")
ІНАКШЕ
  ПОВІДОМИТИ_LCD("Помилка конвертації")
КІНЕЦЬ_ЯКЩО
```

Коли шаблон успішно створений та збережений у буфері сенсора, мікроконтролер ініціює його вивантаження для подальшої передачі. Дані шаблону, що є масивом байтів, приймаються та тимчасово зберігаються в оперативній пам'яті Arduino. Потім цей масив байтів надсилається на персональний комп'ютер через апаратний Serial порт. Для ідентифікації початку та кінця передачі даних шаблону можуть використовуватися спеціальні маркери або попереднє узгодження довжини пакета.

Для забезпечення надійності передачі масиву байтів, що становить біометричний шаблон, на персональний комп'ютер через апаратний Serial порт, доцільно реалізувати простий, але ефективний протокол обміну. Замість прямої потокової передачі, яка може бути вразливою до помилок синхронізації або втрати даних, можна використовувати пакетний підхід. Такий підхід може включати надсилання мікроконтролером спеціальної послідовності стартових байтів, що сигналізують початок передачі шаблону, за якою слідує передача байтів самого шаблону, і завершується все це надсиланням контрольної суми (наприклад, простої суми всіх байтів шаблону по модулю 256 або більш складного CRC) та/або спеціальної послідовності стопових байтів. На стороні ПК приймаючий скрипт тоді зможе чітко ідентифікувати межі пакета, перевірити його цілісність за допомогою контрольної суми і лише після цього приступати до обробки отриманого шаблону.

Це значно підвищує стійкість системи до можливих перешкод на лінії зв'язку або помилок передачі даних.

Після відправлення шаблону на ПК, програма мікроконтролера переходить в режим очікування відповіді. Відповідь від сервера, що містить результат ідентифікації (наприклад, рядок "ДОСТУП_ДОЗВОЛЕНО" або "ДОСТУП_ВІДМОВЛЕНО"), зчитується з Serial порту. Отриманий результат аналізується, і відповідне повідомлення виводиться на LCD-дисплей, інформуючи користувача.

Серверний компонент, розгорнутий на персональному комп'ютері, відповідає за отримання біометричних шаблонів від мікроконтролера, їх обробку, взаємодію з базою даних PostgreSQL для порівняння та збереження, а також за відправку результату ідентифікації назад на мікроконтролер. Він складається з Python-скрипта, що виконує роль клієнта до мікроконтролера та API, і самого серверного API, реалізованого на FastAPI.

Клієнтський застосунок слугує проміжною ланкою. Він використовує бібліотеку pyserial для встановлення зв'язку з Arduino через COM-порт та читання даних (біометричного шаблону), що надходять. Після отримання та можливої попередньої обробки (наприклад, перевірки цілісності пакета), ці дані за допомогою бібліотеки requests надсилаються у вигляді HTTP POST-запиту на відповідний ендпоінт FastAPI сервера.

```
ПРОЦЕДУРА ГоловнийЦиклКлієнта():
  ВІДКРИТИ_SERIAL_ПОРТ_З_ARDUINO(COM_ПОРТ_X,
ШВИДКІСТЬ_Y)
  ПОСТІЙНО_ВИКОНУВАТИ:
    ШАБЛОН_ВІД_ARDUINO = ПрочитатиДаніШаблонуЗ_SerialПорту()
    ЯКЩО ШАБЛОН_ВІД_ARDUINO НЕ ПОРОЖНІЙ ТОДІ
      ДАНІ_ДЛЯ_API                                    =
ПідготуватиДаніДляЗапиту(ШАБЛОН_ВІД_ARDUINO)
```

```
        ВІДПОВІДЬ_API =
НадіслатиHTTP_POST_Запит(URL_FASTAPI_IDENTIFY, ДАНІ_ДЛЯ_API)
        ЯКЩО ВІДПОВІДЬ_API УСПІШНА ТОДІ
         РЕЗУЛЬТАТ_ІДЕНТИФІКАЦІЇ =
РозібратиВідповідьAPI(ВІДПОВІДЬ_API)

НадіслатиРезультатНаArduinoЧерезSerial(РЕЗУЛЬТАТ_ІДЕНТИФІКАЦІЇ)
        ІНАКШЕ
         НадіслатиПомилкуНаArduinoЧерезSerial("ПОМИЛКА_API")
        КІНЕЦЬ_ЯКЩО
       КІНЕЦЬ_ЯКЩО
     КІНЕЦЬ_ПОСТІЙНО_ВИКОНУВАТИ
    КІНЕЦЬ_ПРОЦЕДУРИ
```

Серверний API обробляє HTTP-запити від Python-клієнта. Для ендпоінта ідентифікації, отриманий шаблон (після валідації та декодування) перетворюється на векторне представлення (embedding), сумісне з pgvector. Як було зазначено в підрозділі 3.1, цей крок є важливим для ефективного пошуку за подібністю.

Процес перетворення біометричного шаблону, отриманого від сенсора FPM10A, у векторне представлення (embedding), сумісне з можливостями розширення pgvector, є ключовим для забезпечення точності та ефективності ідентифікації. Шаблон, що генерується сенсором FPM10A, зазвичай є набором мінуцій або іншим специфічним для виробника поданням характерних ознак відбитка. Для використання з pgvector, який оптимізований для роботи з щільними векторами фіксованої розмірності, цей вихідний шаблон потребує трансформації. Ця трансформація може бути реалізована за допомогою різних підходів. Одним із варіантів є використання попередньо навчених моделей глибокого навчання (наприклад, згорткових нейронних мереж), які спеціально тренувалися для генерації векторних ембедингів з зображень відбитків пальців або їхніх характерних файлів. Такі моделі здатні проектувати вхідні дані у багатовимірний

векторний простір, де семантично схожі відбитки матимуть близьке розташування. Альтернативно, можуть застосовуватися більш класичні методи виділення ознак та їх кодування у векторну форму. Важливо, щоб обраний метод векторизації забезпечував високу розрізнювальну здатність, тобто генерував суттєво відмінні вектори для відбитків різних осіб і схожі вектори для різних зразків відбитка однієї особи. Функція ФункціяПеретворенняШаблонуВ_Embedding у псевдокоді якраз інкапсулює цю складну логіку, яка може вимагати інтеграції зі сторонніми бібліотеками машинного навчання або реалізації власних алгоритмів обробки.

```
    ФУНКЦІЯ
ОбробникЗапитуІдентифікації(ОТРИМАНИЙ_ШАБЛОН_BASE64):
        БАЙТИ_ШАБЛОНУ                                    =
ДекодуватиBase64(ОТРИМАНИЙ_ШАБЛОН_BASE64)
        ВЕКТОР_ЗАПИТУ                                    =
ФункціяПеретворенняШаблонуВ_Embedding(БАЙТИ_ШАБЛОНУ)
        РЕЗУЛЬТАТ_З_БД = ВиконатиЗапитДоPostgreSQL_Pgvector(
                    "SELECT user_id, embedding_vector <-> ВЕКТОР_ЗАПИТУ AS
distance
                     FROM fingerprints
                     ORDER BY distance LIMIT 1"
                )

        ЯКЩО РЕЗУЛЬТАТ_З_БД ІСНУЄ ТА РЕЗУЛЬТАТ_З_БД.distance <
ПОРІГ_ПОДІБНОСТІ ТОДІ
          СТАТУС = "ДОСТУП_ДОЗВОЛЕНО"
          USER_ID = РЕЗУЛЬТАТ_З_БД.user_id
          ПОВЕРНУТИ JSON_ВІДПОВІДЬ(статус=СТАТУС, user_id=USER_ID)
        ІНАКШЕ
          СТАТУС = "ДОСТУП_ВІДМОВЛЕНО"
          ПОВЕРНУТИ JSON_ВІДПОВІДЬ(статус=СТАТУС)
```

КІНЕЦЬ_ЯКЩО

КІНЕЦЬ_ФУНКЦІЇ

Процес збереження відбитків пальців у базу даних (реєстрація) відбувається за схожою логікою: шаблон отримується від мікроконтролера, передається на серверний API (наприклад, через окремий ендпоінт /register), перетворюється у векторне представлення і зберігається в базі даних PostgreSQL разом з ідентифікатором користувача за допомогою SQL-команди INSERT.

Порівняння зчитаного відбитка з даними в базі здійснюється на сервері. Після перетворення поточного відбитка у векторне представлення, pgvector виконує пошук найближчого вектора (або векторів) серед збережених у базі даних. Для цього використовуються спеціалізовані оператори, що обчислюють метрику відстані (наприклад, евклідову <-> або косинусну <=>). Ефективність пошуку забезпечується завдяки використанню індексів типу HNSW або IVFFlat. Отримане значення відстані (або подібності) порівнюється з наперед встановленим пороговим значенням. Якщо відстань менша (або подібність більша) за поріг, вважається, що знайдено збіг, і доступ дозволяється. В іншому випадку доступ забороняється.

Представлена програмна реалізація послідовно втілює алгоритмічну логіку, описану в підрозділі 3.2 та візуалізовану на відповідних блок-схемах (Рисунки 3.10, 3.15). Мікроконтролерний вузол ефективно виконує функції збору та первинної передачі біометричних даних. Серверний компонент, завдяки потужностям Python, FastAPI та PostgreSQL з pgvector, забезпечує складні операції обробки, зберігання та швидкого пошуку за подібністю, що є критичним для надійної ідентифікації. Взаємодія між компонентами через послідовний порт та HTTP-протокол дозволяє створити гнучку та функціональну систему контролю доступу.

3.4 Висновок

У третьому розділі кваліфікаційної роботи було здійснено комплексне проєктування та детально описано етапи програмної реалізації автоматизованої системи контролю доступу на базі мікроконтролера Arduino ATMega328 та біометричної ідентифікації за відбитками пальців. Було розроблено та обгрунтовано як апаратну конфігурацію, так і програмну архітектуру системи, що забезпечує виконання поставлених завдань.

На основі аналізу, проведеного в попередніх розділах, була представлена детальна електрична принципова схема підключення ключових апаратних компонентів, включаючи мікроконтролер Arduino Uno R3, оптичний сенсор відбитків пальців FPM10A та рідкокристалічний дисплей LCD 1602. Також було поглиблено описано програмно-апаратну архітектуру системи, що базується на клієнт-серверній моделі. В рамках цієї моделі мікроконтролерний вузол виконує функції збору даних та взаємодії з користувачем, а персональний комп'ютер виступає в ролі сервера, відповідального за обробку даних та управління базою біометричних шаблонів PostgreSQL з розширенням pgvector. Було деталізовано потоки даних та протоколи взаємодії між цими компонентами.

Важливим етапом стала розробка та формалізація алгоритмів роботи системи. Було представлено загальну блок-схему алгоритму ідентифікації користувача, що охоплює всі етапи від сканування відбитка до відображення результату. Крім того, було детально описано та візуалізовано за допомогою блок-схем ключові підпроцеси, зокрема алгоритм зчитування та первинної обробки біометричного шаблону відбитка пальця мікроконтролерним вузлом, що включає взаємодію з сенсором FPM10A та обробку його відповідей.

Далі було докладно розглянуто аспекти програмної реалізації системи. Для мікроконтролерного вузла описано процес програмування в середовищі Arduino IDE мовою C++, включаючи ініціалізацію периферійних пристроїв, логіку взаємодії з сенсором FPM10A для захоплення та отримання шаблону відбитка,

організацію обміну даними з персональним комп'ютером через послідовний порт, та виведення інформації на LCD-дисплей. Для серверної частини було описано реалізацію клієнтського Python-застосунку, що забезпечує комунікацію з мікроконтролером та виступає посередником для взаємодії з серверним API, а також розробку самого серверного API на базі FastAPI. Було пояснено процес збереження біометричних шаблонів (у вигляді векторних представлень) у базі даних PostgreSQL та механізм їх порівняння під час ідентифікації з використанням можливостей розширення pgvector, включаючи формування запитів та аналіз результатів для прийняття рішення про доступ. Основні аспекти програмної логіки були проілюстровані відповідними фрагментами псевдокоду.

Таким чином, у третьому розділі було закладено детальну проектну та програмну основу для створення функціонального прототипу системи біометричного контролю доступу.

ВИСНОВОК

У ході виконання кваліфікаційної роботи було проведено комплексне дослідження та розробку системи автоматичного контролю доступу, що використовує біометричні дані відбитків пальців для ідентифікації користувачів. На початковому етапі було здійснено глибокий аналіз сучасних систем контролю доступу та існуючих біометричних технологій, виявлено їхні переваги, недоліки та сфери застосування. Особливу увагу було приділено методам ідентифікації за відбитками пальців та обгрунтовано актуальність створення доступного та надійного рішення на базі мікроконтролерної техніки.

На основі проведеного аналізу було сформульовано основну мету та завдання дослідження, що полягали у розробці програмно-апаратного комплексу, здатного ефективно виконувати функції біометричної ідентифікації. Було здійснено обгрунтований вибір ключових апаратних компонентів системи, зокрема мікроконтролерної платформи Arduino Uno R3 як центрального керуючого вузла, оптичного сенсора відбитків пальців FPM10A для збору біометричних даних та рідкокристалічного дисплея для взаємодії з користувачем. Важливим аспектом стало проєктування архітектури системи, яка була реалізована за клієнт-серверною моделлю. Ця модель передбачає взаємодію мікроконтролерного клієнта, відповідального за збір даних та управління периферією, з серверним компонентом на базі персонального комп’ютера, на якому розгорнуто базу даних PostgreSQL з розширенням pgvector для надійного зберігання та ефективного порівняння біометричних шаблонів.

Для забезпечення коректного функціонування системи було розроблено детальну електричну принципову схему з’єднання всіх апаратних модулів. Функціональна логіка системи була формалізована у вигляді алгоритмів, представлених блок-схемами, що детально описують процес ініціалізації, сканування відбитка пальця, передачі даних, серверної обробки, взаємодії з базою

даних для ідентифікації один-до-багатьох, та відображення кінцевого результату користувачеві.

Завершальним етапом теоретичного проєктування став детальний опис програмної реалізації як для мікроконтролерного вузла Arduino, так і для серверної частини на ПК. Було розглянуто процес програмування мікроконтролера мовою C++ в середовищі Arduino IDE, включаючи взаємодію з сенсором та дисплеєм, а також організацію обміну даними з сервером. Для серверної частини описано реалізацію Python-застосунку, що виступає посередником, та розробку API на базі FastAPI для обробки запитів, перетворення шаблонів у векторні представлення та виконання операцій порівняння в базі даних PostgreSQL з використанням pgvector. Ключові аспекти програмної логіки були проілюстровані відповідними фрагментами псевдокоду.

Таким чином, виконана робота охопила всі етапи від аналізу предметної області та постановки задачі до детального проєктування апаратної конфігурації, розробки алгоритмів та планування програмної реалізації системи біометричного контролю доступу. Створено комплексну теоретичну та проектну базу, що слугує міцним підгрунтям для подальшої практичної реалізації прототипу системи, його тестування та оцінки ефективності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Bergsma J., Alruwaili M., Alzahrani B. A Systematic Survey of Multi-Factor Authentication for Cloud Infrastructure. *Future Internet*. 2023. Vol. 15, no. 4. URL: https://www.mdpi.com/1999-5903/15/4/146 (дата звернення: 26.05.2025).

2. Міщенко Д. В. Біометричні технології: кордони, контроль доступу, FinTech та інші кейси. *Системи і Бізнес*. 2025. № 1 (135). URL: http://sib.com.ua/sib-1-135-2025/biometriya.html (дата звернення: 26.05.2025).

3. Sharma N., Kumar P. *Access control system using Arduino microcontroller and RFID reader*. 2024. URL: https://www.researchgate.net/publication/377787774_Access_control_system_using_Arduino_microcontroller_and_RFID_reader (дата звернення: 26.05.2025).

4. Mindscope. *Ефективність біометричних технологій в системах контролю доступу*. URL: https://mindscope.biz.ua/efektyvnist-biometrychnyh-tehnologij-v-systemah-kontrolyu-dostupu/ (дата звернення: 26.05.2025).

5. СІБ. Біометрія: сучасні технології автентифікації. *Східноукраїнський науковий вісник*. 2025. № 1(135). URL: http://sib.com.ua/sib-1-135-2025/biometriya.html (дата звернення: 26.05.2025).

6. Pref-Tech. *Facial and Iris Recognition in Access Control Systems*. URL: https://pref-tech.com/blog-post-7/ (дата звернення: 26.05.2025).

7. BCS Consultants. *Top 8 Access Control Technology Trends for 2025*. URL: https://www.bcsconsultants.com/blog/top-8-access-control-technology-trends-for-2025/ (дата звернення: 26.05.2025).

8. Mukhopadhyay S. C. Access Control Systems and Biometrics. *Proceedings of the 6th International Conference on Intelligent Systems and Applications (INTELLIAPP 2023)* / ed. by Y. Maleh, E. Hanafi, A. Haqiq, M. Belaissaoui, M. Essaaidi. Cham : Springer Nature Switzerland, 2024. P. 179–180. (Lecture Notes in Networks and Systems ; vol. 871). DOI: https://doi.org/10.1007/978-3-031-48879-5_13.

9. Koudogbo A., Adechinan A., Finaourou M., Lokossou T. V. Implementation of Microcontroller-Based Home Security System Using TF-Luna LiDAR. *International*

Journal of Engineering, Management and Technology. 2024. Vol. 5, no. 1. URL: https://iiardjournals.org/abstract.php?j=IJEMT&pn=Implementation+of+Microcontroller-Based+Home+Security+System+%0AUsing +TF-Luna+LiDAR+%0A&id=4477 (дата звернення: 26.05.2025).

10. Kanagamalliga S., Rajalingam S., Karthikeyan M., Kannan A. Arduino-Powered Fingerprint Authentication for Door Access Control. *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC) : proceedings of the conference, 22-24 Feb. 2024*. Piscataway, NJ : IEEE, 2024. P. 131–135. DOI: https://ieeexplore.ieee.org/abstract/document/10689965.

11. Ghantasala H., Kendyala R. K., Kumar M. K., Bhargavi K., Ahammed S. F. RFID based Access Control System. *2025 International Conference on Electronics and Renewable Systems (ICEARS) : proceedings of the conference, 13-15 March 2025*. Piscataway, NJ : IEEE, 2025. P. 1-5. URL: https://ieeexplore.ieee.org/abstract/document/10941511 (дата звернення: 26.05.2025).

12. ASD USA. *The Importance of Access Control and Surveillance System Integration for Enterprise Security*. URL: https://www.asd-usa.com/blog/access-control-surveillance-integration (дата звернення: 26.05.2025).

13. Kyivstar Hub. *Останні тренди кібербезпеки*. URL: https://hub.kyivstar.ua/articles/ostanni-trendy-kiberbezpeky (дата звернення: 26.05.2025).

14. Рада національної безпеки і оборони України. *Проєкт Стратегії кібербезпеки України*. 2021. URL: https://www.rnbo.gov.ua/files/2021/STRATEGIYA%20KYBERBEZPEKI/proekt%20strategii_kyberbezpeki_Ukr.pdf (дата звернення: 26.05.2025).

15. CISA. *Catalog of Known Exploited Vulnerabilities*. URL: https://www.cisa.gov/known-exploited-vulnerabilities-catalog (дата звернення: 26.05.2025).

16. University of San Diego. *Top Cybersecurity Threats*. URL: https://onlinedegrees.sandiego.edu/top-cyber-security-threats/ (дата звернення: 26.05.2025).

17. Alsharif M. H., Kuru K., Alzahrani M. Y., Ahmed M. A., Alabdullah N. A., Alhartomi M. A. Cybersecurity in Smart Cities: Challenges, Solutions, and Future Research Directions. *Sensors*. 2023. Vol. 23, no. 18. Art. 8014. DOI: https://doi.org/10.3390/s23188014.

18. Alsabah M., Alqallaf A., AlFaylakawi A., Alhubail A., Hawwar Y. A Comprehensive Review on Cybersecurity and Privacy Issues in Smart Grid Systems. *Energies*. 2025. Vol. 18, no. 8. Art. 1973. DOI: https://doi.org/10.3390/en18081973.

19. Babatunde D., Maskeliunas R., Damaševičius R., Ogundokun R. O. *Biometric Systems: A Comprehensive Review*. Preprint. April 2024. URL: https://www.researchgate.net/publication/381148983_Biometric_Systems_A_Comprehensive_Review (дата звернення: 26.05.2025).

20. Jain A. K., Ross A., Prabhakar S. Biometrics: A Tool for Information Security. *IEEE Transactions on Information Forensics and Security*. 2006. Vol. 1, no. 2. P. 125–143. URL: https://www.researchgate.net/publication/3455239_Biometrics_a_tool_for_information_security_IEEE_Tran_Inform_Forensics_Secur (дата звернення: 26.05.2025).

21. Gram-news.com.ua. *Що таке інформаційна безпека?* 2023. URL: https://gram-news.com.ua/shho-take-informatsijna-bezpeka/ (дата звернення: 26.05.2025).

22. Mogahed M., Abdelrahman M., Alabady S., El-Gayar M. *Biometric Systems: A Comprehensive Review*. ResearchGate. April 2024. URL: https://www.researchgate.net/publication/381148983_Biometric_Systems_A_Comprehensive_Review (дата звернення: 26.05.2025).

23. Recogtech. *FAR and FRR: Security Level Versus Ease of Use*. 2022. URL: https://recogtech.com/en/insights-en/far-and-frr-security-level-versus-ease-of-use/ (дата звернення: 26.05.2025).

24. Infocom. *Контроль доступу. Біометрія в дії*. URL: https://infocom.ua/%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C-%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D1%83-%D0%B1%D1%96%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D1%96%D1%8F-%D0%B2-%D0%B4%D1%96%D1%97/ (дата звернення: 26.05.2025).

25. HF Security. *Optical Fingerprint Scanner vs. Ultrasonic Fingerprint Scanner*. 2024. URL: https://hfsecurity.cn/optical-fingerprint-scanner-vs-ultrasonic-fingerprint-scanner/ (дата звернення: 26.05.2025).

26. Sims G. *How Fingerprint Scanners Work*. Android Authority. 2023. URL: https://www.androidauthority.com/how-fingerprint-scanners-work-670934/ (дата звернення: 26.05.2025).

27. Aratek. *The 4 Fingerprint Sensor Types*. 2023. URL: https://www.aratek.co/news/the-4-fingerprint-sensor-types (дата звернення: 26.05.2025).

28. Irfan M., Ramzan M., Shaikh S., Butt B. Z. Biometric Template Protection: A Systematic Literature Review and Future Research Directions. *International Conference on Cyber Warfare and Security (ICCWS)*. 2023. Vol. 18, no. 1. P. 481–490. URL: https://www.researchgate.net/publication/367368491_Biometric_Template_Protection_A_Systematic_Literature_Review_and_Future_Research_Directions (дата звернення: 26.05.2025).

29. Batarseh M., Suter L. Biometric Data Security: Challenges and Protection Techniques. *Croatian Journal of Electrical Engineering and Computer Science*. 2020. Vol. 18, no. 2. P. 1–9. URL: https://hrcak.srce.hr/file/333669 (дата звернення: 26.05.2025).

30. Council of Europe. *Opinion on the draft Law of Ukraine on Personal Data Protection*. 2023. URL: https://rm.coe.int/ua-opinion-on-the-draft-law-of-ukraine-on-personal-data-protection-/1680ad38bf (дата звернення: 26.05.2025).

31. National Institute of Standards and Technology. *NIST and Biometric Standards*. URL: https://www.nist.gov/biometrics (дата звернення: 26.05.2025).

32. Exabeam. *GDPR Article 9: Special Personal Data Categories and How to Protect Them*. URL: https://www.exabeam.com/explainers/gdpr-compliance/gdpr-article-9-special-personal-data-categories-and-how-to-protect-them/ (дата звернення: 26.05.2025).

33. Onyshchenko S., Burbii A., Boikov A., Riabiy S., Korniiko S. Personal data protection on the internet under martial law: The case of Ukraine. *Amazonia Investiga*. 2023. Vol. 12, no. 69. P. 204–215. DOI: https://doi.org/10.34069/AI/2023.69.09.18.

34. Bioenable Technologies. *Retina Scanner Biometric Device*. URL: https://www.bioenabletech.com/retina-scanner-biometric-device (дата звернення: 26.05.2025).

35. Ali N. S., Alhilali A. H., Alrikabi H. Th. S., Alsharqi H., Al-Sadawi B. Automated attendance management systems: systematic literature review. *International Journal of Technology Enhanced Learning*. 2022. Vol. 14, no. 1. P. 37–65. DOI: https://doi.org/10.1504/IJTEL.2022.120559.

36. National Institute of Standards and Technology. *Biometric Standards Program and Resource Center*. URL: https://www.nist.gov/programs-projects/biometric-standards-program-and-resource-center (дата звернення: 26.05.2025).

37. Priesnitz J., Rathgeb C., Buchmann N., Busch C., Margraf M. An overview of touchless 2D fingerprint recognition. *EURASIP Journal on Image and Video Processing*. 2021. Art. 4. P. 1–28. DOI: https://doi.org/10.1186/s13640-021-00548-4.

38. TechReview. *Biometric Devices in Consumer Electronics*. URL: https://www.techreview.com/biometric-devices (дата звернення: 26.05.2025).

39. Hernandez-de-Menendez M., Morales-Menendez R., Escobar C. A., Arinez J. Biometric applications in education. *International Journal on Interactive Design and Manufacturing (IJIDeM)*. 2021. Vol. 15, no. 2-3. P. 365–380. DOI: https://doi.org/10.1007/s12008-021-00760-6.

40. Identity.com. *The Future of Biometric Data Protection*. URL: https://www.identity.com/the-future-of-biometric-data-protection/ (дата звернення: 26.05.2025).

41. Mantratec. *Encryption in Biometric Technology*. URL: https://www.mantratec.com/Encryption-in-Biometric-Technology (дата звернення: 26.05.2025).

42. Dyman O. V., Havrilenko O. A. Security and Privacy of Biometric Data. *Cybersecurity: Education, Science, Technique*. 2023. Vol. 4, no. 20. P. 103–110. DOI: https://doi.org/10.28925/2663-4023.2023.20.11.

43. Kim H. J. Biometric Technologies and Privacy Protection: A Systematic Review. *Journal of Personalized Medicine*. 2023. Vol. 13, no. 4. Art. 659. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC10098691/ (дата звернення: 26.05.2025).

44. Arduino Documentation. *Arduino UNO Rev3*. URL: https://docs.arduino.cc/hardware/uno-rev3 (дата звернення: 26.05.2025).

45. Santos R., Santos S. *Getting Started with ESP32*. Random Nerd Tutorials. 2024. URL: https://randomnerdtutorials.com/getting-started-with-esp32/ (дата звернення: 26.05.2025).

46. ARDUINO. *Arduino UNO Datasheet*. ALLDATASHEET.COM. URL: https://www.alldatasheet.com/datasheet-pdf/pdf/1943445/ARDUINO/ARDUINO-UNO.html (дата звернення: 26.05.2025).

47. Traditional_Self470. *ESP32 vs Arduino*. Reddit, r/esp32. 18 March 2024. URL: https://www.reddit.com/r/esp32/comments/1bhrf2d/esp32_vs_arduino/ (дата звернення: 26.05.2025).

48. CRYSTAIFONTZ. *LCD-1602A Datasheet*. ALLDATASHEET.COM. URL: https://www.alldatasheet.com/datasheet-pdf/pdf/1574132/CRYSTAIFONTZ/LCD-1602A.html (дата звернення: 26.05.2025).

49. HandsOn Technology. *I2C LCD Module Datasheet*. Rev 1.0. 2018. URL: https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf (дата звернення: 26.05.2025).

50. Chris W. *OLED vs LCD: Which One to Choose*. EEWorld. 2024. URL: https://en.eeworld.com.cn/news/qcdz/eic695177.html (дата звернення: 26.05.2025).

51. Digimax. *Display Technologies Overview*. URL: https://www.digimax.it/en/9-display (дата звернення: 26.05.2025).

52. *FPM10A Fingerprint User Manual*. Scribd. URL: https://www.scribd.com/document/498402401/FPM10A-Fingerprint-User-Manual (дата звернення: 26.05.2025).

53. Robotics Bangladesh. *FPM10A Optical Fingerprint Reader Sensor Module*. URL: https://store.roboticsbd.com/sensors/553-fpm10a-optical-fingerprint-reader-sensor-modules-robotics-bangladesh.html (дата звернення: 26.05.2025).

54. SDC Security. *Understanding electromagnetic locks*. URL: https://www.sdcsecurity.com/docs/whitepapers-emlocks.pdf (дата звернення: 26.05.2025).

55. ButterflyMX. *Electric locks vs magnetic locks: Which should you choose?* 2021. URL: https://butterflymx.com/blog/electric-locks-vs-magnetic-locks/ (дата звернення: 26.05.2025).

56. Kisi. *Types of electronic locks*. URL: https://www.getkisi.com/guides/electronic-locks (дата звернення: 26.05.2025).

57. YLI Electronic. *Product catalog and specifications*. URL: https://www.yli.cn/en/ (дата звернення: 26.05.2025).

58. ProIT. *Нові горизонти біометричних систем безпеки*. URL: https://proit.com.ua/news/novi-goryzonty-biometrychnyh-system-bezpeky/ (дата звернення: 26.05.2025).

59. IDEMIA. *Biometrics*. URL: https://www.idemia.com/biometrics (дата звернення: 26.05.2025).

60. Almas Industries. *How is biometric data stored?* URL: https://almas-industries.com/blog/how-is-biometric-data-stored/ (дата звернення: 26.05.2025).

61. Suprema. *General – About 1:1 and 1:N Authentication*. URL: https://support.supremainc.com/en/support/solutions/articles/24000006373--general-about-1-1-and-1-n-authentication (дата звернення: 26.05.2025).

62. MoldStud. *Implementing Advanced Security Measures in PostgreSQL*. URL: https://moldstud.com/articles/p-implementing-advanced-security-measures-in-postgresql (дата звернення: 26.05.2025).

63. HyperVerge. *The Future of Biometrics*. 2023. URL: https://hyperverge.co/blog/future-of-biometrics/ (дата звернення: 26.05.2025).

64. Idemia. *Biometric Systems Catalog*. 2024. URL: https://fortisec.com.ua/admin/wp-content/uploads/2024/02/idemia-catalog.pdf (дата звернення: 26.05.2025).

65. EnterpriseDB. *Scaling PostgreSQL for High Availability and Performance*. URL: https://www.enterprisedb.com/scaling-postgresql-high-availability-and-performance (дата звернення: 26.05.2025).

66. Almomani A., Omar K., Zaidan B. B. et al. A Secure and Efficient Biometric Authentication Framework Using Homomorphic Encryption in Cloud Environment. *Future Internet*. 2024. Vol. 16, no. 10. Art. 382. DOI: https://doi.org/10.3390/fi16100382.

67. Ziemann E. *Biometric Database Architecture for Enhanced Privacy and Security*. ODU Undergraduate Research Symposium. 2022. Paper 119. URL: https://digitalcommons.odu.edu/cgi/viewcontent.cgi?article=1118&context=covacci-undergraduateresearch (дата звернення: 26.05.2025).

68. CyberPanel. *PostgreSQL vs MySQL*. URL: https://cyberpanel.net/blog/postgres-vs-mysql (дата звернення: 26.05.2025).

69. Amazon Web Services. *The Difference Between MySQL vs PostgreSQL*. URL: https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/ (дата звернення: 26.05.2025).

70. Severalnines. *Vector Similarity Search with PostgreSQL's pgvector: A Deep Dive*. 2023. URL: https://severalnines.com/blog/vector-similarity-search-with-postgresqls-pgvector-a-deep-dive/ (дата звернення: 26.05.2025).

71. Secoda. *Data Privacy for Postgres*. URL: https://www.secoda.co/glossary/data-privacy-for-postgres (дата звернення: 26.05.2025).

72. MongoDB. *BSON – Binary JSON*. URL: https://www.mongodb.com/resources/languages/bson (дата звернення: 26.05.2025).

73. Almomani A. *MongoDB as a Flexible Platform for Biometric Identity Management*. EasyChair Preprints. 2023. No. 7MMG. URL: https://easychair.org/publications/preprint/7MMG/open (дата звернення: 26.05.2025).

74. MongoDB Documentation. *Atlas Vector Search Overview*. URL: https://www.mongodb.com/docs/atlas/atlas-vector-search/vector-search-overview/ (дата звернення: 26.05.2025).

75. AWS Blog. *Improving MongoDB Atlas Search Elasticity with Amazon S3*. 2023. URL: https://aws.amazon.com/blogs/apn/improving-mongodb-atlas-search-elasticity-with-amazon-s3/ (дата звернення: 26.05.2025)

76. Oracle. *Database Security*. URL: https://www.oracle.com/security/database-security/ (дата звернення: 26.05.2025).

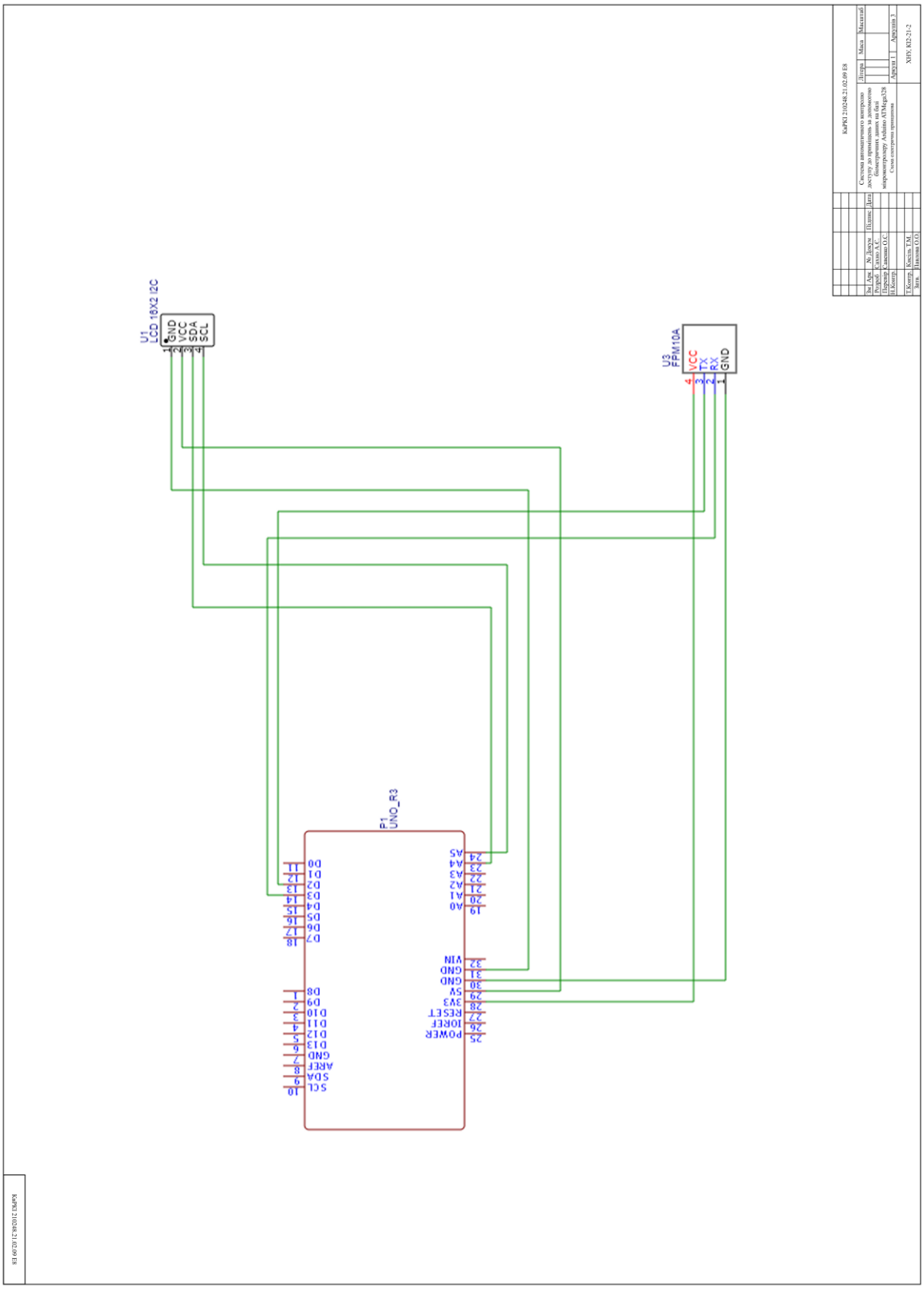
77. Oracle. *Oracle Database Features*. URL: https://www.oracle.com/database/features/ (дата звернення: 26.05.2025).

78. Ilić M., Kopanja L., Zlatković D., Trajković M., Ćurguz D. Microsoft SQL Server and Oracle: comparative performance analysis. *Proceedings of the 7th International Conference Knowledge Management and Informatics (KMI 2021)* : Vrnjačka Banja, Serbia, June 3-4, 2021. Kraljevo : University of Kragujevac, Faculty of Mechanical and Civil Engineering, 2021. P. 33–40. URL: https://www.researchgate.net/profile/Dragan-Zlatkovic/publication/352348811_MICROSOFT_SQL_SERVER_AND_ORACLE_COMPARATIVE_PERFORMANCE_ANALYSIS/links/60c43847a6fdcc2e613650dc/MICROSOFT-SQL-SERVER-AND-ORACLE-COMPARATIVE-PERFORMANCE-ANALYSIS.pdf (дата звернення: 27.05.2025).

ДОДАТОК А

(обов’язковий)

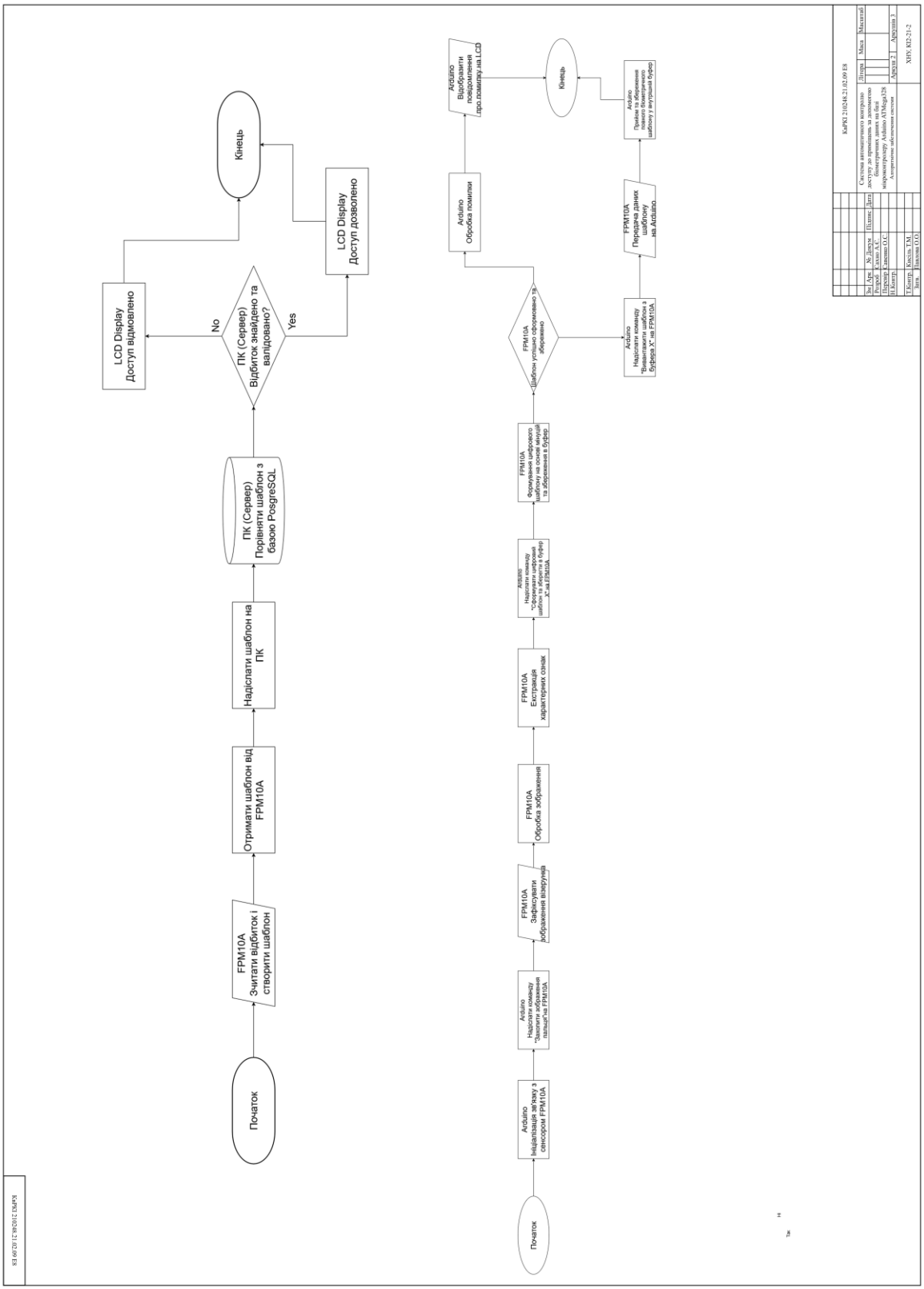
КОПІЯ КРЕСЛЕННЯ “СХЕМА ЕЛЕКТРИЧНА ПРИНЦИПОВА”



ДОДАТОК Б

(обов’язковий)

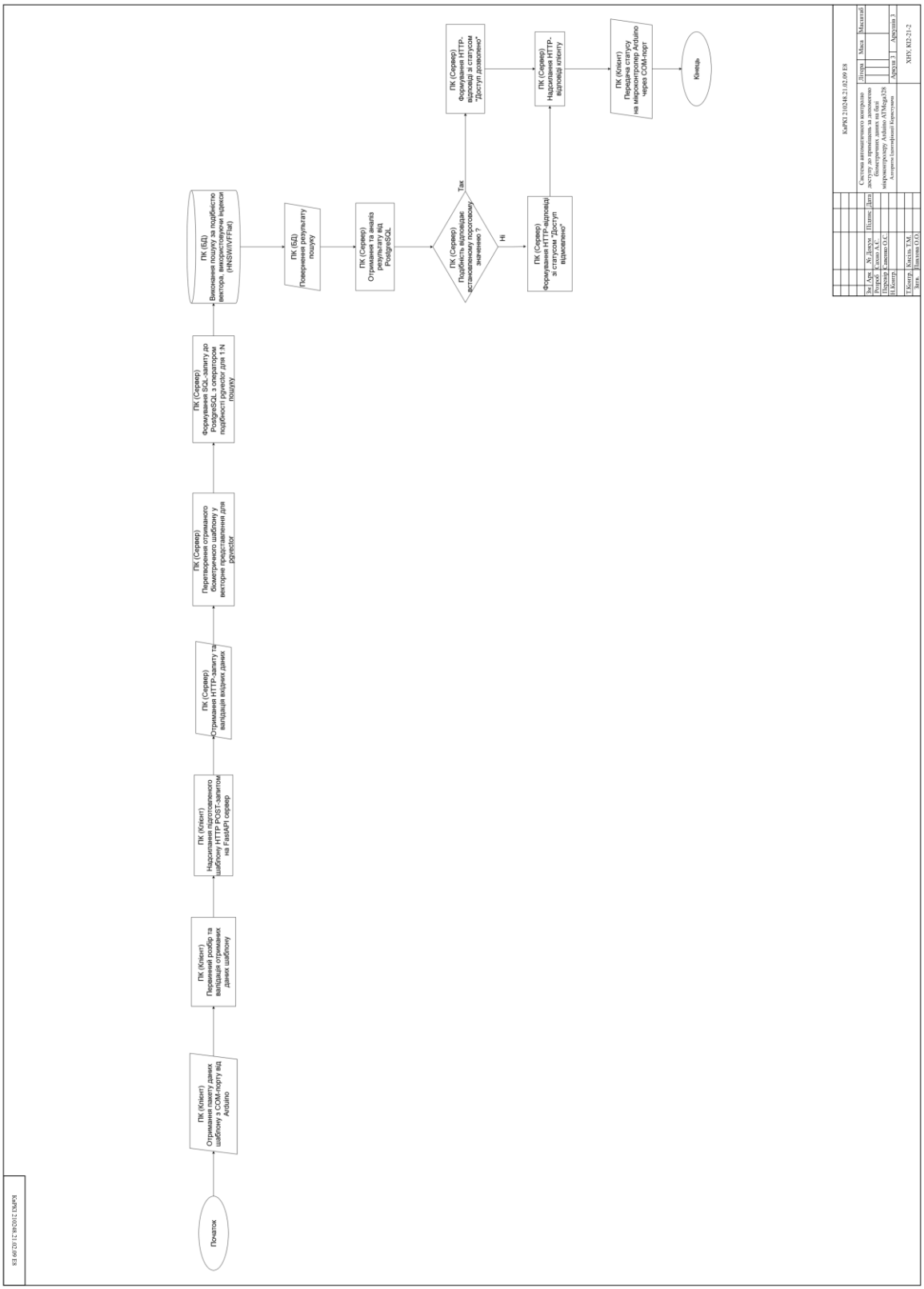
КОПІЯ КРЕСЛЕННЯ “АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ”



ДОДАТОК В

(обов’язковий)

КОПІЯ КРЕСЛЕННЯ “АЛГОРИТМ ІДЕНТИФІКАЦІЇ КОРИСТУВАЧА”



ДОДАТОК Г

(обов’язковий)

ВИХІДНИЙ КОД ДЛЯ МІКРОКОНТРОЛЕРА І СЕРВЕРНОЇ ЧАСТИНИ

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
SoftwareSerial FPM_Serial(A2, A3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&FPM_Serial);
LiquidCrystal_I2C lcd(0x27, 16, 2);
const uint16_t FPM_TEMPLATE_SIZE = 768;
uint8_t currentTemplateBuffer[FPM_TEMPLATE_SIZE];
void displayLCD(String line1, String line2 = "") {
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(line1.substring(0, 16));
if (line2.length() > 0) {
lcd.setCursor(0, 1);
lcd.print(line2.substring(0, 16));
}
}
int getFingerprintTemplateAndDownload() {
displayLCD("Place finger", "Scanning...");
uint8_t p = finger.getImage();
unsigned long startTime = millis();
while (p != FINGERPRINT_OK && (millis() - startTime < 5000)) {
p = finger.getImage();
delay(50);
}
if (p != FINGERPRINT_OK) {
displayLCD("Timeout!", "No finger?");
delay(2000);
return -1;
```

```
}
displayLCD("Image taken", "Processing...");
p = finger.image2Tz(1);
if (p != FINGERPRINT_OK) {
displayLCD("Conversion err", "");
delay(2000);
return -2;
}
displayLCD("Template ready", "Downloading...");
p = finger.getModel();
if (p != FINGERPRINT_OK) {
displayLCD("Download cmd err", "");
delay(2000);
return -3;
}
uint16_t bytesRead = 0;
startTime = millis();
for (int i=0; i < FPM_TEMPLATE_SIZE; i++) {
currentTemplateBuffer[i] = 0;
}
unsigned long packetStartTime = millis();
while(bytesRead < FPM_TEMPLATE_SIZE && (millis() - packetStartTime) <
2000 ) {
if(FPM_Serial.available()){
currentTemplateBuffer[bytesRead] = FPM_Serial.read();
bytesRead++;
}
}
if (bytesRead == FPM_TEMPLATE_SIZE) {
displayLCD("Template got", String(bytesRead) + " bytes");
delay(1000);
return 0;
} else {
displayLCD("Read tmpl. err", String(bytesRead) + "b. Expected " +
String(FPM_TEMPLATE_SIZE));
```

```
delay(2000);
return -4;
}
}
void sendTemplateToPC() {
Serial.write('S');
Serial.write(currentTemplateBuffer, FPM_TEMPLATE_SIZE);
Serial.write('E');
Serial.flush();
}
void setup() {
Serial.begin(9600);
FPM_Serial.begin(57600);
lcd.init();
lcd.backlight();
lcd.clear();
displayLCD("System Booting", "Please wait...");
if (finger.verifyPassword()) {
displayLCD("Sensor Found", "System Ready");
} else {
displayLCD("Sensor Error!", "Check connection");
while (1) { delay(10); }
}
delay(2000);
lcd.clear();
}
void loop() {
displayLCD("Ready for Scan", "");
if (getFingerprintTemplateAndDownload() == 0) {
displayLCD("Sending to PC...", "");
sendTemplateToPC();
unsigned long responseStartTime = millis();
String pcResponse = "";
displayLCD("Waiting PC resp.", "");
while ((millis() - responseStartTime) < 10000) {
```

```
if (Serial.available() > 0) {
pcResponse = Serial.readStringUntil('\n');
pcResponse.trim();
break;
}
}
if (pcResponse.length() > 0) {
if (pcResponse == "ACCESS_GRANTED") {
displayLCD("ACCESS GRANTED", "");
} else if (pcResponse == "ACCESS_DENIED") {
displayLCD("ACCESS DENIED", "");
} else {
displayLCD("PC Response:", pcResponse);
}
} else {
displayLCD("No PC Response", "Timeout!");
}
}
delay(3000);
lcd.clear();
}
```

```
from fastapi import FastAPI, HTTPException, BackgroundTasks
from pydantic import BaseModel
import psycopg2
import psycopg2.extras
import base64
import uvicorn
from typing import List, Optional
DATABASE_URL =
"postgresql://pguser:student1@localhost:5432/dyplomna"
FPM_TEMPLATE_VECTOR_DIMENSION = 128
SIMILARITY_DISTANCE_THRESHOLD = 0.6

app = FastAPI(title="Fingerprint API")
```

```
class FingerprintTemplateData(BaseModel):
    template_data: str

class UserRegistrationData(BaseModel):
    user_id: str
    template_data: str

class ApiResponse(BaseModel):
    status: str
    user_id: Optional[str] = None
    message: Optional[str] = None

def get_db_connection_internal():
    try:
        conn = psycopg2.connect(DATABASE_URL)
        return conn
    except psycopg2.OperationalError as e:
        print(f"Database connection failed: {e}")
        raise HTTPException(status_code=503, detail="Database
service unavailable.")

def convert_raw_template_to_vector_embedding(raw_template_bytes:
bytes) -> List[float]:
    print(f"Placeholder: Converting template of length
{len(raw_template_bytes)} to vector.")
    if len(raw_template_bytes) < FPM_TEMPLATE_VECTOR_DIMENSION:
        raise ValueError("Template data too short for placeholder
embedding.")

    embedding_vector = [(float(byte_val) / 255.0) for byte_val in
raw_template_bytes[:FPM_TEMPLATE_VECTOR_DIMENSION]]
    print(f"Placeholder: Generated embedding of dimension
{len(embedding_vector)}.")
    return embedding_vector
```

```
@app.post("/identify", response_model=ApiResponse)
async def identify_fingerprint_api_endpoint(data:
FingerprintTemplateData):
    db_conn = None
    try:
        raw_template_bytes = base64.b64decode(data.template_data)
        query_vector_embedding =
convert_raw_template_to_vector_embedding(raw_template_bytes)

        db_conn = get_db_connection_internal()
        with
db_conn.cursor(cursor_factory=psycopg2.extras.RealDictCursor) as
db_cursor:
            sql_query_str = "SELECT user_id, (embedding <-> %s) AS
distance FROM fingerprints ORDER BY distance ASC LIMIT 1;"
            db_cursor.execute(sql_query_str,
(query_vector_embedding,))
            db_result = db_cursor.fetchone()

        if db_result:
            retrieved_user_id = db_result["user_id"]
            calculated_distance = db_result["distance"]
            if calculated_distance < SIMILARITY_DISTANCE_THRESHOLD:
                return ApiResponse(status="ACCESS_GRANTED",
user_id=str(retrieved_user_id))
            else:
                return ApiResponse(status="ACCESS_DENIED",
message="Low similarity score.")
        else:
            return ApiResponse(status="ACCESS_DENIED",
message="Fingerprint not found in database.")

    except ValueError as val_err:
```

```
        raise HTTPException(status_code=400, detail=f"Invalid
template data: {val_err}")
    except psycopg2.Error as db_err:
        raise HTTPException(status_code=500, detail=f"Database query
error: {db_err}")
    except Exception as exc:
        raise HTTPException(status_code=500, detail=f"Internal
server error: {exc}")
    finally:
        if db_conn:
            db_conn.close()

@app.post("/register", response_model=ApiResponse)
async def register_fingerprint_api_endpoint(data:
UserRegistrationData):
    db_conn = None
    try:
        raw_template_bytes = base64.b64decode(data.template_data)
        embedding_to_store =
convert_raw_template_to_vector_embedding(raw_template_bytes)

        db_conn = get_db_connection_internal()
        with db_conn.cursor() as db_cursor:
            sql_insert_str = "INSERT INTO fingerprints (user_id,
embedding) VALUES (%s, %s);"
            db_cursor.execute(sql_insert_str, (data.user_id,
embedding_to_store))
            db_conn.commit()

        return ApiResponse(status="REGISTRATION_SUCCESSFUL",
user_id=data.user_id)

    except ValueError as val_err:
        raise HTTPException(status_code=400, detail=f"Invalid
template data for registration: {val_err}")
```

```
    except psycopg2.IntegrityError as integrity_err:
        if db_conn: db_conn.rollback()
        raise HTTPException(status_code=409, detail=f"User ID may
already exist or data integrity issue: {integrity_err}")
    except psycopg2.Error as db_err:
        if db_conn: db_conn.rollback()
        raise HTTPException(status_code=500, detail=f"Database
registration error: {db_err}")
    except Exception as exc:
        if db_conn: db_conn.rollback()
        raise HTTPException(status_code=500, detail=f"Internal
server error during registration: {exc}")
    finally:
        if db_conn:
            db_conn.close()

if __name__ == "__main__":
    print("Starting FastAPI server for Fingerprint Identification
API...")
    print(f"Database URL (placeholder): {DATABASE_URL}")
    print(f"Vector Dimension Expected:
{FPM_TEMPLATE_VECTOR_DIMENSION}")
    print(f"Similarity Distance Threshold: <
{SIMILARITY_DISTANCE_THRESHOLD}")
    print("Ensure PostgreSQL with pgvector is running and the
'fingerprints' table is set up correctly.")
    uvicorn.run(app, host="0.0.0.0", port=8000)
```

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: Артем САХНО

Співавтор:

Назва: Сахно_Система автоматичного контролю доступу до приміщень за допомогою біометричних даних на базі мікроконтролеру Arduino ATMega328

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:2.1%

Коефіцієнт подібності 2:0.6%

Мікропробіли: 8

Заміна букв: 7

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-05-29 08:04:36.0

Після аналізу Звіту подібності констатую наступне:

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

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

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

Обгрунтування:

2025-05-29 Доцент Андрій Нічепорук

Дата експерт

Thu May 29 11:44:30 EEST 2025, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 24.0%

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

	Document		Sum coincidence on the DB	
ID: 242401 Title: БКР Система автоматичного контролю доступу до приміщень за допомогою біометричних даних на базі мікроконтролеру Arduino ATMega328 Added in a DB: 2025-05-29 Authors: Артем САХНО Heads: Олег САВЕНКО Consultants: Opponents:	Symbols	Lexemes	Symbols	Lexemes
	140037	893	34258 (24%)	206 (23%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes
240803	Title: Звіт з ПДП Система автоматичного контролю доступу до приміщень за допомогою біометричних даних на базі мікроконтролеру Arduino ATMega328 Added in a DB: 2025-05-04 Authors: Сахно А.Є. Heads: Савенко О.С. Consultants: Opponents:	33444 (24.0%)	196 (22.0%)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Сахно Артем Євгенович

Тема: Система автоматичного контролю доступу до приміщень за допомогою біометричних даних на базі мікроконтролеру Arduino ATMega328

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 83

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка та дослідження функціональних характеристик системи автоматичного контролю доступу до приміщень, що базується на мікроконтролері Arduino та використовує біометричні дані відбитків пальців для ідентифікації користувачів.

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 дипломної роботи проведено грунтовне дослідження теоретичних та практичних аспектів побудови систем автоматичного контролю доступу із використанням біометричних даних. У другому розділі було розроблено та теоретично обгрунтовано архітектуру автоматизованої системи контролю доступу, що базується на біометричній ідентифікації користувачів за відбитками пальців та було створено комплексну апаратно-програмну архітектуру системи біометричного контролю доступу. У третьому розділі було здійснено комплексне проєктування та детально описано етапи програмної реалізації автоматизованої системи контролю доступу на базі мікроконтролера Arduino ATMega328.

4. Позитивні сторони роботи: достатня практична цінність роботи.

5. Негативні сторони роботи: обмежена продуктивність мікроконтролера Arduino Uno R3, що може впливати на швидкість обробки біометричних даних у реальному часі, використання оптичного сенсора FPM10A має певні обмеження

щодо розпізнавання при забрудненні або пошкодженні відбитків пальців. Крім того, клієнт-серверна архітектура потребує стабільного з'єднання між мікроконтролером і ПК, що може ускладнити застосування системи в деяких умовах.

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 належному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки "добре", 4.50 (B).

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Сегодіх Микола Васильович, ст. викладач, доктор філософії, кафедра Кібербезпеки

"06" 06 2025 р. _____ (підпис)

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

Артема САХНА

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

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

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

06.06. 2025 року

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

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система автоматичного контролю доступу до приміщень за допомогою біометричних даних на базі мікроконтролеру Arduino ATMega328

Автор: Артем САХНО

Спеціальність: 123– Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Олег САВЕНКО, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

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

1) містяться переважно в аналітичних розділах, не стосуються авторських результатів;
2) усі запозичення мають належні посилання або є загальновживаними технічними формулюваннями;
3) деякі збіги стосуються технічних назв, кодів і таблиць, що не є об'єктом авторського права;
4) ймовірні «спотворення» викликані автоматичним форматуванням і не є навмисною модифікацією;
5) рівень подібності є низьким (2.1%) і не перевищує допустимих меж, що підтверджено офіційним експертним висновком.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає за коефіцієнтом подібності 1 2.1%, а за коефіцієнтом подібності 2 – 0.6% і адресується до 28 першоджерел; за даними системи Anti-Plagiarism складає 24%, що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи — Олег САВЕНКО

Гарант ОП — Андрій НІЧЕПОРУК

Завідувач кафедри КІІС — Ольга ПАВЛОВА