Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: *Комп'ютерна мультимодальна система біометричної автентифікації на основі Raspberry Pi*

Виконав: студент 4 курсу, групи СІ-42
спеціальності 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

(підпис) *Ковалишин В.П.* (прізвище та ініціали)

Керівник (підпис) *Паламар А.М.* (прізвище та ініціали)

Нормоконтроль (підпис) *Луцик Н.С.* (прізвище та ініціали)

Завідувач кафедри (підпис) Осухівська Г.М. (прізвище та ініціали)

Рецензент (підпис) *Матійчук Л.П.* (прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет *комп'ютерно-інформаційних систем і програмної інженерії*
(повна назва факультету)

Кафедра *комп'ютерних систем та мереж*
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ *Осухівська Г.М.*
(підпис) (прізвище та ініціали)
«*25*» *квітня* 2026 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня *бакалавр*
(назва освітнього ступеня)

за спеціальністю *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

студента *Ковалишина Віктора Петровича*
(прізвище, ім'я, по батькові)

1. Тема роботи *Комп'ютерна мультимодальна система біометричної автентифікації на основі Raspberry Pi*

Керівник роботи *Паламар Андрій Михайлович, к.т.н., доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від *« 25 » квітня 2026 року № 4/9-188*
2. Термін подання студентом завершеної роботи 16.06.2026
3. Вихідні дані до роботи *Технічне завдання*

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Аналіз технічного завдання
2. Проєктна частина
3. Практична частина
4. Безпека життєдіяльності, основи охорони праці.
Висновки
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Структурна схема системи
2. Схема електрична принципова
3. Блок-схема алгоритму роботи
4. Структурна схема програмного забезпечення

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	*Сенчишин В.С., к.т.н., доц. каф. МТ*		

7. Дата видачі завдання *25.04.2026 р.*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	*Розробка технічного завдання*	*26.01 – 02.02*	
2.	*Аналіз технічного завдання, вимог до комп'ютерної системи, та можливих рішень поставленого завдання*	*03.02 – 15.02*	
3.	*Розроблення структури, вибір апаратного забезпечення, проєктування комп'ютеризованої системи*	*20.04 – 28.04*	
4.	*Реалізація алгоритму, написання програмного забезпечення, моделювання комп'ютерної системи*	*29.04 – 09.05*	
5.	*Безпека життєдіяльності, основи охорони праці*	*10.05 – 20.05*	
6.	*Оформлення пояснювальної записки і графічного матеріалу*	*21.05 – 7.06*	
7.	*Перевірка на академічний плагіат, перевірка керівником та консультантами*	*8.06 – 14.06*	
8.	*Попередній захист кваліфікаційної роботи бакалавра*	*15.06 – 21.06*	
9.	*Захист кваліфікаційної роботи бакалавра*	*23.06.2026*	

Студент _____ *Ковалишин В.П.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Паламар А.М.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Ковалишин В.П. Комп’ютерна мультимодальна система біометричної автентифікації на основі Raspberry Pi: робота на здобуття освітнього ступеня бакалавра: спец. 123 — комп’ютерна інженерія. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2026.

Ключові слова: мультимодальна біометрія, автентифікація, одноплатний мікрокомп'ютер, розпізнавання облич, голосовий пароль, Telegram-бот, OpenCV, розумний замок.

Кваліфікаційна робота присвячена розробці апаратно-програмного комплексу для керування фізичним доступом за допомогою двофакторної біометричної автентифікації (голос та геометрія обличчя). Спроєктовано апаратну частину на базі одноплатного мікрокомп’ютера з використанням мікрофона, камери та сервоприводу. Програмне забезпечення створено мовою Python (OpenCV, SpeechRecognition) із реалізацією Telegram-бота для віддаленого моніторингу та фотофіксації спроб доступу. Для енергозбереження систему обладнано кнопкою апаратного пробудження. Тестування підтвердило надійність системи, її стійкість до шумів та високу швидкість реакції на несанкціонований вхід.

Отримані результати можуть бути безпосередньо використані для створення сучасних автономних систем безпеки розумного будинку чи офісних приміщень. Розроблений комплекс відрізняється економічною доцільністю, високим рівнем захисту даних, масштабованістю та простотою інтеграції в існуючу інфраструктуру.

ANNOTATION

Kovalyshyn V.P. Multimodal Biometric Authentication Computer System Based on Raspberry Pi. Bachelor's Graduation Thesis: speciality 123 — Computer engineering. Ternopil: Ternopil Ivan Puluj National Technical University, 2026.

Keywords: multimodal biometrics, authentication, single-board microcomputer, face recognition, voice password, Telegram bot, OpenCV, smart lock.

The qualification work is devoted to developing a hardware and software complex for physical access control using two-factor biometric authentication (voice and facial geometry). The hardware is designed based on a single-board microcomputer using a microphone, a camera, and a servo drive. The software is written in Python (OpenCV, SpeechRecognition) and features a Telegram bot for remote monitoring and photo recording of access attempts. A hardware wake-up button is implemented for power saving. Testing confirmed the system's reliability, noise resistance, and rapid response to unauthorized access.

The obtained results can be directly used to create modern autonomous security systems for smart homes or office premises. The developed complex is characterized by economic feasibility, a high level of data protection, scalability, and ease of integration into existing infrastructure.

ЗМІСТ

СПИСОК СКОРОЧЕНЬ 7
ВСТУП 8
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ 10
1.1 Огляд сфер застосування мультимодальних систем біометричної автентифікації..... 10
1.2 Аналіз вимог до комп'ютерної мультимодальної системи біометричної автентифікації..... 11
1.3 Огляд існуючих засобів для біометричного контролю фізичного доступу приміщень 12
1.3.1 Біометричні термінали контролю доступу ZKTeco серії ProFace..... 13
1.3.2 Система розпізнавання облич Hikvision MinMoe 14
1.3.3 Розумний замок із мультимодальною автентифікацією Xiaomi Smart Door Lock 15
1.4 Аналіз можливих рішень поставленого завдання..... 16
РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА..... 18
2.1 Розробка структури комп'ютерної мультимодальної системи біометричної автентифікації..... 18
2.2 Розробка апаратного забезпечення комп'ютерної системи 20
2.2.1 Одноплатний мікрокомп'ютер Raspberry Pi..... 20
2.2.2 Модуль відеокамери 22
2.2.3 Периферійні аудіопристрої системи 22
2.2.4 Сервопривід керування замком.....26
2.3 Розробка електричної схеми пристрою..... 28
2.4 Обґрунтування вибору середовища розробки..... 30

					КС КРБ 123.175.00.00 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Ковалишин В.П.			Комп'ютерна мультимодальна система біометричної автентифікації на основі Raspberry Pi	Літ.	Арк. 5 / Аркушів
Перевір.		Паламар А.М.					
Реценз.		Матійчук Л.П.				ТНТУ, каф. КС, гр. СІ-42	
Н. Контр.		Луцик Н.С.					
Затверд.		Осухівська Г.М.					

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА 31
3.1 Розробка алгоритму роботи мультимодальної системи біометричної автентифікації 31
3.2 Розробка програмного забезпечення 34
3.2.1 Опис використаних бібліотек 35
3.2.2 Підпрограма голосової автентифікації 37
3.2.3 Підпрограма візуальної автентифікації 38
3.2.4 Головний цикл програми та керування сервоприводом 39
3.3 Реалізація системи сповіщень за допомогою Telegram-бота 41
3.3.1 Опис Telegram API та процесу реєстрації бота 41
3.3.2 Реалізація відправки тривожних повідомлень та фотофіксації 42
3.4 Тестування системи в реальних умовах 44
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ . 48
4.1 Можливість виникнення статичної електрики та заходи боротьби з нею 48
4.2 Захист електрообладнання від короткого замикання, перенавантаження 50
ВИСНОВКИ 54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 56
Додаток А Технічне завдання
Додаток Б Перелік елементів
Додаток В Лістинг програми

СПИСОК СКОРОЧЕНЬ

СКУД – Система контролю та управління доступом;

ЦОД – Центри обробки даних;

ШІМ – Широтно-імпульсна модуляція;

КПД – коефіцієнт помилкового допуску;

КПВ – коефіцієнт помилкового відмови;

HOG – Histogram of Oriented Gradients;

MIPI-CSI – Mobile Industry Processor Interface - Camera Serial Interface;

MIPI-DSI – Mobile Industry Processor Interface - Display Serial Interface;

RFID – Radio Frequency Identification;

UVC – USB Video Class.

ВСТУП

Забезпечення надійної безпеки та контролю фізичного доступу до приміщень є однією з найважливіших проблем у сучасному світі. Традиційні засоби захисту, такі як механічні ключі, магнітні картки або прості цифрові паролі, мають суттєві недоліки, оскільки їх можна легко загубити, викрасти чи передати стороннім особам. Використання біометричних технологій вирішує частину цих проблем, проте однофакторні системи (наприклад, перевірка лише за відбитком пальця або лише за обличчям) залишаються вразливими до методів підміни та негативного впливу зовнішнього середовища. Впровадження мультимодальних систем, які комбінують кілька незалежних біометричних ознак, дозволяє суттєво підвищити точність ідентифікації та загальний рівень захисту об'єкта.

У зв’язку з цим актуальною є задача розроблення комп’ютерної мультимодальної системи біометричної автентифікації на базі високопродуктивних мікрокомп'ютерів. Така система дозволить не лише гарантувати високу надійність перевірки користувача за допомогою двофакторної авторизації, а й автоматизувати процес фізичного доступу, наприклад, відмикати механізм замка лише при збігу геометрії обличчя і голосового пароля, паралельно забезпечуючи дистанційний моніторинг ситуації.

Мета роботи – розробити комп’ютерну мультимодальну систему біометричної автентифікації на основі Raspberry Pi, яка забезпечить точну двофакторну перевірку користувача, автоматизоване керування механізмом замка, оптимізацію енергоспоживання та миттєве віддалене сповіщення власника про спроби доступу [1].

Для досягнення поставленої мети необхідно вирішити такі завдання:

− вивчити існуючі підходи до побудови систем контролю фізичного доступу та визначити недоліки класичних і однофакторних методів автентифікації;

− сформулювати технічні вимоги до проєктуємої мультимодальної системи на основі аналізу потреб безпеки приміщень;

− дослідити існуючі рішення та обґрунтувати вибір апаратної складової, зокрема одноплатного мікрокомп’ютера та необхідних периферійних модулів (камери, мікрофона, сервоприводу);

− спроєктувати структуру та алгоритм роботи пристрою, передбачивши апаратний режим очікування для ефективного енергозбереження;

− створити програмне забезпечення системи мовою Python, реалізувавши алгоритми комп'ютерного зору, розпізнавання голосу, ШІМ-керування замком та інтеграцію з месенджером Telegram;

− провести тестування розробленої системи в умовах реальної експлуатації, оцінити її стійкість до фонових шумів та загальну ефективність роботи.

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Огляд сфер застосування мультимодальних систем біометричної автентифікації

Сучасний етап розвитку СКУД характеризується стрімким переходом від традиційних методів ідентифікації (механічні ключі, RFID-картки, PIN-коди) до біометричних технологій. Незважаючи на широке розповсюдження однофакторних біометричних систем (наприклад, сканерів відбитків пальців або систем розпізнавання облич), вони мають ряд суттєвих недоліків, таких як вразливість до методів підміни та залежність від умов навколишнього середовища. Вирішенням цих проблем є мультимодальні біометричні системи, які комбінують дві або більше незалежних біометричних характеристик, що значно підвищує точність розпізнавання та надійність захисту [2]. Мультимедіа це не тільки спосіб подачі контенту, а й технології. Вони базуються на апаратних та програмних засобах, які також включають різні елементи.

Сфери застосування мультимодальних систем автентифікації постійно розширюються. Насамперед, вони є критично важливими для об'єктів підвищеної безпеки, таких як урядові установи, банківські сховища, ЦОД та наукові лабораторії. У таких приміщеннях комбінація розпізнавання обличчя та голосу (або відбитка пальця) гарантує, що доступ отримає виключно авторизована особа, унеможливлюючи використання вкрадених перепусток чи фотографій.

Сучасні системи контролю доступу все частіше використовують методи машинного навчання для аналізу різноманітних біометричних параметрів користувача, починаючи від динаміки натискання клавіш і закінчуючи

комплексним розпізнаванням обличчя та голосу [3].

У корпоративному секторі та офісних центрах мультимодальні системи застосовуються для розмежування доступу до різних зон (наприклад, вільний доступ до загальних коридорів за обличчям, але доступ до кабінету керівництва чи серверної лише за умови додаткового підтвердження голосом). Це дозволяє гнучко налаштовувати політику безпеки.

Останнім часом такі технології активно впроваджуються у сферу розумних будинків (Smart Home). Використання мікрокомп'ютерних систем дозволяє власникам приватних будинків та квартир відмовитися від носіння ключів. Система може автоматично розпізнавати жителів за обличчям при наближенні до дверей і просити вимовити кодову фразу для остаточного підтвердження доступу. Це не лише зручно, але й забезпечує високий рівень захисту від несанкціонованого проникнення [2].

Крім того, мультимодальна біометрія знаходить застосування в медичних установах для доступу до приміщень зі зберіганням наркотичних препаратів або конфіденційних баз даних пацієнтів, де безконтактна авторизація (обличчя + голос) є особливо важливою з санітарно-епідеміологічних міркувань.

1.2 Аналіз вимог до комп’ютерної мультимодальної системи біометричної автентифікації

Розробка комп’ютерної мультимодальної системи біометричної автентифікації на базі мікрокомп'ютера (наприклад, Raspberry Pi) вимагає чіткого формулювання функціональних, технічних та експлуатаційних вимог, що гарантуватимуть її надійність, безпеку та зручність.

Функціональні вимоги визначають основні завдання, які має виконувати система. Головною вимогою є забезпечення двофакторної автентифікації: система повинна безперервно або за тригером зчитувати відеопотік для виявлення обличчя, а також записувати аудіосигнал для розпізнавання

голосової команди (пароля). У разі успішного збігу обох параметрів з базою еталонів, система має подавати керуючий сигнал на виконавчий механізм (сервопривід), що імітує відкриття замка. Додатковою важливою функцією є система сповіщень: пристрій повинен мати інтеграцію з месенджерами (наприклад, Telegram) для відправки повідомлень та фотографій користувача під час успішних чи невдалих спроб доступу.

Технічні вимоги стосуються апаратної бази та точності алгоритмів. Згідно з міжнародними стандартами оцінки безпеки біометричних систем, ключовими показниками є коефіцієнт помилкового допуску (КПД) та коефіцієнт помилкової відмови (КПВ) [4]. Використання двох модальностей (обличчя і голос) дозволяє знизити КПД практично до нуля. Апаратна частина має бути достатньо продуктивною для локальної обробки алгоритмів комп'ютерного зору (наприклад, бібліотеки OpenCV) без значних затримок. Відповідно, система повинна включати мікрокомп'ютер, модуль камери з достатньою роздільною здатністю, мікрофон із функцією шумозаглушення, апаратну кнопку та сервопривід.

Експлуатаційні та енергозберігаючі вимоги. Оскільки система працюватиме як дверний замок, вона не повинна постійно споживати максимальну потужність. Необхідно реалізувати режим очікування (сплячий режим), вихід з якого відбуватиметься за допомогою апаратного переривання (натискання тактової кнопки). Крім того, алгоритми розпізнавання голосу повинні бути стійкими до фонового шуму приміщення, а комп'ютерний зір — адаптуватися до різного рівня освітлення.

1.3 Огляд існуючих засобів для біометричного контролю фізичного доступу приміщень

На сучасному ринку систем безпеки представлено широкий спектр засобів для біометричного контролю фізичного доступу. Вони різняться за використовуваними модальностями, форм-фактором, рівнем безпеки та

ціновим сегментом. Для обґрунтування власної розробки доцільно розглянути популярні комерційні рішення та проаналізувати їхні переваги й недоліки.

1.3.1 Біометричні термінали контролю доступу ZKTeco серії ProFace

Серія терміналів ProFace від компанії ZKTeco є одним із передових рішень у сфері корпоративного контролю доступу (рис. 1.1).

Рисунок 1.1 – ZKTeco

Ці пристрої підтримують мультимодальну автентифікацію, комбінуючи розпізнавання обличчя, малюнка вен долоні, відбитка пальця та традиційних RFID-карт [5].

Основною перевагою системи є надвисокий рівень безпеки та захист від підробок (анти-спуфінг алгоритми), що дозволяє відрізняти живе обличчя від фотографії чи маски. Термінали мають високу швидкість розпізнавання та здатні зберігати тисячі шаблонів локально. Проте, основними недоліками є висока вартість обладнання, закритість програмного забезпечення та надмірність функціоналу для застосування у невеликих приміщеннях або системах розумного будинку. Крім того, ці термінали зазвичай не

підтримують голосову автентифікацію, що унеможливлює використання секретних кодових слів [5].

1.3.2 Система розпізнавання облич Hikvision MinMoe

Термінали Hikvision серії MinMoe (рис. 1.2). набули значної популярності завдяки використанню алгоритмів глибокого навчання (Deep Learning), що забезпечує точність розпізнавання облич на рівні понад 99 % навіть у складних умовах освітлення [6].

Рисунок 1.2 – Hikvision серії MinMoe

Система забезпечує безконтактний доступ і часто комбінується із вимірюванням температури тіла.

Перевагами Hikvision MinMoe є швидкість спрацьовування (менше 0.2 секунди) та можливість глибокої інтеграції з системами відеоспостереження. Недоліком є те, що у базових версіях це переважно однофакторна система (лише обличчя). Хоча існує можливість використання карток, повноцінної мультимодальної біометрії (наприклад, поєднання обличчя та голосу) не передбачено. Також, як і у випадку із ZKTeco, це комерційно закрите рішення, яке важко кастомізувати під специфічні індивідуальні сценарії з використанням IoT-сповіщень у популярні месенджери [6].

1.3.3 Розумний замок із мультимодальною автентифікацією Xiaomi Smart Door Lock

Для приватного сектору (квартири та будинки) найпопулярнішими рішеннями є розумні замки типу Xiaomi Smart Door Lock (рис. 1.3).

Рисунок 1.3 – Xiaomi Smart Door Lock

Вони пропонують "все-в-одному" рішення: відкриття за допомогою відбитка пальця, PIN-коду, NFC, Bluetooth, а у флагманських моделях — за допомогою 3D-сканування обличчя [7].

Беззаперечними перевагами таких замків є їхній естетичний вигляд, автономність (робота від батарейок протягом кількох місяців) та інтеграція в екосистему розумного дому. Однак існують і суттєві недоліки. По-перше, голосове розпізнавання в таких системах практично не використовується як засіб безпеки (голос використовується лише для керування асистентами всередині дому, а не для відмикання дверей зовні). По-друге, користувач обмежений пропрієтарним додатком виробника, який часто зберігає дані на закордонних хмарних серверах. Неможливість самостійно змінити логіку роботи або налаштувати пряму відправку фотографій зловмисників у власний Telegram-канал робить такі пристрої менш гнучкими для розробників та ентузіастів [7].

Аналіз існуючих засобів свідчить, що комерційні рішення є або занадто дорогими і закритими (корпоративний сектор), або не підтримують потрібну

комбінацію модальностей (обличчя + голос) і гнучкість налаштувань (приватний сектор). Це повністю обґрунтовує доцільність розробки власної мультимодальної системи на базі мікрокомп'ютера з відкритим вихідним кодом.

1.4 Аналіз можливих рішень поставленого завдання

Розроблення комп’ютерної мультимодальної системи біометричної автентифікації вимагає детального аналізу підходів до побудови апаратно-програмного комплексу. Головним викликом є обробка мультимедійних даних з камери та мікрофона у реальному часі та управління апаратною периферією.

Вибір остаточного архітектурного рішення базується на методах порівняльної оцінки програмного забезпечення, що дозволяє знайти оптимальний компроміс між продуктивністю локальних обчислень та енергоефективністю системи [8].

Для вирішення цього завдання можна застосувати мікроконтролери з хмарною обробкою даних. Цей підхід забезпечує низьке енергоспоживання, проте критично залежить від стабільного інтернет-з'єднання і створює загрози конфіденційності при передачі біометрії через відкриті мережі. Іншим варіантом є використання повноцінного персонального комп'ютера. Він надає необмежені обчислювальні потужності, але відрізняється високою вартістю, великими габаритами та значним енергоспоживанням.

Найбільш оптимальним і збалансованим рішенням є використання одноплатного мікрокомп'ютера. Пристрої такого класу забезпечують необхідну продуктивність для локальної обробки алгоритмів комп'ютерного зору безпосередньо на місці встановлення. Наявність повноцінної операційної системи сімейства Linux та вбудованих портів вводу-виводу дозволяє безперешкодно керувати сервоприводом і зчитувати стан апаратних переривань. Локальна обробка біометрії забезпечує безпеку та незалежність від хмарних сервісів. Для нівелювання підвищеного енергоспоживання

доцільно реалізувати режим очікування з пробудженням системи за допомогою тактової кнопки.

Програмну частину найефективніше реалізувати мовою Python завдяки її потужній екосистемі для задач комп'ютерного зору. Розпізнавання облич оптимально виконувати за допомогою бібліотеки OpenCV, а обробку голосового пароля реалізувати через модуль SpeechRecognition. Для віддаленого моніторингу та сповіщень раціонально використати інтеграцію з Telegram API. Це дозволяє миттєво надсилати власнику текстові повідомлення і фотографії у разі несанкціонованої спроби входу без необхідності розробки окремого мобільного застосунку. Такий комплексний підхід повністю відповідає технічним вимогам і дозволяє створити надійну та автономну систему безпеки.

РОЗДІЛ 2 ПРОЄКТНА ЧАСТИНА

2.1 Розробка структури комп'ютерної мультимодальної системи біометричної автентифікації

Проєктована комп'ютерна мультимодальна система біометричної автентифікації має чітку архітектуру, що складається з пристроїв збору інформації, центрального обчислювального блоку, а також засобів виводу та мережевої комунікації. На рисунку 2.1 зображена структурна схема розробленої системи.

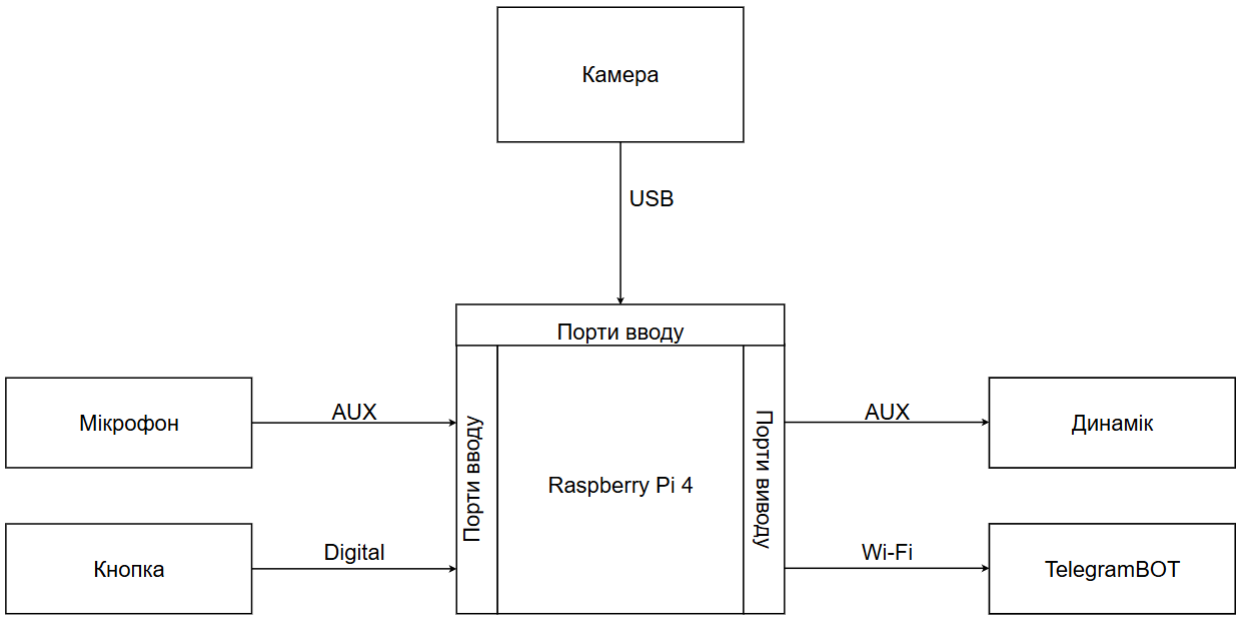

Рисунок 2.1 – Структурна схема комп'ютерної мультимодальної системи біометричної автентифікації

Основу системи становить високопродуктивний одноплатний мікрокомп'ютер Raspberry Pi 4, який виконує роль центрального ядра. Він відповідає за зчитування мультимедійних даних, їх локальну обробку

нейромережевими алгоритмами (комп'ютерний зір та розпізнавання мови) та формування відповідних керуючих сигналів. До портів вводу обчислювального блоку підключено наступні периферійні пристрої, які забезпечують збір біометричних даних та керування режимами роботи, а саме камера, мікрофон та кнопка.

Камера, яка взаємодіє з мікрокомп'ютером через високошвидкісний інтерфейс USB. Модуль відповідає за безперервне або тригерне захоплення відеопотоку, необхідного для виявлення та розпізнавання геометрії обличчя користувача.

Мікрофон, який підключається через аналоговий аудіовхід AUX. Призначений для запису голосового пароля. Для коректної роботи аудіовходу задіюються апаратні аудіокодеки мікрокомп'ютера, що забезпечує чисте оцифрування звуку без апаратних затримок.

Кнопка, яка підключена до цифрового порту вводу-виводу загального призначення (Digital / GPIO). Виконує роль апаратного переривання для виведення системи з енергоефективного режиму очікування (сну) та ініціації процесу автентифікації.

До портів виводу системи належать компоненти зворотного зв'язку та віддаленого моніторингу, а саме : динамік та TelegramBOT.

Динамік, який підключений через аудіовихід AUX, забезпечує локальний звуковий зворотний зв'язок з користувачем, наприклад, відтворення сервісних повідомлень про успішну авторизацію, помилку доступу або прохання повторити голосову команду.

TelegramBOT, що реалізується через дротове мережеве підключення Ethernet або за допомогою встроєного модуля Wi-Fi, забезпечує зв'язок пристрою з глобальною мережею Інтернет для відправки миттєвих сповіщень (текстових повідомлень та фотографій з камери) на мобільний пристрій власника у разі спроб доступу до приміщення.

Запропонована структура системи дозволяє реалізувати надійний цикл двофакторної біометричної перевірки, зберігаючи високу швидкість обробки

даних завдяки потужностям Raspberry Pi 4 та забезпечуючи зручний віддалений контроль через месенджер.

2.2 Розробка апаратного забезпечення комп’ютерної системи

Апаратне забезпечення комп'ютерної мультимодальної системи біометричної автентифікації спроєктовано з урахуванням необхідності обробки мультимедійних даних у режимі реального часу, підтримки стабільного мережевого з'єднання та мінімізації енергоспоживання. Комплекс складається з центрального мікрокомп'ютера та набору периферійних модулів для введення і виведення інформації [9].

2.2.1 Одноплатний мікрокомп’ютер Raspberry Pi

У ролі центрального обчислювального ядра системи використано високопродуктивний одноплатний мікрокомп’ютер Raspberry Pi 4 (рис. 2.2).

Рисунок 2.2 – Одноплатний мікрокомп’ютер Raspberry Pi 4

Цей пристрій побудований на базі чотириядерного 64-бітного процесора Broadcom BCM2711 (чотири ядра ARM Cortex-A72), що забезпечує високу швидкість виконання багатопоточних задач комп'ютерного зору та цифрової обробки сигналів порівняно з попередніми поколіннями одноплатних обчислювальних платформ [9].

Технічні характеристики одноплатного мікрокомп’ютера Raspberry Pi

наведені в таблиці 2.1.

Таблиця 2.1 – Характеристики Raspberry Pi

Параметр	Значення
Центральний процесор (CPU)	Чотириядерний 64-бітний Broadcom BCM2711
Графічний процесор (GPU)	Broadcom VideoCore VI
Оперативна пам'ять (RAM)	4 ГБ LPDDR4-3200 SDRAM
Вбудована пам'ять (Storage)	Слот для карти MicroSD
Мережеві інтерфейси	Гігабітний Ethernet, дводіапазонний Wi-Fi (2.4/5.0 ГГц) 802.11ac, Bluetooth 5.0, BLE
Відеовиходи	2 x micro-HDMI , MIPI-DSI
Аудіоінтерфейси	4-контактний роз'єм 3.5 мм , цифрове аудіо через HDMI
Інтерфейси камери	1 x MIPI-CSI
Порти USB	2 x USB 3.0, 2 x USB 2.0
Низькорівневі інтерфейси	40-пінова колодка GPIO (підтримка I2C, SPI, UART, PWM, Digital I/O)
Живлення	5 В / 3 А через порт USB Type-C, живлення через GPIO, підтримка PoE
Операційні системи	Raspberry Pi OS (Debian), Ubuntu, Windows 10 IoT Core, RetroPie та ін.

Наявність 4 ГБ оперативної пам'яті стандарту LPDDR4-3200 SDRAM гарантує відсутність затримок під час завантаження й розгортання нейромережевих моделей у пам'ять пристрою. Інтеграція периферійних пристроїв здійснюється через 40-пінову колодку GPIO, порти USB 3.0/2.0 та гігабітний Ethernet-інтерфейс. Для забезпечення стабільної роботи апаратних

інтерфейсів на рівні операційної системи Linux було здійснено конфігурацію кастомних оверлеїв дерева пристроїв (device tree overlays) та налаштування параметрів ядра, що дозволило уникнути конфліктів при розподілі переривань [9].

Для забезпечення обробки відеопотоку та аудіосигналів у режимі реального часу застосовано принципи паралельних та розподілених обчислень, що ефективно реалізуються на багатоядерній архітектурі сучасного процесора Broadcom [10].

2.2.2 Модуль відеокамери

Для захоплення відеопотоку й фіксації геометрії обличчя у системі використано відеокамеру Creality Nebula (рис. 2.3).

Рисунок 2.3 – Відеокамера Creality Nebula

У структурі двофакторної автентифікації вона виконує роль оптичного сенсора для підпрограми візуального контролю. Пряме підключення пристрою до порту USB мікрокомп’ютера Raspberry Pi 4 мінімізує затримки передачі даних, забезпечуючи обробку інформації в режимі реального часу. Обґрунтування вибору даної моделі камери зумовлене її високими технічними та експлуатаційними показниками (табл. 2.2), які безпосередньо впливають на точність роботи нейромережевих алгоритмів розпізнавання.

Таблиця 2.2 – Характеристики Creality Nebula

Параметр	Значення
Роздільна здатність	1920x1080 (Full HD)
Кут огляду (FOV)	~118° (ширококутний об'єктив)
Нічне бачення	Підтримується
Інтерфейс підключення	USB 2.0 (Type-A)
Протокол обміну даними	UVC (USB Video Class)
Напруга живлення	5 В (від порту USB)
Сумісність з ОС	Linux, Windows, macOS

Камера має апаратну роздільну здатність Full HD (1920×1080 пікселів), що дозволяє програмній бібліотеці OpenCV формувати чіткі кадри з високою щільністю точок, необхідною для безпомилкової локалізації ключових антропометричних точок обличчя користувача. Завдяки ширококутному об'єктиву з кутом огляду близько 118 градусів, пристрій забезпечує широку зону охоплення, що суттєво підвищує ергономічність системи, оскільки користувачеві не потрібно суворо позиціонувати голову на фіксованій висоті перед запірним механізмом.

Взаємодія обчислювального ядра із відеокамерою реалізована на базі стандартизованого міжнародного протоколу USB Video Class (UVC). Інтеграція цього протоколу на рівні операційної системи Linux (Ubuntu) забезпечує повну сумісність обладнання за принципом «Plug-and-Play». Це повністю нівелює необхідність розробки чи встановлення додаткових сторонніх пропрієтарних драйверів, гарантуючи високу стабільність програмно-апаратного інтерфейсу та виключаючи ризик виникнення системних конфліктів при розподілі динамічної пам'яті [11].

Оскільки система контролю фізичного доступу орієнтована на безперервне автономне функціонування, критично важливою характеристикою обраного модуля є наявність вбудованого інфрачервоного

(ІЧ) нічного бачення. Дане інженерне рішення дозволяє оптичній системі безперешкодно фіксувати геометрію обличчя авторизованих осіб та здійснювати чітку фотофіксацію потенційних зловмисників у сутінках або за умов повної темряви, що значно підвищує захищеність об'єкта.

2.2.3 Периферійні аудіопристрої системи

Голосова автентифікація користувача та відтворення сервісних сповіщень вимагають високої якості оцифрування й відтворення звукового потоку без затримок та сторонніх спотворень. Для комплексної реалізації цих функцій в апаратній структурі системи використано зовнішню звукову карту Gemix SC-02, яка підключається до мікрокомп'ютера через інтерфейс USB (рис. 2.4).

Рисунок 2.4 – Звукову карта Gemix SC-02

Даний апаратний модуль оснащений двома незалежними аналоговими роз'ємами типу AUX 3.5 мм, що мають стандартне інтерфейсне маркування і призначені для роздільного підключення ліній аудіовходу (зовнішнього мікрофона) та аудіовиходу (акустичного динаміка або навушників). Така роздільна архітектура портів дозволяє ефективно мінімізувати перехресні завади в провідниках та забезпечує стабільний повнодуплексний режим роботи аудіопідсистеми, що є обов'язковою умовою для одночасного високоякісного захоплення голосового сигналу та відтворення голосових статусів у реальному часі [12]. Зведені технічні характеристики зовнішньої

звукової карти Gemix SC-02, що відображають параметри розрядності та частоти дискретизації інтегрованих кодеків, наведені у таблиці 2.3.

Таблиця 2.3 – Характеристики Gemix SC-02

Параметр	Значення
Інтерфейс підключення до плати	USB 2.0 Type-A
Порти для периферії	2 x 3.5 мм jack (AUX мікрофон / динамік)
Режим обробки сигналів	Стерео

До інтерфейсних гнізд звукової карти Gemix SC-02 підключаються гарнітурні навушники JBL Quantum N100 (рис. 2.5), які одночасно виконують роль як високочутливого мікрофона для запису та реєстрації голосового пароля, так і динаміка для забезпечення локального звукового зворотного зв'язку з користувачем.

Рисунок 2.5 – Гарнітурні навушники JBL Quantum N100

Аналоговий сигнал проходить первинну апаратну обробку та фільтрацію безпосередньо через вбудовані аудіокодеки звукової карти. Технічні характеристики гарнітурних навушників JBL Quantum N100 наведені в таблиці 2.4.

Таблиця 2.4 – Характеристики JBL Quantum N100

Параметр	Значення
Тип підключення	Дротовий, через роз'єм AUX
Робочий діапазон частот динаміків	20 Гц – 20 000 Гц
Робочий діапазон частот мікрофона	100 Гц – 10 000 Гц
Спрямованість мікрофона	Направлений
Функціональне призначення у схемі	Реєстрація голосового пароля та звукова індикація статусів

Налагодження роботи USB-аудіопристрою на рівні архітектури підсистеми ALSA операційної системи Linux дозволило повністю усунути затримки сигналу та фонові апаратні шуми під час захоплення ефіру, що безпосередньо й критично впливає на точність аналізу та підвищує відсоток успішного розпізнавання секретного кодового слова.

2.2.4 Сервопривід керування замком

Сервопривід є основним виконавчим механізмом системи, який відповідає за фізичне керування запірним пристроєм (замком) після успішної двофакторної біометричної автентифікації користувача. Для реалізації цієї функції в апаратній частині обрано мікросервопривід типу SG90 (рис. 2.6).

Рисунок 2.6 – Мікросервопривід типу SG90

Цей вибір зумовлений його компактними габаритами, низьким рівнем споживання енергії у стані спокою та достатнім крутним моментом для імітації відкриття засувки [13].

Керування виконавчим механізмом сервоприводу здійснюється за допомогою широтно-імпульсної модуляції (ШІМ), керуючий сигнал якої програмно генерується на одному з цифрових виводів загального призначення (GPIO) мікрокомп’ютера Raspberry Pi 4. Плавно змінюючи шпаруватість робочого імпульсу (відношення тривалості імпульсу до його періоду), мікрокомп'ютер задає точний кут позиціонування валу двигуна в робочому діапазоні від 0° до 180°. Це дозволяє з високою надійністю та мінімальною механічною затримкою переводити запірний механізм замка у фіксовані положення «відкрито» або «закрито» після успішного проходження біометричної верифікації. Живлення сервоприводу здійснюється безпосередньо від вбудованої шини 5 В мікрокомп'ютера із підключенням до спільного контуру заземлення (GND), що суттєво спрощує загальну апаратну архітектуру пристрою та позбавляє необхідності використання додаткових зовнішніх модулів живлення.

Технічні характеристики мікросервоприводу типу SG90 наведені в таблиці 2.5.

Таблиця 2.5 – Характеристики SG90

Параметр	Значення
Робоча напруга	від 4,8 В до 6,0 В
Крутний момент	1,8 кг/см (при живленні 4,8 В)
Швидкість обертання	0,12 с / 60°
Інтерфейс підключення	3-провідний (VCC, GND, ШІМ-сигнал)

Використання такого сервоприводу забезпечує високу надійність роботи виконавчої частини та швидку реакцію системи на команди керуючого алгоритму.

2.3 Розробка електричної схеми пристрою

Основою для фізичної реалізації апаратної частини мультимодальної системи слугує електрична принципова схема (рис. 2.7). Вона наочно відображає логіку підключення усіх периферійних пристроїв до центрального обчислювального ядра — мікрокомп'ютера Raspberry Pi 4, а також організацію кіл живлення та передачі сигналів [14]. Дана схема розроблена з суворим дотриманням специфікацій інтерфейсів вводу-виводу плати. Живлення системи здійснюється від автономного джерела постійного струму (Power Bank) з напругою 5 В та силою струму 3 А через штатний роз'єм USB Type-C, що забезпечує стабільну роботу процесора при максимальних навантаженнях під час нейромережевого розпізнавання.

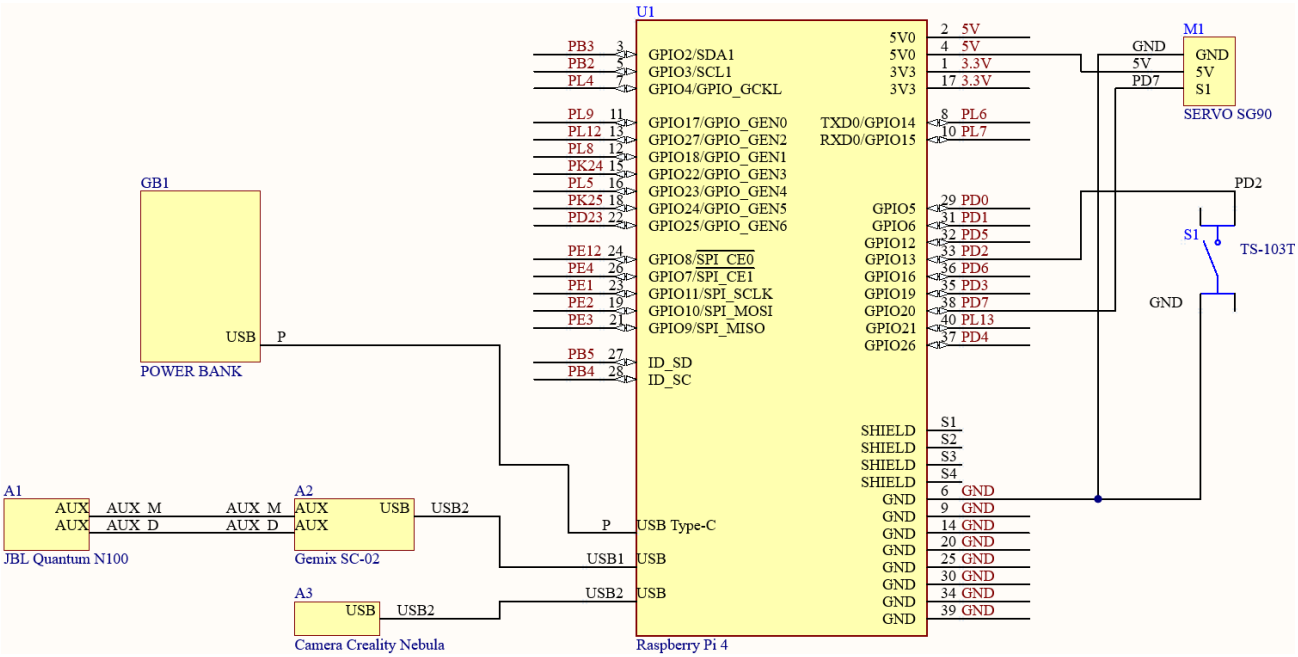

Рисунок 2.7 – Схема електрично принципова

Підключення периферійних пристроїв апаратно-програмного комплексу реалізовано наступним чином.

Джерело живлення (Power Bank) підключається безпосередньо до роз'єму USB Type-C мікрокомп'ютера Raspberry Pi 4 (на принциповій схемі позначено як лінія P). Це забезпечує стабільну вхідну напругу 5 В для

живлення основної обчислювальної плати та всієї підключеної до неї периферії.

USB-камера Creality Nebula підключається до одного з роз'ємів USB мікрокомп'ютера (лінія USB2). Використання стандартизованого інтерфейсу USB забезпечує одночасну подачу необхідного живлення на модуль камери та гарантує високошвидкісну передачу диференціального сигналу цифрового відеопотоку до центрального процесора.

Аудіосистема інтегрована в комплекс за допомогою зовнішньої звукової карти Gemix SC-02, яка підключається до вільного порту USB (лінія USB1). У свою чергу, навушники з мікрофоном JBL Quantum N100 під'єднуються до звукової карти через два роздільні аналогові роз'єми (лінії AUX_M для вхідного сигналу мікрофона та AUX_D для вихідного сигналу динаміків). Така архітектура забезпечує якісне апаратне оцифрування звуку за межами основної плати, уникаючи використання інтегрованого кодека Raspberry Pi та суттєво знижуючи рівень апаратних шумів.

Виконавчий механізм, роль якого виконує сервопривід SG90, має трипровідну лінію підключення. Контакт живлення безпосередньо підключено до шини +5 В, а земляний контакт — до спільної "землі" (GND). Сигнальний дріт (S1), який відповідає за передачу керуючого ШІМ-сигналу (PWM), підключено до цифрового виводу PD7, що фізично відповідає порту GPIO20 на стандартній 40-піновій колодці мікрокомп'ютера.

Тактова кнопка TS-103T, призначена для ініціалізації процесу розпізнавання, підключена за класичною схемою [15]. Один її контакт з'єднаний із цифровим виводом PD2 (відповідає порту GPIO13), а інший — із загальним контактом GND. Для стабілізації логічного рівня та запобігання хибним спрацьовуванням використовується внутрішній підтягуючий резистор (pull-up) мікроконтролера [15]. При фізичному натисканні кнопки електричний ланцюг замикається на "землю", формуючи на вході логічний нуль, що миттєво генерує апаратне переривання для операційної системи.

2.4 Обґрунтування вибору середовища розробки

Вибір програмного середовища та операційної системи є критично важливим етапом, оскільки від нього залежить стабільність роботи пристрою, сумісність із драйверами периферії та загальна швидкодія алгоритмів мультимодальної автентифікації.

В якості базової операційної системи для Raspberry Pi було обрано Ubuntu Image (адаптований для ARM-архітектури) [16]. Використання повноцінного дистрибутива сімейства Linux обґрунтовується низкою беззаперечних переваг:

- нативна підтримка архітектури aarch64 (ARM64);
- широка екосистема пакетів;
- стабільність мережевих сервісів.

Основним середовищем для написання програмного коду обрано високорівневу мову Python 3 [17]. Такий вибір обґрунтований тим, що ця мова є загальновизнаним стандартом у галузі машинного навчання та комп'ютерного зору.

Використання мови Python забезпечує безшовну інтеграцію всіх необхідних для проєкту бібліотек: OpenCV та dlib відповідають за захоплення відеопотоку й обробку геометрії обличчя, тоді як SpeechRecognition використовується для оцифрування та розпізнавання голосових команд. Безпосередня апаратна взаємодія реалізується через спеціалізовані модулі (наприклад, RPi.GPIO або periphery), які надають доступ до переривань від кнопки та генерують керуючий ШІМ-сигнал для сервоприводу, а для створення системи віддалених сповіщень застосовуються зручні бібліотеки-обгортки для Telegram API, такі як aiogram або telebot.

Таким чином, поєднання адаптованого образу Ubuntu та екосистеми мови Python забезпечує оптимальний баланс між продуктивністю, швидкістю розробки та надійністю готового програмно-апаратного комплексу.

РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

3.1 Розробка алгоритму роботи мультимодальної системи біометричної автентифікації

Основою програмного забезпечення розробленої мультимодальної системи біометричної автентифікації є комплексний алгоритм обробки даних, що виконує роль головного координаційного центру. Його архітектуру побудовано за математичним принципом скінченного автомата (state machine), який логічно розділяє загальний цикл роботи пристрою на окремі, суворо детерміновані стани. Такий підхід дозволяє безшовно та безпечно інтегрувати низькорівневі процеси (як-от обробку апаратних переривань від датчиків) із високорівневими ресурсоємними задачами машинного навчання та асинхронною мережевою комунікацією.

Програмний комплекс спроєктовано з глибоким урахуванням критичних вимог до енергоефективності, загальної швидкодії та максимальної відмовостійкості процесу авторизації. Для оптимізації використання апаратних ресурсів мікрокомп'ютера логіка коду передбачає раціональний розподіл обчислювальних потоків. Це означає, що найбільш складні завдання розпізнавання образів та цифрової обробки сигналів не виконуються у фоновому режимі постійно, а активуються виключно за фактичної потреби. Водночас базовий принцип безпеки об'єкта спирається на концепцію жорсткої послідовної верифікації: кожен наступний рівень перевірки розблоковується виключно після безпомилкового проходження попереднього.

Загальна структура розробленого алгоритму охоплює повний спектр автоматизованих дій — від підготовки системних ресурсів до фізичного генерування керуючих імпульсів для запірного механізму та віддаленого

інформування власника. Детальний покроковий опис цього функціонального циклу, який ілюструє переходи між базовими станами системи залежно від успішності чи неуспішності проходження біометричного контролю, послідовно наведено нижче. Блок-схему головного циклу роботи системи наведено на рисунку 3.1.

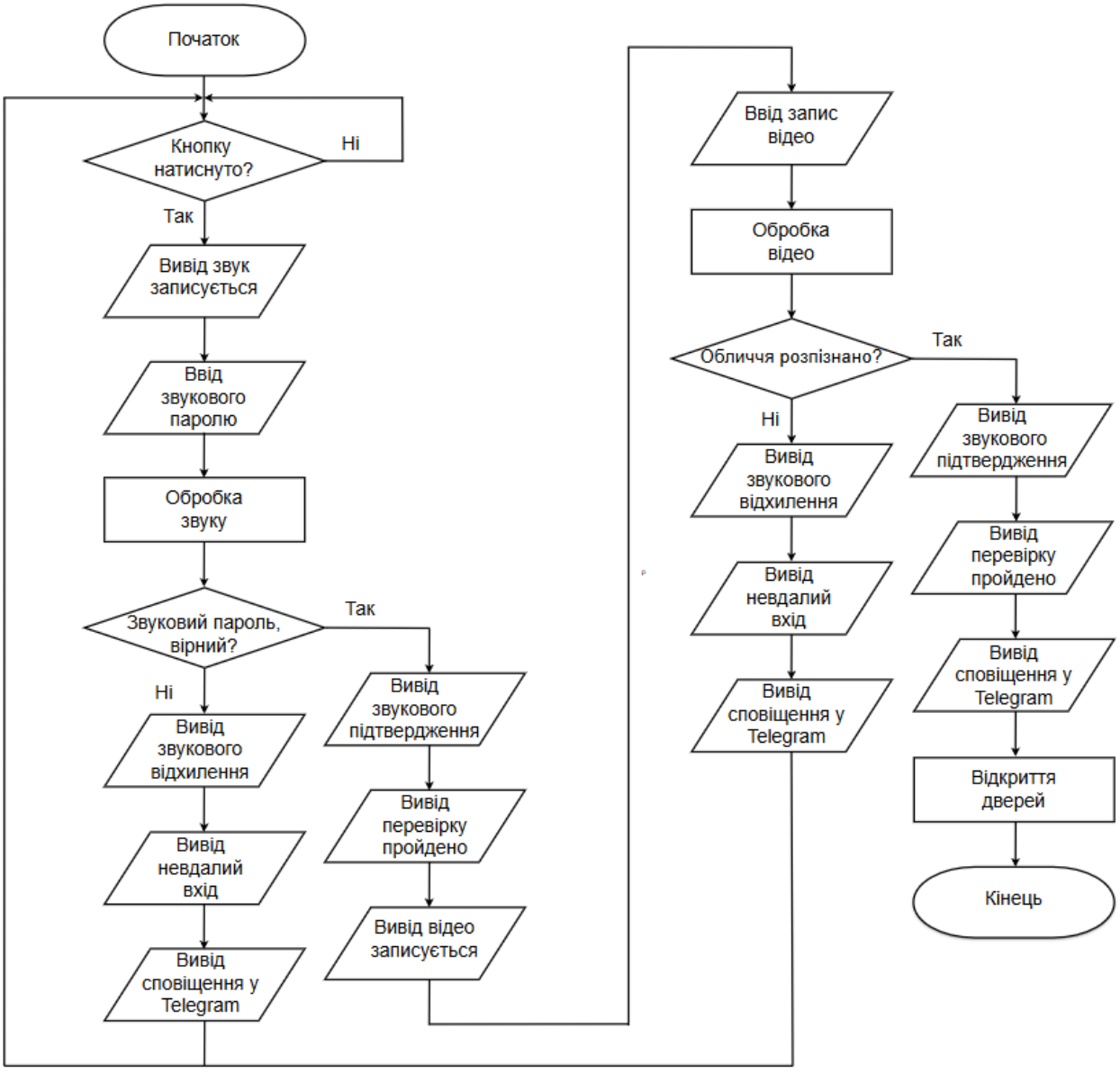

Рисунок 3.1 – Блок-схема головного циклу роботи системи

Робота пристрою розпочинається з етапу ініціалізації. На цьому кроці операційна система мікрокомп'ютера завантажує необхідні бібліотеки Python, ініціалізує порти введення-виведення (GPIO) для роботи з кнопкою та

сервоприводом, а також встановлює з'єднання з серверами Telegram API для забезпечення роботи бота. У оперативну пам'ять завантажуються каскади Хаара або попередньо натреновані нейромережеві моделі для розпізнавання геометрії обличчя, що дозволяє уникнути затримок під час безпосереднього процесу автентифікації.

Після успішного завершення ініціалізації система переходить у режим очікування (Standby). У цьому стані камера та мікрофон деактивовані, а процесор споживає мінімальну кількість електроенергії. Вихід системи зі сплячого режиму відбувається виключно за умови генерації апаратного переривання, яке ініціюється натисканням тактової кнопки користувачем.

Отримавши сигнал переривання, алгоритм запускає перший етап перевірки — візуальну автентифікацію. Камера робить знімок або записує короткий відеопотік, після чого алгоритми комп'ютерного зору здійснюють пошук обличчя в кадрі та порівнюють його з еталонними біометричними векторами власника (матрицями ознак), що зберігаються в локальній базі даних.

Якщо обличчя не розпізнано, система негайно фіксує спробу несанкціонованого доступу, робить фотографію зловмисника, відправляє її разом із тривожним текстовим повідомленням через Telegram-бот власнику та повертається у режим очікування.

У разі успішного проходження візуального контролю система ініціює другий фактор перевірки — голосову автентифікацію. Користувачу надається певний час (зазвичай кілька секунд) для виголошення секретної кодової фрази. Мікрофон записує аудіосигнал, який оцифровується та передається на модуль розпізнавання мови. Якщо розпізнаний текст збігається з еталонним паролем, автентифікація вважається повністю успішною. Якщо ж пароль хибний, система аналогічно формує повідомлення про відмову в доступі та блокує подальше виконання циклу.

Завершальним етапом алгоритму, за умови успішного проходження обох факторів біометричної перевірки, є активація виконавчого механізму.

Мікрокомп'ютер генерує ШІМ-сигнал, який повертає вал сервоприводу на заданий кут, відкриваючи замок. Паралельно власник отримує інформаційне повідомлення про успішне відкриття дверей. Через попередньо запрограмований проміжок часу (затримку) система автоматично повертає сервопривід у початкове положення (закриває замок) і знову переходить у режим очікування до наступного натискання кнопки. Такий жорсткий послідовний алгоритм гарантує надійність системи та унеможливлює обхід безпеки при використанні лише одного з біометричних параметрів.

3.2 Розробка програмного забезпечення

Програмне забезпечення системи реалізовано мовою Python 3 завдяки потужній екосистемі бібліотек для машинного навчання, комп'ютерного зору та зручній роботі з апаратними інтерфейсами.

Архітектура коду побудована за строгим модульним принципом із розподілом підзадач на незалежні функції. Такий підхід, разом із застосуванням перевірених патернів проєктування, суттєво спрощує налагодження, мінімізує ризики критичних збоїв, полегшує раннє виявлення помилок та забезпечує масштабованість функціоналу системи [18].

Загальна структурна схема програмного забезпечення, наведена на рисунку 3.2, наочно відображає взаємодію всіх компонентів коду.

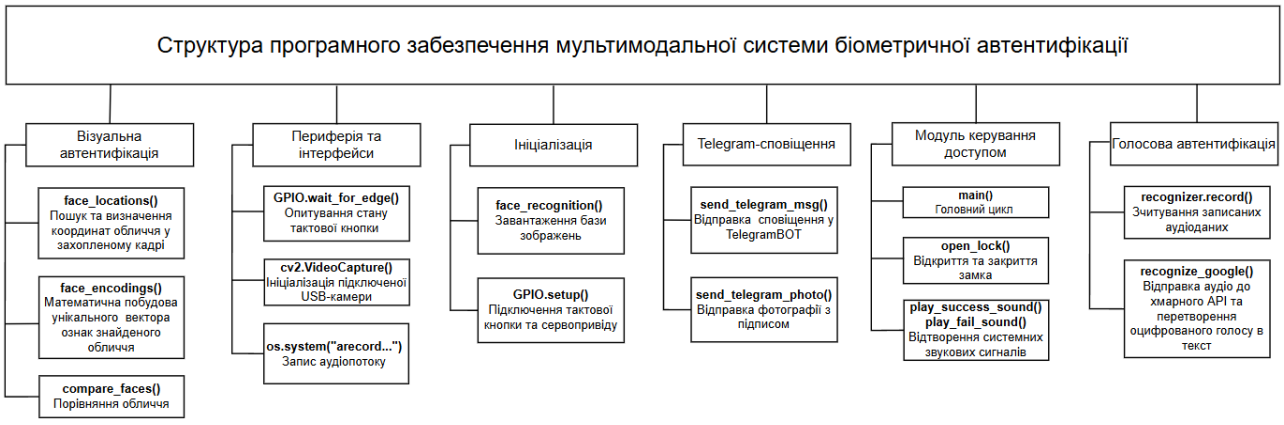

Рисунок 3.2 – Структура програмного забезпечення

Архітектура системи побудована за модульним принципом і поділяється на шість основних блоків:

- ініціалізація системи;
- керування периферією та інтерфейсами;
- візуальна автентифікація;
- голосова автентифікація;
- головний модуль керування доступом;
- підсистема Telegram-сповіщень.

Такий поділ дозволяє чітко розмежувати процеси збору даних, зокрема опитування кнопки та захоплення відео й аудіо, від процесів їхньої подальшої обробки, таких як аналіз біометричних векторів і керування сервоприводом. Кожен блок виконує свою задачу за допомогою ізольованих спеціалізованих функцій, наприклад face_encodings для розпізнавання обличчя або recognize_google для обробки голосу. Це гарантує незалежне, швидке та стабільне виконання кожного етапу перевірки.

Окремими ізольованими блоками реалізовано логіку генерації точних керуючих ШІМ-сигналів для електромеханічного сервоприводу замка, звукову індикацію поточних статусів доступу та підсистему відправки захищених текстових і фотосповіщень власнику через Telegram Bot API.

3.2.1 Опис використаних бібліотек

Для реалізації заявленого функціоналу в програмі використовується низка спеціалізованих бібліотек (рис. 3.3).

```
import time
import os
import requests
import cv2
import face_recognition
import speech_recognition as sr
import OPi.GPIO as GPIO
```

Рисунок 3.3 – Лістинг коду з підключенням бібліотек

Програмна реалізація підсистеми комп'ютерного зору базується на використанні відкритої бібліотеки cv2 (OpenCV). Цей потужний інструмент застосовується для низькорівневої ініціалізації підключеної USB-камери, безперервного захоплення кадрів відеопотоку в режимі реального часу та їхньої попередньої цифрової обробки. Завдяки оптимізованим алгоритмам, OpenCV забезпечує швидке перетворення колірних просторів, нормалізацію освітленості та масштабування зображень, що є критично важливим етапом підготовки графічних даних перед їх подачею на вхід складних нейромережевих моделей розпізнавання [19].

Для безпосередньої ідентифікації користувача застосовується бібліотека face_recognition, яка є високорівневою обгорткою над потужним C++ фреймворком машинного навчання dlib. Даний програмний модуль відповідає за первинну локалізацію обличчя на кадрі за допомогою алгоритмів HOG або згорткових нейронних мереж. Після виявлення обличчя бібліотека будує унікальний 128-мірний вектор ознак (face encoding), що описує його ключові антропометричні характеристики. Порівняння цього вектора з попередньо збереженим еталонним зразком дозволяє системі з високою математичною точністю підтверджувати або відхиляти авторизацію особи, зводячи до мінімуму ймовірність хибного допуску сторонніх [20].

Модуль голосової верифікації реалізовано за допомогою бібліотеки speech_recognition, яка забезпечує комплексну роботу з аудіоданими на високому рівні абстракції. Вона виконує функції захоплення звукового потоку із зовнішнього мікрофона, його оцифрування, автоматичного відсікання тиші та фонових шумів. Найважливішою функцією бібліотеки є забезпечення зручної взаємодії з хмарними або локальними API для перетворення вимовних фраз у машинний текст (Speech-to-Text). Використання цієї бібліотеки гарантує стабільне та швидке декодування вимовленого пароля, що слугує другим незалежним фактором автентифікації [21].

Фізична взаємодія програмного забезпечення з підключеними електронними компонентами здійснюється через спеціалізовану бібліотеку

OPi.GPIO [22]. Цей модуль надає прямий програмний доступ до портів вводу-виводу (GPIO), дозволяючи системі миттєво реагувати на апаратні переривання, викликані натисканням тактової кнопки активації. Крім того, саме ця бібліотека відповідає за генерацію точних керуючих сигналів широтно-імпульсної модуляції (ШІМ), які необхідні для правильного позиціонування валу сервоприводу та імітації роботи фізичного замка.

Інтеграція системи безпеки з мережевими сервісами та механізмами віддаленого інформування власника реалізується за допомогою бібліотеки requests. Даний інструмент стандарту де-факто у мові Python використовується для виконання синхронних HTTP-запитів до серверів Telegram Bot API. Завдяки цій бібліотеці програма здатна оперативно формувати та відправляти захищені POST-запити, які містять текстові сповіщення про статуси доступу (успішна авторизація чи спроба злому), а також прикріплені фотографії невідомих осіб, зафіксованих камерою, забезпечуючи надійний віддалений моніторинг об'єкта.

Крім того, у коді реалізовано спеціальний блок (на базі бібліотеки ctypes), який приховує системні попередження драйвера ALSA [12], що часто виникають у Linux-системах під час ініціалізації звукових пристроїв, забезпечуючи чистий вивід у консоль.

3.2.2 Підпрограма голосової автентифікації

Першим етапом процедури біометричної перевірки виступає захоплення та розпізнавання голосової команди. Відповідна програмна логіка повністю інкапсульована у функції voice_check() (рис. 3.4). Слід зазначити, що пряма взаємодія з апаратними аудіопристроями безпосередньо з середовища Python часто викликає системні конфлікти, пов'язані з політикою прав доступу (зокрема, постійною вимогою запуску скриптів із правами суперкористувача sudo). Для усунення цих обмежень та забезпечення стабільної роботи мікрофона, процес захоплення звукового потоку реалізовано шляхом делегування цієї задачі нативній консольній утиліті arecord. Її прямий

системний виклик виконується на базовому рівні операційної системи Linux, що дозволяє ефективно працювати з аудіодрайверами та гарантує надійний запис аудіоданих без порушення загальної архітектури безпеки програмного комплексу.

```
def voice_check():
    """Апаратний запис звуку (обхід блокувань sudo) та розпізнавання"""
    print("\n--- Етап 1: Голосова перевірка ---")
    print("Скажіть пароль (запис триватиме 4 секунди)...")

    record_cmd = "arecord -D plughw:2,0 -d 4 -c 1 -f S16_LE -r 44100 -q /tmp/voice_record.wav"
    os.system(record_cmd)

    print("Обробка аудіо...")
    try:
        with sr.AudioFile("/tmp/voice_record.wav") as source:
            audio = recognizer.record(source)
            text = recognizer.recognize_google(audio, language="uk-UA").lower()

            print(f"Система почула: ,,{text},,")
            return text

    except sr.UnknownValueError:
        print("Не вдалося розпізнати слова (можливо тиша).")
        return None
    except sr.RequestError:
        print("Помилка сервісу розпізнавання (немає інтернету?).")
        return None
    except Exception as e:
        print(f"Апаратна помилка файлу: {e}")
        return None
```

Рисунок 3.4 – Лістинг коду функції voice_check()

Апаратний запис триває 4 секунди з частотою дискретизації 44100 Гц, після чого файл зберігається у тимчасову директорію /tmp/voice_record.wav. Далі аудіофайл завантажується модулем speech_recognition [21], який відправляє його на обробку рушію Google Speech Recognition із зазначенням української мови розпізнавання (uk-UA).

У разі успішного розпізнавання функція повертає текстовий рядок, який у головному циклі порівнюється із заданим ключовим словом.

3.2.3 Підпрограма візуальної автентифікації

Другий етап автентифікації виконується підпрограмою face_check() (рис. 3.5). Вона активує підключену USB-камеру, захоплює один кадр, зберігає

його для можливої відправки у Telegram та проводить біометричний аналіз.

```
def face_check(known_face_encoding):
    """Захоплює кадр, зберігає його та перевіряє обличчя"""
    print("\n--- Етап 2: Запуск камери для сканування обличчя ---")
    video_capture = cv2.VideoCapture(0)
    time.sleep(1)

    ret, frame = video_capture.read()
    video_capture.release()

    if not ret:
        print("Помилка доступу до камери.")
        return False, None

    photo_path = "/tmp/captured_face.jpg"
    cv2.imwrite(photo_path, frame)

    small_frame = cv2.resize(frame, (0, 0), fx=0.5, fy=0.5)
    rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)

    face_locations = face_recognition.face_locations(rgb_small_frame)
    face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)

    if not face_encodings:
        print("Обличчя не знайдено в кадрі.")
        return False, photo_path

    match = face_recognition.compare_faces([known_face_encoding], face_encodings[0], tolerance=0.5)
    return match[0], photo_path
```

Рисунок 3.5 – Лістинг коду функції face_check()

Для пришвидшення роботи алгоритму на апаратному забезпеченні мікрокомп'ютера, розмір захопленого кадру програмно зменшується вдвічі (fx=0,5, fy=0,5) перед передачею його до нейромережі. Функція face_encodings вилучає біометричні ознаки знайденого обличчя, а compare_faces зіставляє їх із попередньо завантаженим еталоном власника (з допуском tolerance=0,5, що є оптимальним балансом між строгістю перевірки та КХВ) [20].

3.2.4 Головний цикл програми та керування сервоприводом

Головна функція main() ініціалізує систему, завантажує еталонне фото власника у пам'ять та переводить мікрокомп'ютер у режим очікування. Для економії ресурсів застосовано функцію GPIO.wait_for_edge(), яка призупиняє виконання програми до моменту реєстрації спадаючого фронту (FALLING) на піні, до якого підключена кнопка.

Замикання кнопки запускає послідовну перевірку voice_check() та face_check(). Лише за умови успішного проходження обох перевірок викликається підпрограма open_lock() (рис. 3.6).

```
def open_lock():
    """Імітація відкриття замка сервоприводом (Програмний ШІМ 1000мкс - 2000мкс)"""
    print("\n--- Етап 3: Відкриття замка (сервопривід) ---")
    try:
        print("Відкриття замка (імпульс 2000 мкс)...")
        # 50 циклів = 1 секунда (50 * 20мс)
        for _ in range(50):
            GPIO.output(SERVO_PIN, GPIO.HIGH)
            time.sleep(0.002)   # HIGH на 2 мілісекунди
            GPIO.output(SERVO_PIN, GPIO.LOW)
            time.sleep(0.018)   # LOW на 18 мілісекунд

        time.sleep(1.5) # Тримаємо двері відкритими 1.5 секунди

        print("Закриття замка (імпульс 1000 мкс)...")
        for _ in range(50):
            GPIO.output(SERVO_PIN, GPIO.HIGH)
            time.sleep(0.001)   # HIGH на 1 мілісекунду
            GPIO.output(SERVO_PIN, GPIO.LOW)
            time.sleep(0.019)   # LOW на 19 мілісекунд

    except Exception as e:
        print(f"Помилка сервоприводу: {e}")
```

Рисунок 3.6 – Лістинг коду функції open_lock()

Через відсутність вільного апаратного ШІМ-генератора на обраному піні, керування сервоприводом SG90 [13] реалізовано методом програмної генерації широтно-імпульсної модуляції (Software PWM). Для відкриття замка генерується імпульс тривалістю 2000 мкс (2 мс HIGH / 18 мс LOW), а для закриття — 1000 мкс (1 мс HIGH / 19 мс LOW).

Після виконання процедури відкриття та закриття система завершує цикл і знову переходить у стан глибокого очікування апаратного переривання від кнопки, що гарантує безперервну та стабільну роботу комплексу.

3.3 Реалізація системи сповіщень за допомогою Telegram-бота

Важливою складовою сучасної системи контролю доступу є можливість дистанційного моніторингу та оперативного інформування власника про події на об'єкті. Замість розробки окремого мобільного застосунку, що потребує підтримки власних серверів та складного процесу розгортання, у даній роботі прийнято рішення інтегрувати систему сповіщень на базі популярного месенджера Telegram. Це дозволяє забезпечити миттєву доставку як текстових тривожних повідомлень, так і мультимедійних даних (фотографій зловмисників) з мінімальними затримками. Для забезпечення стабільної та швидкої передачі тривожних сповіщень через мережу Інтернет необхідно враховувати принципи оптимізації маршрутизації даних, особливо в умовах підключення пристрою до великих мереж зі складною топологією [23].

3.3.1 Опис Telegram API та процесу реєстрації бота

Взаємодія мультимодальної системи з месенджером здійснюється через офіційний HTTP-інтерфейс Telegram Bot API. Цей інтерфейс дозволяє програмно надсилати запити на сервери Telegram, які потім пересилають повідомлення на мобільний пристрій цільового користувача [24].

Для того щоб система могла відправляти повідомлення, необхідно було створити унікального віртуального користувача — бота. Процес реєстрації здійснюється безпосередньо у додатку Telegram за допомогою спеціального системного інструменту BotFather. Алгоритм реєстрації складається з наступних кроків:

1) Ініціалізація створення нового бота командою /newbot.

2) Вибір публічного імені (наприклад, "SmartLock Bot") та унікального ідентифікатора (username), який обов'язково має закінчуватися на "bot".

3) Отримання унікального API Token — довгого рядка символів (наприклад, 8825276158:AAEH...), який виступає в ролі секретного ключа для авторизації запитів від імені мікрокомп'ютера [24].

Окрім токена бота, для адресної відправки повідомлень власнику необхідно знати його унікальний числовий ідентифікатор чату (Chat ID). Його було отримано шляхом надсилання повідомлення спеціальному сервісному боту (наприклад, @userinfobot), який повертає системний ідентифікатор користувача.

Отримані значення TELEGRAM_TOKEN та CHAT_ID зберігаються у вигляді глобальних констант на початку головного скрипта системи.

3.3.2 Реалізація відправки тривожних повідомлень та фотофіксації

Програмна реалізація взаємодії з Telegram API написана мовою Python з використанням бібліотеки requests, яка дозволяє виконувати синхронні POST-запити до серверів месенджера [25].

Логіка сповіщень розділена на дві окремі функції: send_telegram_msg() (рис. 3.7) для швидких текстових статусів та send_telegram_photo() для відправки зображень із підписом.

```
def send_telegram_msg(message):
    """Відправка звичайного текстового сповіщення у Telegram"""
    url = f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendMessage"
    payload = {'chat_id': CHAT_ID, 'text': message}
    try:
        requests.post(url, data=payload, timeout=5)
    except Exception:
        pass
```

Рисунок 3.7 – Лістинг коду функції send_telegram_msg()

Функція відправки текстових повідомлень формує URL-адресу із використанням токена бота та передає корисне навантаження (payload) у форматі JSON, що містить ідентифікатор отримувача та сам текст.

Особлива увага приділена обробці мережевих помилок. Використання блоку try...except та параметра timeout=5 гарантує, що у разі тимчасової відсутності інтернет-з'єднання або недоступності серверів Telegram, головний цикл системи не зависне на етапі відправки, а продовжить свою роботу

(наприклад, дозволить відкрити замок або перейти у режим очікування).

Для реалізації фотофіксації несанкціонованого доступу використовується метод API sendPhoto. Під час невдалої візуальної автентифікації система зберігає захоплений камерою кадр у тимчасовий файл (наприклад, /tmp/captured_face.jpg). Функція send_telegram_photo() (рис. 3.8) зчитує цей файл у бінарному режимі та відправляє його на сервер у складі багаточастинного (multipart) POST-запиту.

```
def send_telegram_photo(caption, photo_path):
    """Відправка фотографії з підписом у Telegram"""
    url = f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendPhoto"
    try:
        with open(photo_path, 'rb') as photo:
            payload = {'chat_id': CHAT_ID, 'caption': caption}
            files = {'photo': photo}
            requests.post(url, data=payload, files=files, timeout=10)
    except Exception as e:
        print(f"[Помилка мережі] Не вдалося відправити фото: {e}")
```

Рисунок 3.8 – Лістинг коду функції send_telegram_photo()

У головному циклі програми ці функції викликаються на кожному етапі авторизації. Наприклад, при ініціалізації пристрою власник отримує повідомлення "Система увімкнена". Якщо зловмисник намагається пройти розпізнавання, система миттєво формує та надсилає повідомлення із прикріпленою фотографією особи, яка стоїть перед дверима. Слід зазначити, що впровадження подібних рішень у сфері фізичного контролю доступу вимагає системного керування проєктами в галузі кібербезпеки. Це передбачає не лише апаратний захист пристрою, але й суворе забезпечення конфіденційності локально збережених біометричних даних та використання виключно HTTPS для передачі тривожних сповіщень [26].

3.4 Тестування системи в реальних умовах

Завершальним етапом розробки мультимодальної системи біометричної автентифікації є перевірка її працездатності, стабільності та точності в умовах, наближених до реальної експлуатації. Тестування дозволяє виявити потенційні вразливості, оцінити швидкість реакції апаратно-програмного комплексу та переконатися у коректній роботі системи сповіщень.

Процес перевірки працездатності програмного забезпечення базується на підходах модульного тестування ізольованих компонентів (робота з камерою, мікрофоном, Telegram API), що концептуально наближено до принципів тестування мікросервісних архітектур [27].

Для проведення випробувань усі апаратні компоненти (мікрокомп'ютер Raspberry Pi 4, USB-камера, мікрофон, тактова кнопка та сервопривід) були зібрані у єдиний тестовий стенд (рис. 3.9).

Рисунок 3.9 – Зовнішній вигляд зібраного прототипу мультимодальної системи

Першим етапом тестування є перевірка штатного режиму роботи системи під час успішної авторизації власника, що дозволяє підтвердити коректність спрацьовування обох біометричних факторів. Користувач підходить до системи та натискає тактову кнопку для пробудження пристрою.

У відповідь система активує мікрофон і просить вимовити пароль. Власник чітко вимовляє ключове слово "грім". Система розпізнає слово, відтворює звуковий сигнал підтвердження та фіксує успішне проходження першого етапу. Після цього одразу активується камера, а власник дивиться в об'єктив. Алгоритм OpenCV успішно знаходить обличчя в кадрі, вилучає вектор ознак та зіставляє його з еталоном. У фіналі алгоритм підтверджує збіг і відправляє у Telegram повідомлення про успішний вхід та відкритий замок, після чого генерує керуючий сигнал ШІМ для сервоприводу (рис. 3.10). Виконавчий механізм виконує поворот валу на півтори секунди, а потім повертається у закрите положення. Система успішно переходить у режим сну.

Рисунок 3.10 – Успішна авторизація

Наступним кроком перевіряється реакція системи на спробу доступу з хибним голосовим паролем або повною відсутністю звуку. Користувач натискає кнопку активації та на запит пароля вимовляє неправильне слово або мовчить. Модуль розпізнавіання мови миттєво фіксує хибне слово або

відсутність звуку. Система негайно перериває процес авторизації, при цьому наступний етап візуальної перевірки обличчя навіть не запускається. Пристрій одразу відтворює звуковий сигнал помилки та відправляє тривожне повідомлення у Telegram (рис. 3.11).

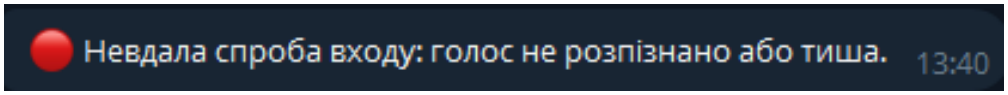

Рисунок 3.11 – Відмова у доступі через неправильний голосовий пароль

Заключним етапом є перевірка реакції системи на спробу доступу сторонньої особи, якій відомий правильний голосовий пароль. Стороння особа без наявності обличчя в базі даних натискає кнопку та вимовляє правильне кодове слово. Система успішно підтверджує перший етап. Далі активується камера, стороння особа дивиться в об'єктив. Алгоритм порівняння фіксує відсутність збігу з еталоном власника. Система відмовляє у доступі. У цей момент камера робить знімок зловмисника, зберігає його та миттєво відправляє у Telegram разом із відповідним підписом про нерозпізнане обличчя (рис. 3.12). Сервопривід не активується, і замок залишається повністю закритим.

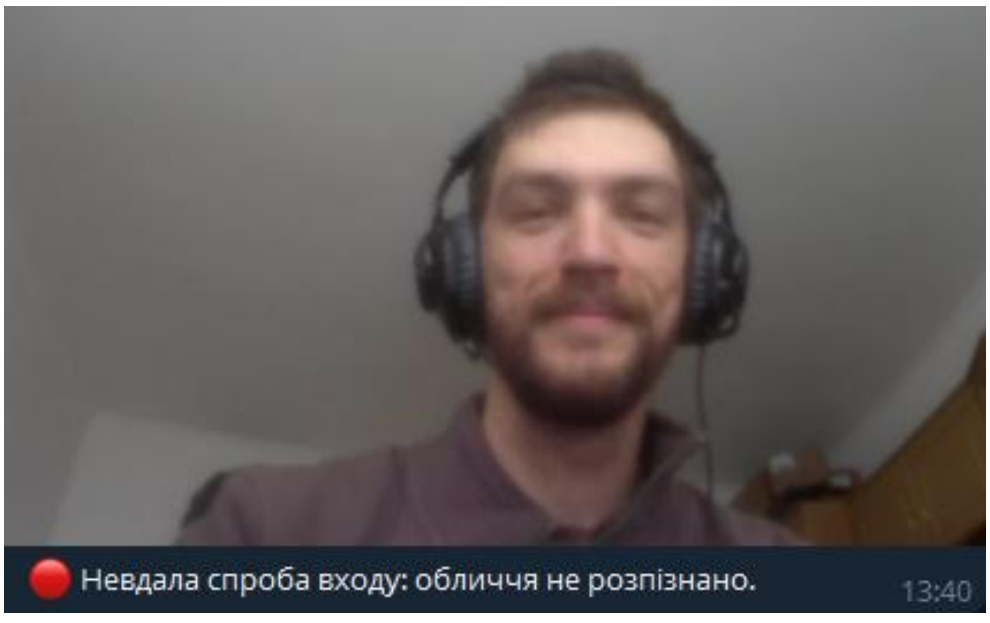

Рисунок 3.12 – Відмова у доступі через нерозпізнане обличчя

Проведені експерименти підтвердили повну працездатність розробленої системи. Апаратне переривання від кнопки працює без хибних спрацьовувань, забезпечуючи надійний вихід зі сплячого режиму. Мультимодальний підхід довів свою ефективність, оскільки система блокує доступ при відхиленні хоча б одного з параметрів. Швидкість виконання повного циклу в середньому становить близько шести секунд. Інтеграція з Telegram API функціонує бездоганно і забезпечує миттєву доставку фотографій та текстових сповіщень. Загалом розроблений комплекс повністю відповідає всім поставленим вимогам технічного завдання.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Можливість виникнення статичної електрики та заходи боротьби з нею

Питання захисту від статичної електрики набуває критичної ваги на етапі апаратного збирання, налаштування та технічного обслуговування мультимодальної системи біометричної автентифікації. Хоча мікрокомп’ютер та підключені до нього периферійні пристрої живляться від абсолютно безпечної для людини напруги постійного струму (рівні логіки 3,3 В та живлення 5 В), накопичений на тілі розробника статичний заряд становить величезну загрозу для високотехнологічної електроніки [28]. Згідно з моделлю людського тіла (HBM), електрична ємність людини становить приблизно 100–250 пФ. Внаслідок трибоелектричного ефекту при терті одягу чи ходінні по діелектричному покриттю на тілі може накопичуватися електростатичний потенціал від 3000 В до 35000 В. При розряді на контакти мікросхем, які виготовлені за надзвичайно чутливою субмікронною КМОН-технологією (межа пробою яких часто становить лише 30–100 В), цей потенціал миттєво пробиває надтонкі шари діелектрика товщиною у кілька нанометрів. Це призводить до теплового руйнування p-n переходів, незворотного виходу обладнання з ладу або виникнення прихованих мікродефектів, що спричиняють раптові програмно-апаратні збої [29].

Для повної нейтралізації цих ризиків необхідно застосовувати такі індивідуальні та колективні засоби захисту:

–використання антистатичного браслета, підключеного до шини заземлення через струмообмежувальний резистор номіналом 1 МОм, який гарантує плавне стікання заряду та обмежує струм через тіло людини

безпечним рівнем менше 0,5 мА навіть у разі випадкового торкання фазного проводу 220 В [23];

– носіння спеціалізованого одягу (питомий поверхневий опір тканини не повинен перевищувати 10^7 Ом) та антистатичного взуття, що унеможливлює накопичення заряду при терті [30];

–обладнання робочого столу струморозсіювальним антистатичним килимком із поверхневим опором у діапазоні 10^6–10^9 Ом;

–використання професійних паяльних станцій із гальванічною розв'язкою та заземленим жалом (опір між жалом та контуром заземлення будівлі не має перевищувати 2 Ом) [29];

–застосування локальних іонізаторів повітря за умов підвищеної небезпеки для нейтралізації об'ємних зарядів на пластикових корпусах периферії.

Серед організаційно-технічних заходів ключовими є такі правила поводження з компонентами:

–зберігання тимчасово не задіяних модулів у металізованих антистатичних пакетах із багатошаровим екрануванням;

– утримання друкованих плат виключно за діелектричні текстолітові торці для уникнення контакту зі струмопровідними лініями [28];

– підтримання відносної вологості повітря в приміщенні на рівні 40–60% (при падінні вологості нижче 30 % напруга статики зростає експоненціально, досягаючи критичних 15000 В від звичайного тертя синтетики);

– підключення чи відключення периферійних пристроїв виключно після стовідсоткового фізичного знеструмлення системи та зняття залишкової напруги з конденсаторів [30].

Не менш важливим аспектом комплексного захисту від електростатичних розрядів є регулярний аудит та перевірка працездатності застосовуваних засобів індивідуального та колективного захисту. Усі антистатичні браслети, кабелі заземлення та струморозсіювальні килимки з часом можуть втрачати свої властивості через механічне зношення або

пошкодження внутрішніх струмопровідних волокон. Тому перед початком кожного етапу робіт з інтеграції апаратної частини необхідно проводити інструментальний контроль опору ланцюга за допомогою спеціалізованих тестерів-мегомметрів. Опір має знаходитися у суворо визначених межах (від 1 до 10 МОм), що гарантує ефективне стікання накопиченого заряду та одночасно забезпечує безпеку оператора [29].

Особливу увагу слід приділяти захисту найбільш вразливих аналого-цифрових вузлів розробленого комплексу, якими є світлочутлива матриця підключеної USB-камери та вбудований аудіокодек мікрофона. Навіть мікроскопічні електростатичні розряди, які не призводять до повного виходу цих модулів з ладу, здатні викликати незворотну деградацію їхньої напівпровідникової структури. На практиці це проявляється у вигляді появи "битих" пікселів на зображенні або суттєвого підвищення рівня фонового теплового шуму в аудіотракті [28]. Оскільки алгоритми машинного навчання вкрай чутливі до чистоти вхідних даних, такі приховані апаратні дефекти можуть різко знизити загальну точність розпізнавання обличчя та голосу, зводячи нанівець надійність усієї системи безпеки [30].

4.2 Захист електрообладнання від короткого замикання, перенавантаження

Оскільки розроблена система контролю доступу призначена для безперервної цілодобової роботи (24/7), захист електрообладнання від коротких замикань та термічних перенавантажень є критичним завданням пожежної безпеки [30]. Головний процесор працює в умовах екстремального теплового навантаження, генеруючи до 10–15 Вт теплової потужності під час обробки нейромережевих відеопотоків, а електромеханічний замок створює значні індуктивні навантаження (пускові струми двигуна можуть стрибкоподібно зростати до 500–750 мА).

Для теплового захисту та температурного моніторингу застосовуються наступні інженерні рішення:

– встановлення масивних мідних або алюмінієвих радіаторів через теплопровідні термоінтерфейси з коефіцієнтом теплопровідності не менше 3–5 Вт/(м·К) [28];

– використання активного міні-кулера (5 В, 0,1 А) для створення примусового повітряного потоку та ефективного розсіювання тепла;

– монтаж системи у корпусі з негорючого пластику (з індексом горючості V-0) або алюмінію з достатньою площею конвекційних отворів;

– налаштування операційної системи на апаратне зниження тактової частоти (троттлінг) при досягненні кристалом температури 80°C та примусове аварійне відключення живлення при перегріві понад 85°C.

Крім того, оскільки розроблений комплекс має підключення до глобальної мережі (Telegram-бот), його архітектура тісно переплітається з концепцією Інтернету речей (IoT). Інтеграція подібних IoT-пристроїв у загальну інфраструктуру дозволяє реалізувати додаткові сценарії безпеки об'єкта, зокрема постійний автоматизований моніторинг температурних аномалій, що слугує превентивним заходом для забезпечення пожежної безпеки приміщення [31].

Апаратний захист ліній живлення та сигнальних кіл гарантується такими заходами:

– використання якісного промислового блоку живлення номіналом 5 В із забезпеченням стабільної сили струму 3–4 А, який оснащено вбудованими модулями OVP (захист від перенапруги понад 5.25 В), OCP (захист від перевантаження по струму) та SCP (захист від короткого замикання) [29];

– живлення індуктивних механізмів (замка) через окремі силові лінії для уникнення просідання напруги на платі нижче критичних 4,75 В [28];

– передача керуючого ШІМ-сигналу через оптопару для гальванічної ізоляції портів мікрокомп'ютера від зворотних ЕРС;

– інтеграція в лінію живлення електродвигуна самовідновлюваного полімерного запобіжника (PTC) з порогом спрацьовування 1 А [30].

Апаратний захист ліній живлення та сигнальних кіл гарантується використанням якісного промислового блоку живлення номіналом 5 В із забезпеченням стабільної сили струму 3–4 А, який оснащено вбудованими модулями OVP, OCP та SCP. Загалом, застосування науково обґрунтованих методів та засобів підвищення надійності комп'ютеризованих модулів живлення гарантує безперебійне функціонування мікрокомп'ютера Raspberry Pi навіть за умов сильних перешкод та нестабільності вхідної електромережі, що є обов'язковою умовою для СКУД [32].

Оскільки для повністю автономної та незалежної роботи системи (на випадок знеструмлення будівлі) використовується резервне джерело живлення на основі літій-іонних акумуляторів (Power Bank), надзвичайно важливою вимогою є використання моделей із вбудованими інтелектуальними контролерами балансування комірок. Ефективні алгоритми балансування заряду запобігають небезпечному перезаряду або глибокому розряду окремих елементів батареї, що є критично важливим фактором для уникнення їхнього термічного розгону та самозаймання під час тривалої безперервної експлуатації системи [33].

У сфері електромонтажних заходів та пожежної профілактики висуваються наступні суворі вимоги:

– використання зовнішніх кабелів живлення із перерізом мідних жил не менше 0,5–0,75 мм², що розраховані на довготривале навантаження до 5 А без нагрівання ізоляції понад 65°C;

– прокладання рухомої проводки на дверному полотні у гнучких кабель-каналах для унеможливлення перетирання ізоляції та виникнення замикання на заземлений металевий каркас [29];

– герметизація всіх відкритих ділянок паяних з'єднань діелектричними термоусадочними трубками з клейовою основою;

– наявність у приміщенні легкодоступних вуглекислотних (ВВК-2) або порошкових вогнегасників, які є діелектриками та дозволені для гасіння електроустановок з напругою до 1000 В [30];

– гасіння можливого займання виключно після стовідсоткового знеструмлення всієї апаратури від мережі 220 В через ввідний автоматичний вимикач [28].

Окрім суто апаратних засобів захисту, високий рівень надійності системи забезпечується впровадженням апаратно-програмних механізмів контролю за станом живлення та обчислювального процесу. Зокрема, у мікрокомп'ютері доцільно активувати апаратний сторожовий таймер, який на низькому рівні безперервно відстежує працездатність операційної системи [28]. У разі зависання центрального процесора через короткочасні провали напруги або вплив потужних електромагнітних завад від роботи електродвигуна, сторожовий таймер автоматично ініціює перезавантаження плати. Це запобігає тривалому перебуванню процесора у стані максимального енергоспоживання при програмному зависанні, що суттєво знижує ризик неконтрольованого перегріву кристала [30].

Важливою складовою довгострокової пожежної профілактики є регулярне технічне обслуговування змонтованого комплексу. Оскільки система працює у реальних умовах, її радіатори охолодження та вентиляційні отвори з часом неминуче забруднюються дрібнодисперсним пилом, який є чудовим теплоізолятором та може сприяти перегріву компонентів [29]. Періодичне очищення апаратної частини від пилу, а також візуальна ревізія стану гнучких шлейфів і силових кабелів у місцях їхнього згину на дверних петлях, дозволяють завчасно виявити потенційні місця короткого замикання. Такий превентивний підхід гарантує, що розроблена система не лише безпечно функціонуватиме на етапі запуску, але й збереже високі показники інженерної відмовостійкості протягом усього терміну служби [28].

ВИСНОВКИ

У ході виконання кваліфікаційної роботи була розроблена комп’ютерна мультимодальна системи біометричної автентифікації на базі одноплатного мікрокомп'ютера. Впровадження розробленого апаратно-програмного комплексу дозволяє значно підвищити рівень надійності та автоматизації систем контролю фізичного доступу до приміщень.

У результаті виконання роботи досягнуто наступних результатів:

1) Проведено аналіз предметної області та технічного завдання. Виявлено вразливості класичних та однофакторних біометричних систем до методів підміни. Обґрунтовано доцільність застосування мультимодального підходу, що поєднує перевірку геометрії обличчя та голосового пароля. Сформульовано чіткі функціональні та експлуатаційні вимоги до системи.

2) Спроектовано апаратну частину системи. В якості центрального обчислювального ядра обрано високопродуктивний мікрокомп’ютер. Підібрано та інтегровано необхідну периферію: USB-камеру, мікрофон із підтримкою I2S, сервопривід для керування механізмом замка та тактову кнопку. Розроблено структурну та електричну принципову схеми пристрою, що забезпечують його надійну роботу та підтримку апаратного режиму енергозбереження.

3) Розроблено програмне забезпечення мовою Python. Створено ефективний алгоритм двофакторної автентифікації. За допомогою бібліотек комп'ютерного зору (OpenCV, face_recognition) та розпізнавання мовлення (SpeechRecognition) реалізовано локальну обробку біометричних даних, що гарантує високий рівень приватності.

4) Реалізовано підсистему дистанційного моніторингу. Виконано інтеграцію розробленого комплексу з Telegram API. Налаштовано автоматичну відправку текстових сповіщень про стан доступу та миттєву фотофіксацію зловмисників у разі невдалої спроби авторизації, що забезпечує постійний віддалений контроль за безпекою об'єкта.

5) Проведено успішне тестування прототипу. Експериментальна перевірка системи в реальних умовах підтвердила безперебійну роботу всіх складових. Зафіксовано високу швидкість обробки даних (близько 6-8 секунд на повний цикл перевірки) та надійне блокування доступу при розбіжності хоча б одного з біометричних параметрів.

6) Опрацьовано питання охорони праці та безпеки життєдіяльності. Визначено ключові заходи безпеки під час проектування, монтажу та тривалої експлуатації електронного обладнання. Розроблено рекомендації щодо захисту пристрою від статичної електрики, перегріву процесора та короткого замикання.

Таким чином, поставлена мета кваліфікаційної роботи досягнута в повному обсязі. Розроблена система повністю відповідає технічним вимогам, відрізняється високою надійністю, зручністю використання та є готовою до впровадження у сучасну інфраструктуру систем безпеки і технологій «розумного будинку».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жаровський Р.О., Луцик Н.С., Осухівська Г.М., Паламар А.М., Тиш Є.В. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія». Тернопіль: ТНТУ, 2024. 39 с.

2. Jain A. K., Ross A., Nandakumar K. Introduction to Biometrics. New York : Springer, 2011. 313 p.

3. Shabliy N., Lupenko S., Lutsyk N., Yasniy O., Malyshevska O. Keystroke dynamics analysis using machine learning methods. Applied Computer Science. 2021. Vol. 17, No. 4. P. 75-83.

4. ISO/IEC 19795-1:2021. Information technology — Biometric performance testing and reporting — Part 1: Principles and framework. Geneva : International Organization for Standardization, 2021. 46 p.

5. ProFace X Series: Advanced Face & Palm Recognition Terminal. ZKTeco Official Website. URL: https://www.zkteco.com/en/product_detail/ProFaceX (дата звернення: 25.05.2026).

6. MinMoe Face Recognition Terminals. Hikvision Official Website. URL: https://www.hikvision.com/en/products/Access-Control-Products/Face-Recognition-Terminals/ (дата звернення: 25.05.2026).

7. Xiaomi Smart Door Lock: Product Specifications. Mi Global Home. URL: https://www.mi.com/global/smart-home/smart-door-lock (дата звернення: 25.05.2026).

8. Kharchenko A., Bodnarchuk I., Yatcysyn V. The Method for Comparative Evaluation of Software Architecture with Accounting of Trade-offs. American Journal of Information Systems. 2014. Vol. 2, No. 1. P. 20-25.

9. Raspberry Pi 4 Computer Model B Datasheet. Raspberry Pi (Trading) Ltd., 2019. 11 p.

10. Луцків А., Лупенко С., Пасічник В. Паралельні та розподільнені обчислення. Навчальний посібник. Львів: Видавництво «Магнолія 2006», 2024. 566 с.

11. USB Video Class (UVC) Specification 1.5. USB Implementers Forum, 2012. URL: https://www.usb.org/document-library/video-class-v15-document-set (дата звернення: 25.05.2026).

12. ALSA project - the C library reference. Advanced Linux Sound Architecture. URL: https://www.alsa-project.org/ (дата звернення: 25.05.2026).

13. SG90 9g Micro Servo Motor Datasheet. Tower Pro, 2020. 4 p.

14. RK3399 Technical Reference Manual. Rockchip Electronics Co., Ltd., 2017. URL: http://rockchip.wikidot.com/rk3399 (дата звернення: 25.05.2026).

15. Button and Switch Interfacing Techniques. Electronics Tutorials, 2023. URL: https://www.electronics-tutorials.ws/switch/switch_bounce.html (дата звернення: 25.05.2026).

16. Ubuntu Server for ARM. Canonical Ltd., 2024. URL: https://ubuntu.com/download/server/arm (дата звернення: 25.05.2026).

17. Python 3.12 Reference Manual. Python Software Foundation, 2024. URL: https://docs.python.org/3/reference/ (дата звернення: 25.05.2026).

18. Yatsyshyn V., Pastukh O., Kukharska V., Palamar A., Kulikov S. Method and tool of detecting software architecture patterns in the process of computer systems development. CEUR Workshop Proceedings, 4th International Workshop on Information Technologies (ITTAP 2024), Ternopil, Ukraine. 2024. Vol. 3896. P. 12-24.

19. OpenCV-Python Tutorials. OpenCV Documentation, 2024. URL: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html (дата звернення: 25.05.2026).

20. Geitgey A. Face Recognition: The world's simplest facial recognition api for Python. GitHub, 2023. URL: https://github.com/ageitgey/face_recognition (дата звернення: 25.05.2026).

21. SpeechRecognition Library Documentation. PyPI, 2023. URL: https://pypi.org/project/SpeechRecognition/ (дата звернення: 25.05.2026).

22. OPi.GPIO: A drop-in replacement for RPi.GPIO. PyPI, 2022. URL: https://pypi.org/project/OPi.GPIO/ (дата звернення: 25.05.2026).

23. Озарків Т., Жаровський Р. Оптимізація роботи протоколу EIGRP в умовах великих мереж зі складною топологією. Матеріали XII Міжнародної науково-технічної конференції молодих учених. Тернопіль: ТНТУ. 2023. С. 442.

24. Telegram Bot API Documentation. Telegram FZ-LLC, 2024. URL: https://core.telegram.org/bots/api (дата звернення: 25.05.2026).

25. Reitz K. Requests: HTTP for Humans. Python Software Foundation, 2023. URL: https://requests.readthedocs.io/ (дата звернення: 25.05.2026).

26. Stadnyk M., Palamar A. Project management features in the cybersecurity area. Scientific Journal of TNTU, Ternopil, Ukraine, 2022. Vol. 106, No 2. P. 54–62.

27. Свергун С., Жаровський Р. Тестування програмного забезпечення побудованого на мікросервісній архітектурі. Матеріали X науково-технічної конференції ТНТУ. Тернопіль: ТНТУ. 2022. С. 92.

28. Основи охорони праці. / Під ред. Ткачука К.Н., Халімовського Н.О. – К.: Основа, 2006. 448 с.

29. Геврик Є.О. Охорона праці. – К.: Ельга, Ніка-Центр, 2003. 280 с.

30. Мохняк С.М. Безпека життєдіяльності. Навчальний посібник. – Львів: вид. НУ „Львівська політехніка”, 2009. 264 с.

31. Palamar A., Palamar M. Fire Safety Monitoring System Based on Internet of Things. CEUR Workshop Proceedings, 1st International Workshop (CITI 2023), Ternopil. 2023. Vol. 3468. P. 164-172.

32. Palamar A. Methods and means of increasing the reliability of computerized modular uninterruptible power supply system. Scientific Journal of TNTU, Ternopil, 2020. Vol. 99, No 3. P. 133–141.

33. Voloskyi V., Leshchyshyn Y., Romanyshyn N., Palamar A., Tarasenko L. Method and algorithm for efficient cell balancing in the lithium-ion battery control system. CEUR Workshop Proceedings (BAIT 2024), Zboriv. 2024. Vol. 3842. P. 258-267.

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

"Затверджую"

Завідувач кафедри КС

_____ Осухівська Г.М.

" 2 " лютого 2026 р.

КОМП'ЮТЕРНА МУЛЬТИМОДАЛЬНА СИСТЕМА БІОМЕТРИЧНОЇ АВТЕНТИФІКАЦІЇ НА ОСНОВІ RASPBERRY PI

ТЕХНІЧНЕ ЗАВДАННЯ

на _9_ листках

Вид робіт: Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»	«ВИКОНАВЕЦЬ»
Керівник кваліфікаційної роботи	Студент групи СІ-42
_____ к.т.н., доц. Паламар А.М.	_____ Ковалишин В.П.
" 2 " лютого 2026 р.	" 2 " лютого 2026 р.

Тернопіль 2026

1 Загальні відомості

1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: «Комп’ютерна мультимодальна система біометричної автентифікації на основі Raspberry Pi».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.175.00.00

1.2 Виконавець

Студент групи СІ-42, факультету комп’ютерно-інформаційних систем і програмної інженерії, кафедри комп’ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Ковалишин В.П.

1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/9-188 від 24.04.2026 р.)

1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 26.01.2026 р.

Плановий термін завершення виконання кваліфікаційної роботи – 21.06.2026 р.

1.5 Порядок оформлення та пред’явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ISO, ЕСКД, ЕСПД та ДСТУ.

Пред’явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи – наявності пояснювальної записки та графічного матеріалу.

Пред’явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

2 Призначення і цілі створення системи

2.1 Призначення системи

Мультимодальна система біометричної автентифікації призначена для:

– Автоматизованого контролю фізичного доступу до приміщень;

– Двофакторної перевірки користувачів;

– Дистанційного моніторингу та фотофіксації спроб доступу через месенджер Telegram.

2.2 Мета створення системи

Метою кваліфікаційної роботи є розробка апаратно-програмного комплексу мультимодальної біометричної автентифікації на основі високопродуктивного одноплатного мікрокомп'ютера з використанням алгоритмів комп'ютерного зору.

2.3 Характеристика об'єкту

Розроблювана система призначена для використання в системах безпеки розумних будинків, офісних приміщень та зон обмеженого доступу.

3 Вимоги до системи

3.1 Вимоги до системи в цілому

Комп’ютерна мультимодальна система біометричної автентифікації повинна забезпечувати надійний контроль фізичного доступу до приміщень шляхом обов'язкової перевірки двох незалежних факторів: геометрії обличчя та голосового пароля. Головними вимогами до комплексу є висока точність і швидкість розпізнавання у режимі реального часу, енергоефективність (наявність апаратного режиму очікування), відмовостійкість локальної обчислювальної бази (мікрокомп'ютера) та можливість дистанційного моніторингу спроб доступу через месенджер Telegram. Загалом, система повинна бути автономною, захищеною від методів підміни біометрії та зручною в повсякденній експлуатації.

3.1.1 Вимоги до структури та функціонування системи

Система автентифікації повинна складатись з:

– Апаратного комплексу;

– Програмного модуля обробки біометричних даних;

– Підсистеми віддалених сповіщень;

3.1.2 Вимоги до способів та засобів зв’язку між компонентами системи

Основна вимога до засобів інформаційного обміну – це мінімальна затримка при передачі мультимедійних потоків від камери та мікрофона до процесора, а також наявність стабільного інтернет-з'єднання для роботи Telegram API.

3.1.3 Вимоги до режимів функціонування системи

Для системи визначено два основні режими функціонування:

– Режим очікування;

– Активний режим;

У разі виникнення аварійного режиму (наприклад, відсутність інтернету) система повинна локально відмовляти у доступі та повертати механізм замка у закритий стан.

3.1.4 Вимоги по діагностуванню системи

Для діагностування системи використовуються вбудовані інструменти ОС Linux (перевірка статусів служб) та консольне логування процесу виконання Python-скрипта (вивід повідомлень про помилки камер, аудіокодеків або API).

3.1.5 Перспективи розвитку, проектування системи

Дана система може бути розширена шляхом додавання нових біометричних модальностей (наприклад, сканера відбитків пальців), розширення локальної бази користувачів, а також інтеграції з комплексними платформами розумного будинку (наприклад, Home Assistant).

3.2 Показники призначення

Система повинна забезпечувати високу точність розпізнавання: зводити до мінімуму коефіцієнт хибного допуску (FAR) завдяки мультимодальності. Повинна бути передбачена можливість масштабування бази еталонних облич.

3.2.1 Вимоги до надійності

Система повинна автоматично відновлювати свої функції при виникненні наступних ситуацій:

– при збоях в системі електропостачання мікрокомп'ютера;

– при тимчасовій втраті з'єднання з мережею Інтернет;

– при програмних винятках (помилках читання відеопотоку).

3.3 Вимоги до безпеки

Оскільки система працює від безпечної напруги 5 В, ризик ураження струмом мінімальний. Однак блок живлення повинен забезпечувати захист від коротких замикань. Усі струмопровідні контакти GPIO та з'єднання сервоприводу повинні бути надійно ізольовані для запобігання пожежонебезпечних ситуацій.і

3.3.1 Вимоги до експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи

Пристрій повинен експлуатуватися за таких умов мікроклімату:

– температуру повітря в межах від +5ºС до +40ºС;

– відносну вологість повітря до 80% (без утворення конденсату).

Періодичне технічне обслуговування (очищення об'єктива камери, перевірка механіки сервоприводу та стану контактів кнопки) має проводитися не рідше ніж один раз на рік.

3.4 Вимоги до захисту інформації від несанкціонованого доступу

Система повинна забезпечувати високий рівень приватності: обробка біометричних векторів (обличчя) має відбуватися виключно локально, без передачі вихідних зображень на сторонні хмарні сервери розпізнавання. Доступ до операційної системи мікрокомп'ютера повинен бути захищений надійними паролями та протоколом SSH.

3.4.1 Вимоги по збереженню інформації при аваріях

Еталонні фотографії власника та конфігураційні файли повинні зберігатися на незалежній пам’яті пристрою (eMMC або MicroSD). При вимкненні живлення налаштування системи не повинні втрачатися.

3.4.2 Вимоги по стандартизації і уніфікації

Апаратні компоненти (мікрокомп'ютер, блок живлення) повинні мати сертифікати відповідності та безпеки (CE, RoHS). Програмний код має відповідати загальноприйнятим стандартам написання коду (PEP 8 для Python).

3.4.3 Вимоги до функцій (завдань), що виконуються системою:

– вихід з режиму сну за допомогою апаратної кнопки;

– захоплення та розпізнавання голосової команди (пароля);

– захоплення та розпізнавання геометрії обличчя користувача;

– ШІМ-керування електромеханічним замком (сервоприводом);

– миттєва відправка текстових сповіщень та фотографій зловмисників через Telegram-бот.

4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

– пояснювальної записки;

– графічного матеріалу:

а) Структурна схема.

б) Схема електрична принципова.

в) Блок схема алгоритму роботи програми.

г) Результати моделювання системи.

*Примітка: У комплект документації можуть вноситися зміни та доповнення в процесі розробки.

5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка технічного завдання	*26.01 – 02.02*
2	Аналіз технічного завдання, вимог до комп'ютерної системи, та можливих рішень поставленого завдання	*03.02 – 15.02*
3	Розроблення структури, вибір апаратного забезпечення, проєктування комп'ютеризованої системи	*20.04 – 28.04*
4	Реалізація алгоритму, написання програмного забезпечення, моделювання комп'ютерної системи	*29.04 – 09.05*
5	Безпека життєдіяльності, основи охорони праці	*10.05 – 20.05*
6	Оформлення пояснювальної записки і графічного матеріалу	*21.05 – 7.06*
7	Перевірка на академічний плагіат, перевірка керівником та консультантами	*8.06 – 14.06*
8	Попередній захист кваліфікаційної роботи бакалавра	*15.06 – 21.06*
9	Захист кваліфікаційної роботи бакалавра	*23.06.2026*

6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.

Додаток Б

Перелік елементів

Поз. позначення	*Найменування*	*Кіл.*	*Примітка*
	Сервопривід		
M1	Сервопривід SG90	1	
	Кнопка		
S1	TS-103T Кнопка тактова	1	
	Пристрої		
U1	Одноплатний комп’ютер Raspberry Pi 4	1	
A1	Навушники JBL Quantum N100	1	
A2	USB звукова карта Gemix SC-02	1	
A3	USB камера Creality Nebula	1	
GB1	Power Bank	1	

					КС КРБ 123.175.00.00 ПЕ		
Зм.	*Арк.*	*№ докум.*	*Підпис*	*Дата*			
Розробив		*Ковалишин В.П.*			*Комп’ютерна мультимодальна система біометричної автентифікації на основі Raspberry Pi*	*Літ*	*Арк.* 70 / *Аркушів* 1
Перевірив		*Паламар А.М.*					
Резензент		*Матійчук Л.П.*				*ТНТУ, каф. КС, гр. СІ-42*	
Н. Контр.		*Луцик Н.С.*					
Зав. каф.		*Осухівська Г.М.*			*Перелік елементів*		

Додаток В

Лістинг програми

Лістинг В.1 – Код програми одноплатного мікрокомп’ютера Raspberry Pi 4 для реалізації мультимодальної системи біометричної автентифікації

```
import time
import os
import requests
import cv2
import face_recognition
import speech_recognition as sr
import OPi.GPIO as GPIO

# --- ПРИХОВУВАННЯ ПОМИЛОК ALSA ---
from ctypes import *
try:
    ERROR_HANDLER_FUNC = CFUNCTYPE(None, c_char_p, c_int,
c_char_p, c_int, c_char_p)
    def py_error_handler(filename, line, function, err, fmt):
        pass
    c_error_handler = ERROR_HANDLER_FUNC(py_error_handler)
    asound = cdll.LoadLibrary('libasound.so.2')
    asound.snd_lib_error_set_handler(c_error_handler)
except Exception:
    pass
# ------------------------------------------------

# ================ НАЛАШТУВАННЯ ================

TELEGRAM_TOKEN =
"8825276158:AAEHBHcrdYq6VseJ4jDY2xb_ipWuCkEo120"
CHAT_ID = "5689972138"
```

```
BUTTON_PIN = 'PL12'
SERVO_PIN = 'PD7'

KEYWORD = "грім"
OWNER_IMAGE_PATH = "owner_face.jpg"

recognizer = sr.Recognizer()

# ================ ФУНКЦІЇ СИСТЕМИ ================

def send_telegram_msg(message):
    """Відправка звичайного текстового сповіщення у Telegram"""
    url =
f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendMessage"
    payload = {'chat_id': CHAT_ID, 'text': message}
    try:
        requests.post(url, data=payload, timeout=5)
    except Exception:
        pass

def send_telegram_photo(caption, photo_path):
    """Відправка фотографії з підписом у Telegram"""
    url =
f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendPhoto"
    try:
        with open(photo_path, 'rb') as photo:
            payload = {'chat_id': CHAT_ID, 'caption': caption}
            files = {'photo': photo}
            requests.post(url, data=payload, files=files,
timeout=10)
    except Exception as e:
        print(f"[Помилка мережі] Не вдалося відправити фото:
{e}")

def play_success_sound():
```

```
    """1 довгий звук (1 секунда, 1000 Гц) з гучністю 7.5%"""
    print("[Звук] Успіх (1 довгий)")
    os.system("play -v 0.07 -nq -t alsa synth 1 sine 1000
2>/dev/null")

def play_fail_sound():
    """4 коротких звука (по 0.2 секунди) з гучністю 7.5%"""
    print("[Звук] Помилка (4 коротких)")
    for _ in range(4):
        os.system("play -v 0.07 -nq -t alsa synth 0.2 sine 1000
2>/dev/null")
        time.sleep(0.1)

def voice_check():
    """Апаратний запис звуку (обхід блокувань sudo) та
розпізнавання"""
    print("\n--- Етап 1: Голосова перевірка ---")
    print("Скажіть пароль (запис триватиме 4 секунди)...")

    record_cmd = "arecord -D plughw:2,0 -d 4 -c 1 -f S16_LE -r
44100 -q /tmp/voice_record.wav"
    os.system(record_cmd)

    print("Обробка аудіо...")
    try:
        with sr.AudioFile("/tmp/voice_record.wav") as source:
            audio = recognizer.record(source)
            text = recognizer.recognize_google(audio,
language="uk-UA").lower()

            print(f"Система почула: ,,{text},,")
            return text

    except sr.UnknownValueError:
        print("Не вдалося розпізнати слова (можливо тиша).")
```

```
        return None
    except sr.RequestError:
        print("Помилка сервісу розпізнавання (немає
інтернету?).")
        return None
    except Exception as e:
        print(f"Апаратна помилка файлу: {e}")
        return None

def face_check(known_face_encoding):
    """Захоплює кадр, зберігає його та перевіряє обличчя"""
    print("\n--- Етап 2: Запуск камери для сканування обличчя --
-")
    video_capture = cv2.VideoCapture(0)
    time.sleep(1)

    ret, frame = video_capture.read()
    video_capture.release()

    if not ret:
        print("Помилка доступу до камери.")
        return False, None

    photo_path = "/tmp/captured_face.jpg"
    cv2.imwrite(photo_path, frame)

    small_frame = cv2.resize(frame, (0, 0), fx=0.5, fy=0.5)
    rgb_small_frame = cv2.cvtColor(small_frame,
cv2.COLOR_BGR2RGB)

    face_locations =
face_recognition.face_locations(rgb_small_frame)
    face_encodings =
face_recognition.face_encodings(rgb_small_frame, face_locations)
```

```
    if not face_encodings:
        print("Обличчя не знайдено в кадрі.")
        return False, photo_path

    match =
face_recognition.compare_faces([known_face_encoding],
face_encodings[0], tolerance=0.5)
    return match[0], photo_path

def open_lock():
    """Імітація відкриття замка сервоприводом (Програмний ШІМ
1000мкс - 2000мкс)"""
    print("\n--- Етап 3: Відкриття замка (сервопривід) ---")
    try:
        print("Відкриття замка (імпульс 2000 мкс)...")
        # 50 циклів = 1 секунда (50 * 20мс)
        for _ in range(50):
            GPIO.output(SERVO_PIN, GPIO.HIGH)
            time.sleep(0.002)   # HIGH на 2 мілісекунди
            GPIO.output(SERVO_PIN, GPIO.LOW)
            time.sleep(0.018)   # LOW на 18 мілісекунд

        time.sleep(1.5) # Тримаємо двері відкритими 1.5 секунди

        print("Закриття замка (імпульс 1000 мкс)...")
        for _ in range(50):
            GPIO.output(SERVO_PIN, GPIO.HIGH)
            time.sleep(0.001)   # HIGH на 1 мілісекунду
            GPIO.output(SERVO_PIN, GPIO.LOW)
            time.sleep(0.019)   # LOW на 19 мілісекунд

    except Exception as e:
        print(f"Помилка сервоприводу: {e}")

# ================ ГОЛОВНИЙ ЦИКЛ ================
```

```
def main():
    print("Ініціалізація системи...")

    GPIO.setmode(GPIO.SUNXI)
    GPIO.setwarnings(False)

    GPIO.setup(BUTTON_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    GPIO.setup(SERVO_PIN, GPIO.OUT)

    if not os.path.exists(OWNER_IMAGE_PATH):
        print(f"[КРИТИЧНО] Файл {OWNER_IMAGE_PATH} не
знайдено!")
        return

    owner_image =
face_recognition.load_image_file(OWNER_IMAGE_PATH)
    known_face_encoding =
face_recognition.face_encodings(owner_image)[0]

    print("\nСистема готова і переведена у сплячий режим.")
    send_telegram_msg("🟢 Система аутентифікації увімкнена і
готова до роботи.")

    try:
        while True:
            GPIO.wait_for_edge(BUTTON_PIN, GPIO.FALLING)

            time.sleep(0.3)
            if GPIO.input(BUTTON_PIN) == GPIO.HIGH:
                continue

            print("\n" + "="*40)
            print("СИСТЕМУ АКТИВОВАНО! Початок перевірки.")
```

```
            # --- Етап 1: Голос ---
            spoken_word = voice_check()

            if spoken_word is not None and KEYWORD in
spoken_word:
                play_success_sound()
                print("☑ Голос підтверджено.")
            else:
                play_fail_sound()
                if spoken_word:
                    error_msg = f"🔘 Невдала спроба входу:
замість пароля сказано ,,{spoken_word},,"
                else:
                    error_msg = "🔘 Невдала спроба входу: голос
не розпізнано або тиша."

                send_telegram_msg(error_msg)
                print(f"❌ {error_msg}")
                print("❌ Перевірку перервано. Повернення в
сплячий режим.")
                continue

            # --- Етап 2: Обличчя ---
            face_matched, photo_path =
face_check(known_face_encoding)

            if face_matched:
                play_success_sound()
                print("☑ Обличчя підтверджено.")
                if photo_path:
                    send_telegram_photo("☑ Обличчя успішно
розпізнано.", photo_path)
            else:
                play_fail_sound()
```

```
                error_msg = "🔘 Невдала спроба входу: обличчя не
розпізнано."
                if photo_path:
                    send_telegram_photo(error_msg, photo_path)
                else:
                    send_telegram_msg(error_msg)

                print(f"❌ {error_msg}")
                print("❌ Перевірку перервано. Повернення в
сплячий режим.")
                continue

            # --- Етап 3: Відкриття (успіх) ---
            send_telegram_msg("🔓 Успішний вхід. Замок
відкрито.")
            open_lock()

            print("Процедура завершена. Перехід у сплячий
режим.")
            time.sleep(2)

    except KeyboardInterrupt:
        print("\nВихід з програми...")
    finally:
        GPIO.cleanup()
        if os.path.exists("/tmp/voice_record.wav"):
            os.remove("/tmp/voice_record.wav")
        if os.path.exists("/tmp/captured_face.jpg"):
            os.remove("/tmp/captured_face.jpg")

if __name__ == "__main__":
    main()
```