МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ

Навчально-науковий інститут кібернетики,

інформаційних технологій та інженерії

"До захисту допущена"
Зав. кафедри комп'ютерних наук та
прикладної математики
д.т.н., проф. Турбал Ю.В.
«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

Розробка програмно-апаратного забезпечення системи безпеки

Виконала: **Головчук Юлія Дмитрівна** _____
(прізвище, ім'я, по батькові) (підпис)

студентка групи ІПЗ-41 інт.

Керівник: к.т.н., доцент, доцент **Климюк Ю. Є.** _____
(науковий ступінь, вчене звання, посада, прізвище, ініціали) (підпис)

Рівне – 2025

ЗМІСТ

РЕФЕРАТ

Кваліфікаційна робота: 47 сторінок, 19 рисунків, 2 таблиці, 15 джерел.

Актуальність теми: Забезпечення безпеки житлових та комерційних об’єктів є одним із пріоритетних завдань у сучасному технологічному суспільстві. Водночас висока вартість, складність конфігурації та залежність від платних сервісів існуючих професійних систем суттєво обмежують їх доступність, що зумовлює актуальність розробки бюджетних, автономних та простих в інтеграції рішень на основі мікроконтролерів та концепції Інтернету речей.

Метою роботи є проектування та реалізація програмно-апаратного комплексу системи безпеки на основі мікроконтролера ESP32-CAM для комплексного моніторингу визначеної території, що здатна виявляти комплекс таких загроз, як несанкціоноване відчинення дверей, рух в приміщенні та витік горючих газів з подальшою візуальною верифікацією та миттєвим сповіщенням.

Об’єкт дослідження – процеси функціонування та взаємодії компонентів програмно-апаратного забезпечення системи безпеки на базі мікроконтролерних платформ.

Предмет дослідження – методи побудови архітектури, алгоритми обробки даних із сенсорів, протоколи передачі інформації та інструментальні засоби для створення енергоефективних, економічно доступних та простих у конфігурації програмно-апаратних рішень для систем безпеки.

Методи дослідження – теоретичне узагальнення та системний аналіз, порівняльний аналіз, моделювання архітектури та алгоритмів, програмна реалізація, експериментальне тестування.

Ключові слова: ESP32-CAM, система безпеки, сенсори, Інтернет речей, мікроконтролер, кібербезпека, енергоефективність.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ESP32-CAM — мікроконтролер з інтегрованою камерою, модулями Wi-Fi та Bluetooth, що використовується для реалізації логіки системи.

PIR — пасивний інфрачервоний датчик, призначений для детекції руху об’єктів, що випромінюють тепло.

MQ-6 — напівпровідниковий газовий датчик для виявлення в повітрі зрідженого нафтового газу (LPG), метану, бутану та пропану.

MC-38 — магнітоконтактний датчик, що використовується для контролю стану відкриття/закриття дверей або вікон.

UART — універсальний асинхронний приймач-передавач, стандарт послідовної комунікації між електронними пристроями.

GPIO — універсальний порт вводу/виводу, що використовується для взаємодії мікроконтролера із зовнішніми пристроями.

OTA — технологія віддаленого оновлення програмного забезпечення пристрою без фізичного підключення.

PSRAM — псевдостатична оперативна пам’ять, тип динамічної пам’яті з інтегрованим контролером оновлення

ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується стрімкою цифровою трансформацією суспільства, в основі якої лежить концепція Інтернету речей (IoT). Підхід, що передбачає інтеграцію фізичних пристроїв, оснащених сенсорами, виконавчими механізмами та засобами комунікації, у глобальну комп'ютерну мережу, докорінно змінює ключові аспекти повсякденного життя, промисловості та бізнесу. Особливо відчутні зміни відбуваються у сфері безпеки, де традиційні механічні та локальні електронні пристрої поступаються місцем складним, інтелектуальним кіберфізичним комплексам, здатним до автономного моніторингу, інтелектуального аналізу ситуації та мережевої взаємодії в режимі реального часу [1].

Незважаючи на технологічний прогрес, професійні системи безпеки залишаються фінансово та організаційно недоступними для значної частини власників житлової та комерційної нерухомості. У зв'язку з цим виникає об'єктивна потреба в розробці бюджетних, енергоефективних, портативних та інтуїтивно зрозумілих систем моніторингу, що не вимагають від користувача спеціалізованих навичок для монтажу й подальшої експлуатації.

Актуальність теми дослідження зумовлена потребою в доступних та ефективних системах моніторингу безпеки. Поява нового покоління потужних та недорогих мікроконтролерів, зокрема ESP32-CAM, що поєднують в одному модулі достатню обчислювальну потужність, бездротову передачу даних та обробку зображень, створює нові технологічні передумови для вирішення цієї проблеми. Запропонований у роботі підхід спрямований на практичну реалізацію програмно-апаратного комплексу, що використовує потенціал саме таких платформ.

Мета роботи полягає у розробці програмно-апаратного комплексу для автономної системи безпеки на платформі ESP32-CAM, що забезпечить комплексний моніторинг території шляхом виявлення таких загроз, як

несанкціонований рух в приміщенні, відчинення дверей та підвищена концентрація легкозаймистих газів. У разі фіксації інцидентів система надсилає миттєві сповіщення через месенджер Telegram, при цьому тривога, спричинена рухом, підкріплюється фотознімком з місця події, що сприяє підвищенню рівня безпеки об’єктів при мінімальних фінансових і часових витратах.

Для досягнення поставленої мети в роботі використовувалися такі методи дослідження:

1. Метод теоретичного узагальнення та системного аналізу для вивчення еволюції систем безпеки, сучасних підходів до їх побудови та визначення основних вимог до пристрою.
2. Порівняльний аналіз для обґрунтування вибору апаратної платформи та компонентної бази.
3. Моделювання архітектури, структурної схеми та алгоритмів функціонування програмно-апаратного комплексу.
4. Програмна реалізація в середовищі Arduino IDE.
5. Експериментальне тестування прототипу для перевірки його працездатності та відповідності поставленим вимогам.

Наукова новизна отриманих результатів полягає в обґрунтуванні архітектурного підходу до створення компактних автономних систем безпеки, який, на відміну від існуючих, дозволить забезпечити комплексну детекцію загроз з візуальною верифікацією та миттєвим сповіщенням без розгортання складної серверної інфраструктури при мінімальному енергоспоживанні.

Практична цінність роботи визначається тим, що результати дослідження можуть бути використані для створення готового до впровадження, економічно вигідного пристрою для моніторингу та своєчасного реагування на потенційні загрози в житлових приміщеннях, невеликих офісах, гаражах та інших об'єктах, надаючи користувачам повний контроль над власними даними.

РОЗДІЛ І
ТЕОРЕТИЧНА ЧАСТИНА

1.1. Історія та еволюція систем безпеки

Для розуміння значущості та інноваційного потенціалу сучасних рішень у сфері захисту, зокрема розробленої системи, важливо проаналізувати їхній історичний розвиток. Потреба у захисті майна та життя є однією з фундаментальних, що зумовлює постійне вдосконалення методів забезпечення безпеки від примітивних механізмів до складних кіберфізичних комплексів.

На ранніх етапах історії, що охоплюють доіндустріальну епоху, захист об'єктів забезпечувався переважно фізичними бар'єрами, такими як рови, стіни, масивні двері, фортифікаційні споруди, та вимагав постійного нагляду з боку вартових. Хоча існували прості механічні сигналізатори, що сповіщали про відчинення дверей чи воріт за допомогою дзвонів, їхня ефективність була вкрай обмеженою.

Кардинальні зміни у сфері безпеки стали можливими лише з опануванням електрики, що дозволило автоматизувати процес виявлення загроз. Першим практичним втіленням даної ідеї стала електромагнітна система сигналізації, запатентована винахідником Августусом Расселом Поупом у 1853 році [2]. Принцип дії системи полягав у використанні електричного кола, що замикалося при відчиненні дверей або вікон, активуючи звуковий сигналізатор у вигляді дзвону чи сирени [3]. Подібні рішення вперше забезпечили автоматичну детекцію небажаних подій, проте виключно локальний характер сповіщення залишається головним обмеженням.

Цю проблему вирішили в середині XX століття з появою перших централізованих охоронних систем. Сигнали з датчиків почали передаватися дротовими лініями зв'язку на центральну станцію моніторингу, оператори якої викликали відповідні служби реагування. Такий підхід значно підвищив ефективність захисту, проте його впровадження було дороговартісним та

технічно складним через необхідність прокладання розгалужених кабельних мереж.

Подолати ці бар’єри вдалося наприкінці XX століття завдяки стрімкому розвитку бездротових технологій та мікропроцесорів. Це дозволило створювати компактні пристрої зі значно складнішою логікою роботи та усунуло потребу в дорогому монтажі, зробивши системи гнучкими й доступними для масового поширення.

Подальша еволюція тісно пов'язана з концепцією Інтернету речей, що перетворила окремі пристрої на інтегровані інтелектуальні екосистеми. Сучасні охоронні комплекси використовують гібридну модель обчислень, тобто обробка даних відбувається локально на пристрої, а хмарні платформи використовуються для довгострокового зберігання даних, розширеної аналітики та надання віддаленого доступу. Основною зміною стало активне впровадження штучного інтелекту та машинного навчання. Ці технології дозволили перейти від простої реакції на подію до інтелектуального аналізу: розпізнавання об’єктів для зменшення хибних спрацювань, аналізу аномальної поведінки та прогнозування потенційних загроз. Системи здатні інтегруватися з іншими елементами "розумного дому", дозволяючи створювати складні автоматизовані сценарії, що значно підвищує функціональність. Наприклад, можливе автоматичне ввімкнення світла при виявленні руху, блокування дверей та вікон при спрацюванні сигналізації, або перекриття газопостачання при фіксації витоку газу, що забезпечує комплексний захист.

Водночас глобальна цифрова трансформація та стрімкий розвиток інноваційних технологій створюють суттєві виклики, насамперед у сфері кібербезпеки [4]. Сучасні системи безпеки є потенційною мішенню для кібератак, здатних порушити конфіденційність, цілісність або доступність даних. Найбільш поширеними векторами атак є атаки на відмову в обслуговуванні (DoS), перехоплення даних (Man-in-the-middle) та отримання несанкціонованого доступу до пристрою через слабкі паролі або програмні

вразливості. Це вимагає від розробників впровадження комплексних механізмів захисту, включно із шифруванням даних, багатофакторною аутентифікацією та постійним моніторингом вразливостей відповідно до стандартів.

1.2. Архітектура та складові компоненти системи

Проєктована система безпеки є автономним програмно-апаратним комплексом, архітектура якого побудована за модульним принципом. Подібний підхід дозволяє інтеграцію додаткових функціональних блоків у майбутньому. Центральним обчислювальним ядром виступає мікроконтролерний модуль ESP32-CAM (рис.1.1). Вибір модуля обґрунтований його унікальними характеристиками: потужний двоядерний процесор Xtensa LX6 забезпечує достатню продуктивність не лише для виконання основних логічних операцій, а й для ефективної обробки зображень; вбудовані модулі Wi-Fi та Bluetooth слугують надійною основою для бездротової комунікації; мініатюрна інтегрована камера OV2640 дозволяє здійснювати візуальну верифікацію подій, а підтримка карт пам’яті формату microSD розширює можливості локального зберігання даних.

Рис. 1.1. ESP32-CAM модуль з материнською платою

Для реалізації комплексного моніторингу та виявлення загроз до мікроконтролера підключається набір спеціалізованих сенсорів:

1) Пасивний інфрачервоний сенсор руху (PIR) HC-SR505 (рис.1.2)

Призначений для виявлення змін рівня інфрачервоного випромінювання, що дозволяє фіксувати присутність людини чи тварини в полі зору. Сенсор характеризується низьким енергоспоживанням, компактними розмірами та високою чутливістю.

Рис. 1.2. Інфрачервоний датчик руху HC-SR505

2) Датчик газу MQ-6 (рис.1.3)

Напівпровідниковий сенсор резистивного типу, електричний опір якого змінюється залежно від концентрації в повітрі зрідженого нафтового газу (LPG), метану, бутану та пропану. Його аналоговий вихід дозволяє оцінювати рівень загазованості повітря.

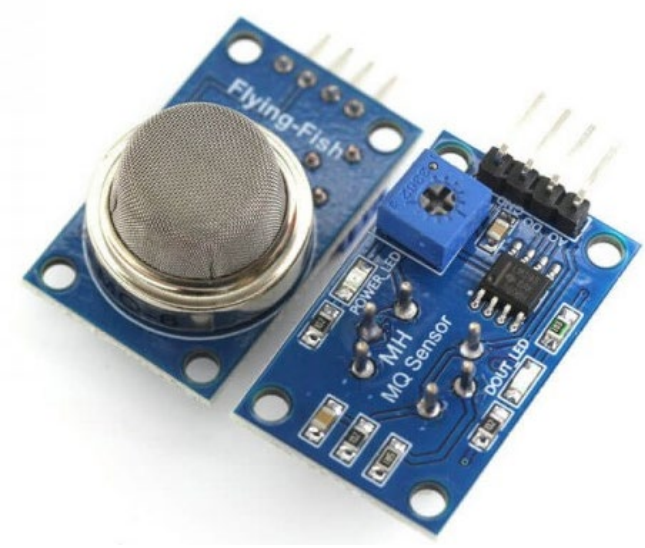


Рис. 1.3. Модуль датчика газу MQ-6

3) Магнітний контактний датчик МС-38 (рис.1.4)

Складається з геркона та магніту. Призначений для контролю стану дверей або вікон (відкрито/закрито) шляхом фіксації їх взаємного положення.

Рис. 1.4. Магнітний датчик відкриття дверей вікна геркон МС-38

Оскільки мікроконтролер ESP32-CAM працює з логічними рівнями 3.3 В, а деякі сенсори можуть вимагати живлення 5 В, для їх коректного узгодження та захисту компонентів у систему інтегровано модуль перетворювача логічних рівнів (Logic Level Shifter), зображений на рисунку 1.5.

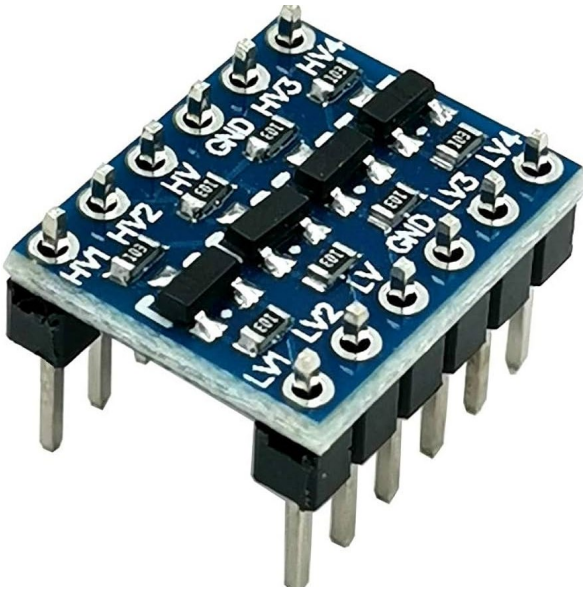


Рис. 1.5. Перетворювач логічних рівнів

Весь комплекс функціонує як єдина система: сенсори фіксують тривожні події, ESP32-CAM обробляє ці дані, робить знімок, зберігає його на microSD-

карту та через Wi-Fi надсилає сповіщення користувачеві в Telegram. Такий підхід забезпечує віддалений моніторинг та швидке реагування на загрози. Гнучкість живлення, що дозволяє працювати як від стаціонарної мережі, так і від портативних джерел (наприклад, Power Bank), гарантує високий рівень автономності системи.

1.3. Вибір технологій та програмного забезпечення

Ключовим етапом для успішної реалізації проєкту є вибір апаратної платформи та програмних інструментів, що відповідають критеріям функціональності, енергоефективності, вартості та потенціалу для масштабування.

Центральним елементом системи, що відповідає за обробку даних, керування периферією та комунікацію, було обрано мікроконтролерний модуль ESP32-CAM. Порівняно з альтернативними платформами, як Arduino Uno чи NodeMCU (на базі ESP8266), модуль ESP32-CAM має ряд переваг:

1) продуктивність: двоядерний процесор Xtensa LX6 та значно більший обсяг вбудованої оперативної пам’яті є критично важливими для захоплення та обробки зображень з камери без суттєвих затримок;
2) розширена пам'ять: наявність на платі додаткового модуля псевдостатичної оперативної пам'яті (PSRAM) об'ємом 4 МБ дозволяє буферизувати зображення високої роздільної здатності перед їх надсиланням, що стабілізує роботу системи;
3) інтегровані компоненти: вбудована камера OV2640 та слот для microSD-карти спрощують апаратну архітектуру та зменшують габарити кінцевого пристрою;
4) зручність обслуговування: підтримка технології оновлення програмного забезпечення "по повітрю" (Over-the-Air, OTA) дозволяє оновлювати прошивку пристрою віддалено, без необхідності фізичного підключення до комп'ютера, що суттєво спрощує його подальше обслуговування.

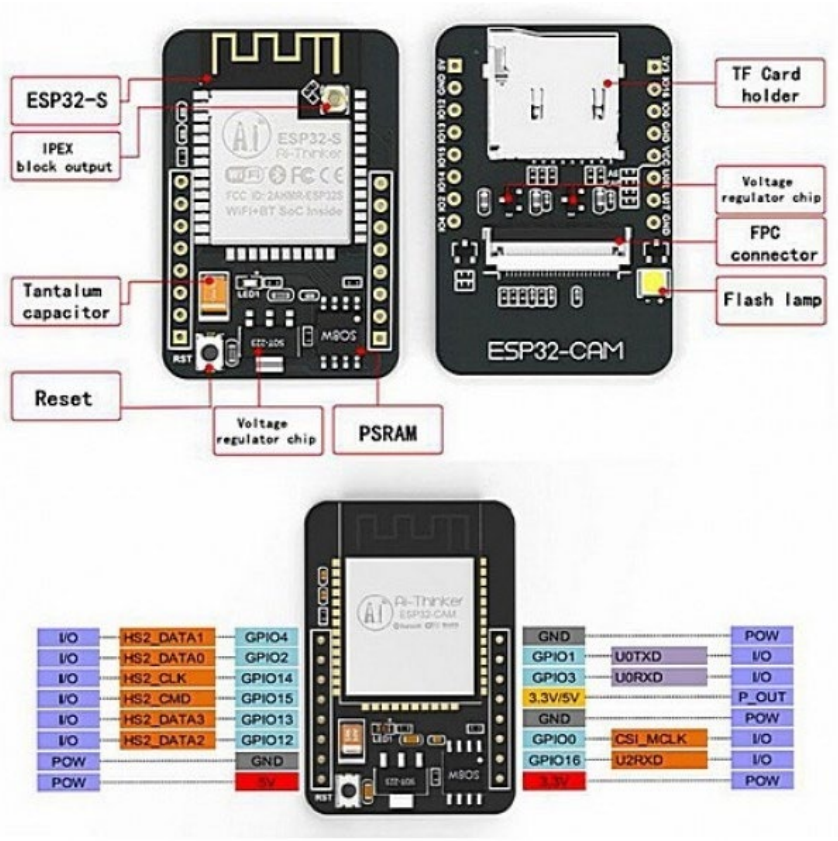


Рис. 1.6. Структура та розпіновка модуля ESP32-CAM

На рисунку 1.6 представлено основні елементи модуля ESP32-CAM, включаючи мікросхему ESP32-S, слот для microSD-карти, FPC-конектор для камери, а також розташування виводів GPIO, що використовуються для підключення сенсорів та інших зовнішніх пристроїв [5].

Для реалізації комплексного моніторингу було обрано комбінацію з трьох датчиків, що реагують на найпоширеніші побутові загрози: несанкціоноване проникнення (рух та відчинення дверей) і витік газу. Такий набір є оптимальним для базового захисту приміщення, оскільки забезпечує баланс між функціональністю, вартістю та простотою інтеграції. Розглянемо принцип їхньої роботи детальніше.

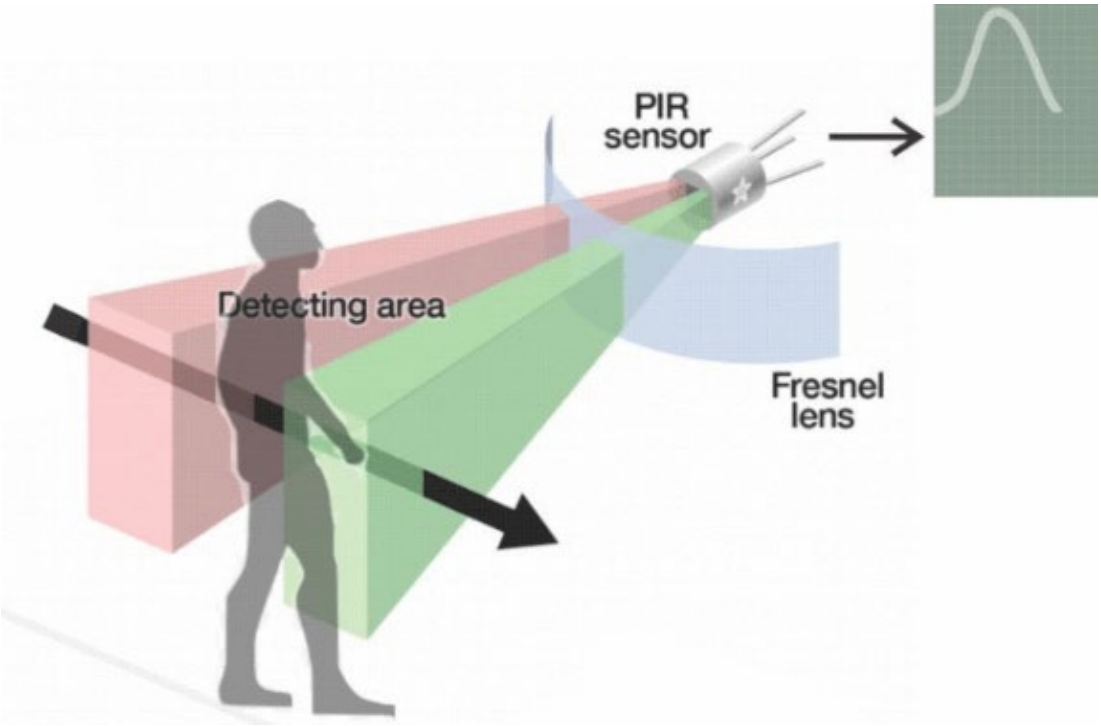


Рис. 1.7. Принцип дії пасивного інфрачервоного сенсора (PIR) з лінзою Френеля

На рисунку 1.7. схематично зображено принцип роботи PIR-сенсора, що широко застосовується в системах автоматизації освітлення, охоронної сигналізації та інтелектуального моніторингу [6]. Назва "пасивний" підкреслює, що сенсор не випромінює енергію, а лише сприймає теплове випромінювання від об’єктів у його полі зору.

Основним елементом конструкції є чутливий піроелектричний елемент, що генерує невелику електричну напругу при зміні температури його поверхні.

Для ефективної детекції руху використовується спеціальна напівсферична лінза Френеля. Вона візуально поділяє зону огляну на активні та пасивні сектори, що дозволяє фокусувати інфрачервоне випромінювання від рухомих об'єктів на піроелектричний елемент. Коли людина перетинає ці сектори, тепловий потік динамічно змінюється. Це дозволяє сенсору ігнорувати статичні джерела тепла (наприклад, радіатори опалення) та ефективно виявляти рух живих істот.

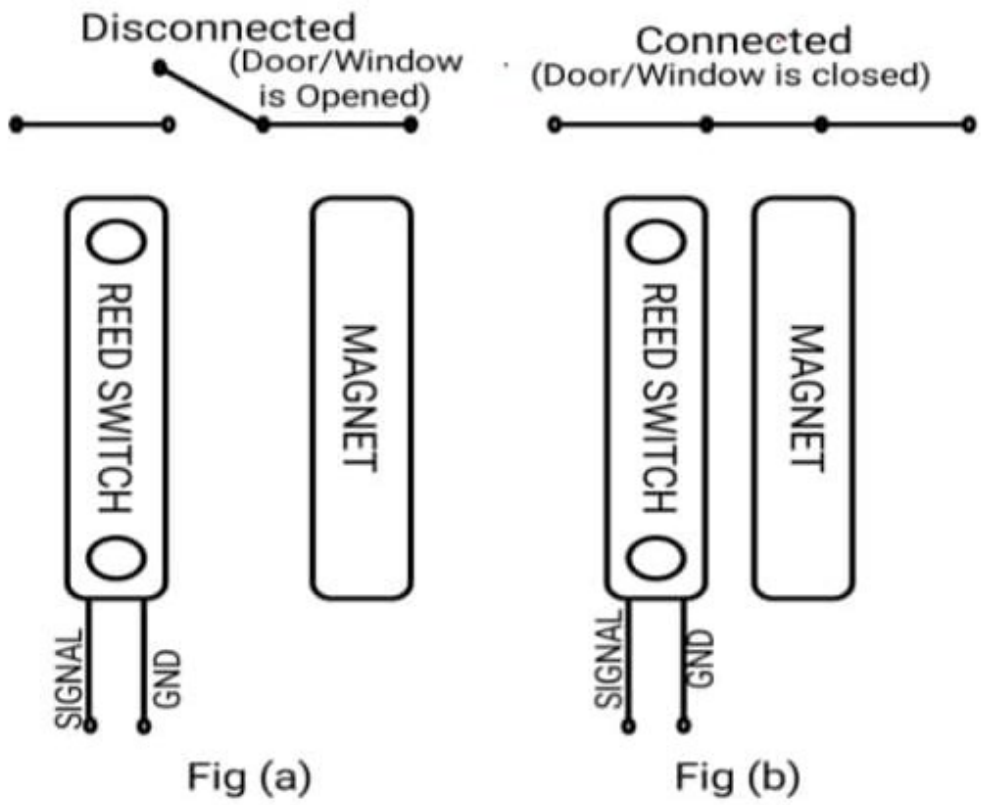


Рис. 1.8. Принцип дії магнітоконтактного датчика МС-38: а) стан "відкрито"; б) "закрито"

На рисунку 1.8. продемонстровано принцип роботи магнітоконтактного датчика МС-38, що є одним із найпростіших пристроїв для контролю стану дверей, вікон, люків та інших рухомих конструкцій [7].

Датчик складається з двох основних частин: основного блоку з герметизованим контактом (Reed Switch) та допоміжного блоку з магнітом (Magnet). Геркон являє собою два феромагнітні контакти в корпусі для захисту від корозії й пилу.

При відчиненні дверей магніт віддаляється від сенсорної частини. Сила магнітного поля слабшає і феромагнітні контакти всередині геркона роз’єднуються. Схематично це зображено як розімкнений вимикач (рис.1.8, а).

Коли двері зачинені, магнітний блок знаходиться поруч із герконом. Сильне магнітне поле діє на феромагнітні контакти, змушуючи їх притягнутися один до одного та замкнути електричне коло. Схематично це представлено як замкнений вимикач (рис.1.8, б).

Ця зміна стану електричного кола (розімкнено або замкнено) реєструється системою сигналізації, яка інтерпретує розмикання контактів (при віддаленні магніту) як подію відкриття, що може сигналізувати про тривогу. Простота конструкції, відсутність власного енергоспоживання в статичному стані та висока надійність роблять такі датчики популярним рішенням для охоронних систем з автономним живленням.

На рисунку 1.9 [8] зображена схема модуля газового сенсора MQ-6, що належить до класу хеморезистивних напівпровідникових датчиків і призначений для виявлення горючих газів, таких як пропан, бутан, метан та зріджений нафтовий газ (LPG). Його робота ґрунтується на зміні електричного опору чутливого матеріалу при взаємодії з молекулами газу.

Чутливим елементом сенсора є шар із напівпровідникового матеріалу — діоксиду олова, нанесений на керамічну підкладку. Всередині підкладки розташований нагрівальний елемент, що підтримує робочу температуру чутливого шару в діапазоні 200–400 °C. Це необхідно для активації хімічних реакцій.

У чистому повітрі молекули кисню з атмосфери адсорбуються на поверхні нагрітих кристалів діоксиду олова, захоплюючи вільні електрони з напівпровідника. Це створює високий потенційний бар'єр, що призводить до дуже високого електричного опору чутливого елемента.

У присутності горючого газу молекули горючого газу вступають у реакцію з адсорбованими іонами кисню, вивільняючи раніше захоплені електрони. Це різко знижує загальний електричний опір сенсора. Чим вища концентрація газу, тим нижчим стає опір.

Модуль MQ-6 обробляє цю зміну опору і надає два типи вихідних сигналів [9]:

1) Аналоговий вихід (AO): зміна опору сенсора перетворюється на зміну вихідної напруги. Цей сигнал дозволяє мікроконтролеру за допомогою

аналого-цифрового перетворювача вимірювати відносну концентрацію газу для моніторингу динаміки.

2) Цифровий вихід (DO): на платі встановлено компаратор (наприклад, LM393), що порівнює напругу з аналогового виходу із пороговим значенням, встановленим потенціометром. Якщо концентрація газу перевищує поріг, напруга на виході сенсора падає нижче порогової, і компаратор видає на вихід DO логічний сигнал тривоги.

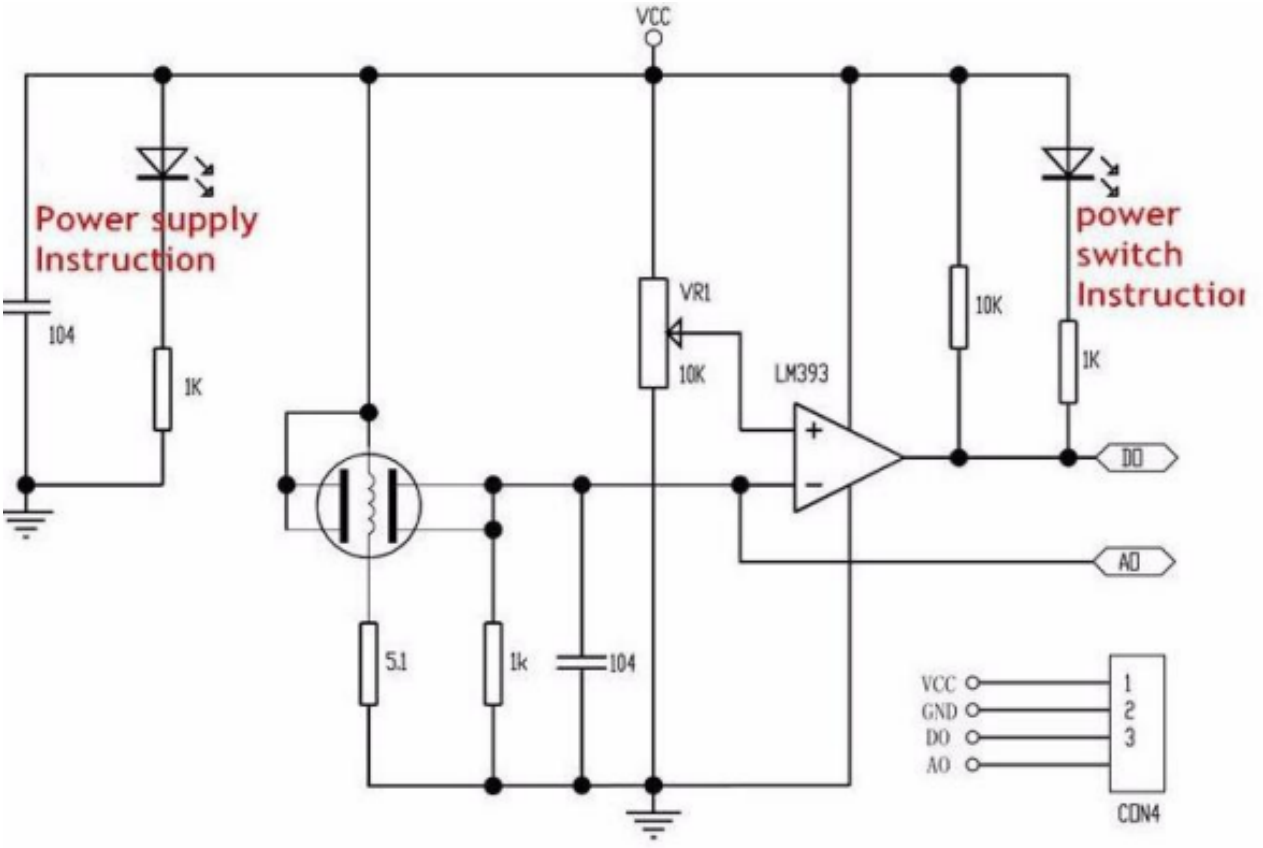


Рис. 1.9. Схема модуля газового сенсора MQ-6

Середовищем розробки програмного забезпечення обрано Arduino IDE завдяки його інтуїтивно зрозумілому інтерфейсу, широкій підтримці апаратних платформ, включно з мікроконтролерами ESP32 та наявності великої спільноти розробників. Це забезпечує доступ до значної кількості готових бібліотек, що суттєво прискорює процес розробки та дозволяє ефективно інтегрувати різні апаратні компоненти.

1.4. Архітектурні моделі Інтернету речей

Для глибшого розуміння принципів побудови IoT-систем, до яких належить і розроблений прототип, важливо розглянути їхні узагальнені архітектурні моделі. Згідно з дослідженням Рахамана М. М. [10], не існує єдиного стандартизованого підходу, проте найчастіше виділяють три- та п'ятирівневі архітектури.

Трирівнева архітектура є базовою моделлю і включає такі рівні:

1) рівень сприйняття (Perception Layer): це фізичний рівень, відповідальний за збір даних із навколишнього середовища. Він складається з сенсорів (для вимірювання температури, руху, вологості тощо), актуаторів (для виконання дій, як увімкнення світла чи закриття замка) та інших кінцевих пристроїв. Основне завдання — перетворення фізичних параметрів на цифрові дані. У розробленій системі цей рівень представлений сенсорами PIR HC-SR505, MQ-6 та MC-38, а також камерою OV2640;
2) мережевий рівень (Network Layer): відповідає за передачу даних, зібраних на рівні сприйняття, до обчислювальних систем для подальшої обробки. Цей рівень охоплює різноманітні комунікаційні технології (Wi-Fi, Bluetooth, Zigbee, 4G/5G, LoRaWAN) та мережеве обладнання (маршрутизатори, комутатори). У роботі реалізовано за допомогою вбудованого в ESP32-CAM модуля Wi-Fi, що підключається до локальної мережі для доступу в Інтернет;
3) рівень застосунків (Application Layer): верхній рівень архітектури, що надає сервіси кінцевому користувачеві. Він інтерпретує отримані дані та реалізує логіку відповідно до призначення системи (моніторинг, керування, аналітика). У роботі цей рівень представлений логікою, реалізованою на мікроконтролері, та інтеграцією з Telegram Bot API, що дозволяє надсилати користувачеві сповіщення та фотографії.

П'ятирівнева архітектура є розширеною версією трирівневої, забезпечуючи більш детальну декомпозицію системи, що особливо важливо для складних комерційних проєктів [10]. Вона включає:

1) рівень об'єктів (Perception/Objects Layer): аналогічний рівню сприйняття у трирівневій моделі;
2) рівень об'єктної абстракції (Object Abstraction Layer): забезпечує попередню обробку, фільтрацію та передачу даних від фізичних пристроїв через захищені канали;
3) рівень управління сервісами (Service Management Layer): відповідає за аналіз даних, прийняття рішень, управління потоками даних та забезпечення безпеки. На цьому рівні відбувається динамічне сполучення сервісів з тими компонентами, які їх потребують, забезпечуючи при цьому гнучкість та сумісність системи;
4) рівень застосунків (Application Layer): реалізує логіку конкретних застосунків та надає користувацькі інтерфейси (наприклад, системи моніторингу розумного будинку, управління логістикою, промислова автоматизація);
5) бізнес-рівень (Business Layer): найвищий рівень, що керує всією IoT-системою, включаючи застосунки та сервіси. На цьому рівні аналізуються дані для прийняття стратегічних рішень, розробляються бізнес-моделі, визначаються стандарти, що сприяють досягненню комерційних цілей.

Розроблена в межах даної роботи система безпеки, хоч і є компактним прототипом, чітко відповідає базовій трирівневій архітектурній моделі, що підтверджує обґрунтованість застосованих підходів при проєктуванні та реалізації.

РОЗДІЛ ІІ.

АНАЛІТИЧНА ЧАСТИНА

2.1. Вимоги до прототипу системи безпеки

Розроблений прототип системи безпеки має відповідати таким основним вимогам:

1) Комплексна детекція загроз: система повинна фіксувати рух в контрольованій зоні, виявляти підвищену концентрацію горючих газів, а також забезпечувати моніторинг несанкціонованого відкриття дверей.
2) Візуальна верифікація: у момент спрацювання датчика руху здійснюється автоматична фотофіксація події за допомогою інтегрованої камери.
3) Оперативне сповіщення: миттєве надсилання повідомлень в Telegram про тип зафіксованої події.
4) Автономність функціонування: прототип має стабільно працювати від портативних джерел живлення. Це забезпечить гнучкість розміщення та безперебійну роботу навіть у разі відключення основного електропостачання.
5) Інтуїтивність налаштування: процес конфігурації системи має бути простим і не вимагати від користувача спеціалізованих технічних знань.
6) Масштабованість: архітектура системи повинна передбачати можливість розширення функціональності шляхом додавання нових сенсорів (наприклад, диму, затоплення) або виконавчих пристроїв.

2.2. Порівняльний аналіз аналогічних рішень

Для визначення конкурентних переваг розробленої системи безпеки було проведено аналіз існуючих рішень, які умовно можна поділити на дві основні категорії: професійні системи та DIY-проєкти ("зроби сам"). Порівняння здійснювалося за такими ключовими критеріями, як функціональні можливості, вартість придбання та експлуатації, доступність для кінцевого користувача, гнучкість у налаштуванні та розширенні, складність інсталяції та конфігурації, а також можливість автономного функціонування.

2.2.1. Професійні системи безпеки

Серед найпоширеніших комерційних рішень у сфері безпеки, представлених на українському та міжнародному ринках, варто відзначити Ajax Systems, Ring Alarm (Amazon), Arlo Security System та Xiaomi Smart Home. Ці платформи поєднують високий рівень якості апаратного виконання, розвинену екосистему інтегрованих пристроїв та інтуїтивно зрозумілі мобільні застосунки для керування. Архітектурно побудовані за моделлю "зірка", де центральний хаб збирає дані з бездротових датчиків і забезпечує зв'язок із хмарною інфраструктурою виробника. Зазвичай вони підтримують такі основні функції, як відеоспостереження в режимі реального часу з можливістю віддаленого моніторингу, різноманітні технології виявлення руху (PIR-сенсори, мікрохвильові радари, відеоаналітика) та інтеграцію з хмарними сервісами для зберігання архівів, даних датчиків і розширеної аналітики. Сучасні системи також активно використовують можливості штучного інтелекту для розпізнавання облич, об’єктів, аналізу поведінки та фільтрації помилкових спрацювань.

Ajax Systems позиціонується як інтегрований професійний комплекс безпеки з акцентом на надійність та простоту експлуатації. Комунікація між компонентами відбувається за допомогою захищеного радіопротоколу Jeweller, що забезпечує дальність зв’язку до 2000 метрів та стійкість до перешкод. Протокол Jeweller використовує блочне шифрування та механізми моніторингу радіоефіру, що дозволяють виявляти спроби глушіння сигналу, забезпечуючи надійний захист каналу зв’язку [11]. Система підтримує широкий спектр датчиків: руху, відкриття, диму, затоплення та розбиття скла. Також інтегровані виконавчі пристрої для керування автоматикою, наприклад "розумні" розетки та реле. Для забезпечення максимальної надійності та оперативної передачі сповіщень застосовується принцип багатоканального зв'язку, а термін автономної роботи датчиків від комплектних батарей сягає семи років. Водночас

до обмежень системи відносять високу початкову вартість та закритість екосистеми, що ускладнює інтеграцію зі сторонніми пристроями.

Ring Alarm від Amazon орієнтована переважно на споживачів у США та ЄС і відрізняється простотою встановлення. Система інтегрована з голосовим помічником Alexa. Однак повноцінне використання її можливостей, включно зі зберіганням відеоархівів та підключенням до професійних моніторингових послуг, вимагає платної підписки. До недоліків також належать менша гнучкість у налаштуванні складних сценаріїв та географічні обмеження доступності деяких сервісів.

Xiaomi Smart Home є бюджетним варіантом для побудови системи розумного дому. Популярність рішення зумовлена великим вибором доступних пристроїв: від базових датчиків до камер, освітлювальних приладів та побутової техніки. Керування здійснюється через універсальний застосунок Mi Home, що підтримує протоколи Zigbee та Wi-Fi. З іншого боку, система значною мірою залежить від доступності зовнішніх європейських чи китайських серверів, що може викликати затримки у сповіщеннях та порушувати питання щодо приватності даних. Офіційна інтеграція з платформами Apple HomeKit чи Google Home часто обмежена і потребує додаткових налаштувань.

Незважаючи на зручність та розширену функціональність, професійні системи мають низку системних недоліків:

1) висока початкова вартість та потенційні регулярні витрати на хмарні сервіси;
2) жорстка прив’язка до пропрієтарної інфраструктури (наприклад, Ajax Cloud, Amazon Alexa), що обмежує гнучкість використання;
3) відсутність можливості гнучкої кастомізації під специфічні потреби користувачів.

2.2.2. DIY-системи на базі мікроконтролерів

Альтернативою комерційним продуктам є самостійне створення систем безпеки на базі одноплатних комп'ютерів, таких як Raspberry Pi, або мікроконтролерів, зокрема Arduino та ESP32. Такі проєкти зазвичай мають відкритий вихідний код, підтримують широкий спектр периферійних пристроїв, а також можуть взаємодіяти з популярними онлайн-сервісами, наприклад Telegram чи Google Drive.

Ключовими перевагами DIY-підходу є економічність, зумовлена значно нижчою вартістю компонентів та високий рівень адаптивності. Користувач отримує повний контроль над програмним кодом, апаратними схемами та даними, що дозволяє інтегрувати нестандартні датчики й реалізовувати унікальні сценарії автоматизації. Водночас цей підхід має суттєві недоліки, висуває високі вимоги до технічних знань, оскільки необхідні навички в галузі програмування та електроніки. Процес монтажу, налаштування та подальшого обслуговування є значно складнішим і тривалішим.

2.3. Обґрунтування переваг розробленого прототипу

Розроблений прототип системи безпеки (рис.2.1) має на меті поєднати переваги обох підходів, мінімізуючи їхні недоліки. Він позиціонується як оптимальний компроміс між вартістю, функціональністю та складністю експлуатації.

Ключові характеристики та переваги розробленого рішення:

1) низька собівартість: використання доступного мікроконтролера ESP32-CAM та поширених сенсорів робить систему економічно вигідною;
2) відносна простота налаштування: застосування середовища розробки Arduino IDE спрощує процес програмування та конфігурації;
3) висока гнучкість та масштабованість: модульна архітектура дозволяє легко інтегрувати додаткові сенсори та реалізовувати нові алгоритми роботи системи, адаптуючи її до змінних потреб;

4) автономне функціонування: можливість живлення від портативних джерел забезпечує мобільність та безперебійну роботу;
5) пряма інтеграція з Telegram: забезпечує оперативне сповіщення без залежності від сторонніх платних сервісів, підвищуючи рівень приватності та незалежності системи;
6) локальне збереження даних: підтримка карт пам'яті microSD для фотофіксації подій забезпечує резервне копіювання та доступ до даних навіть без інтернет-з'єднання.

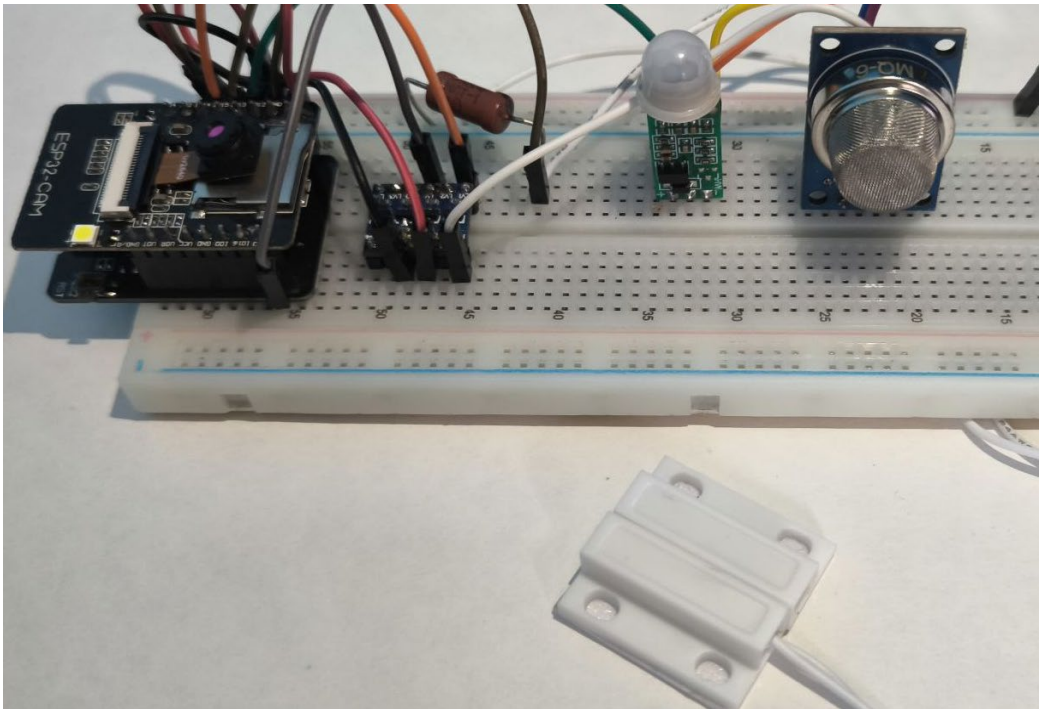


Рис. 2.1. Вигляд прототипу системи безпеки

Порівняльний аналіз (табл.2.1.) демонструє, що розроблена система на базі ESP32-CAM успадковує основні переваги DIY-підходу, такі як низька вартість, висока гнучкість та повний контроль над даними, забезпечуючи при цьому підвищену енергоефективність та є значно простішою в налаштуванні. Особливістю розробленої системи є орієнтація на користувача, який шукає автономну альтернативу, що мінімізує початкові витрати та забезпечує високий рівень приватності з можливістю локального збереження даних.

Табл. 2.1. Порівняльний аналіз систем безпеки

Критерій	Професійні системи (Ajax, Ring, Arlo)	Універсальні DIY на Raspberry Pi	Розроблена система (ESP32-CAM)
Вартість	Висока початкова вартість обладнання. Часто є платна підписка.	Середня вартість: плата, корпус та додаткова периферія.	Мінімальна вартість мікроконтролера та базових датчиків.
Вимоги до налаштування	Просте налаштування в мобільному додатку.	Необхідні навички програмування та налаштування мережевих сервісів.	Потребує прошивки через Arduino IDE.
Енергоефективність	Висока для бездротових датчиків.	Низька. Споживає значну кількість енергії.	Висока. Мікроконтролер підтримує режим глибокого сну.
Локальне збереження даних	Зазвичай дані зберігаються у платному хмарному сховищі.	Дані можна зберігати на SD-карті, зовнішньому диску, або локальному сервері.	Зображення зберігаються безпосередньо на microSD-карту.

РОЗДІЛ ІІІ.

ПРИКЛАДНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1. Проєктування системи

3.1.1. Алгоритм функціонування

Принцип роботи розробленого прототипу системи безпеки полягає у циклічній обробці сигналів, що надходять від підключених сенсорних модулів та реагуванні на виявлені події, які можуть становити потенційну загрозу. Центральним керуючим елементом є мікроконтролер ESP32-CAM, що відповідає за зчитування даних з сенсорів, обробку отриманої інформації, прийняття рішень щодо тривоги та ініціювання процедури сповіщення, включаючи фотофіксацію та передачу даних.

Загальний алгоритм функціонування системи можна представити наступними етапами (рис.3.1):

1) ініціалізація системи: після подачі живлення відбувається процес ініціалізації компонентів, підключення до Wi-Fi та налаштування контактів GPIO як джерел зовнішнього переривання. Після цього система надсилає повідомлення про готовність до роботи;
2) моніторинг стану сенсорів: після завершення етапу ініціалізації система переходить у режим безперервного моніторингу. Мікроконтролер здійснює періодичне зчитування даних з усіх підключених сенсорів;
3) виявлення джерела тривоги: дані, отримані від сенсорів, аналізуються відповідно до зміни їхнього логічного стану. Для PIR-сенсора, магнітного датчика МС-38 та цифрового виходу (D0) датчика газу MQ-6 система відстежує перехід сигналу з низького рівня на високий (або навпаки, залежно від логіки спрацювання). Така зміна стану розпізнається системою як небезпечна подія;
4) реагування на тривогу: у відповідь на виявлену загрозу мікроконтролер ініціює такі дії:

а) активація камери: активується вбудована камера ESP32-CAM та захоплюється фотографічний знімок контрольованої зони;

б) формування повідомлення: генерується текстове повідомлення, що містить інформацію про тип спрацьованого датчика та час події;

в) передача даних: встановлюється з'єднання з Telegram Bot API [12] через мережу Wi-Fi та здійснюється передача сформованого текстового повідомлення та зображення до Telegram-бота, що надсилає їх користувачеві;

5) повернення в режим очікування: після завершення процедури сповіщення система повертається в режим безперервного моніторингу стану сенсорів, очікуючи на нові події.

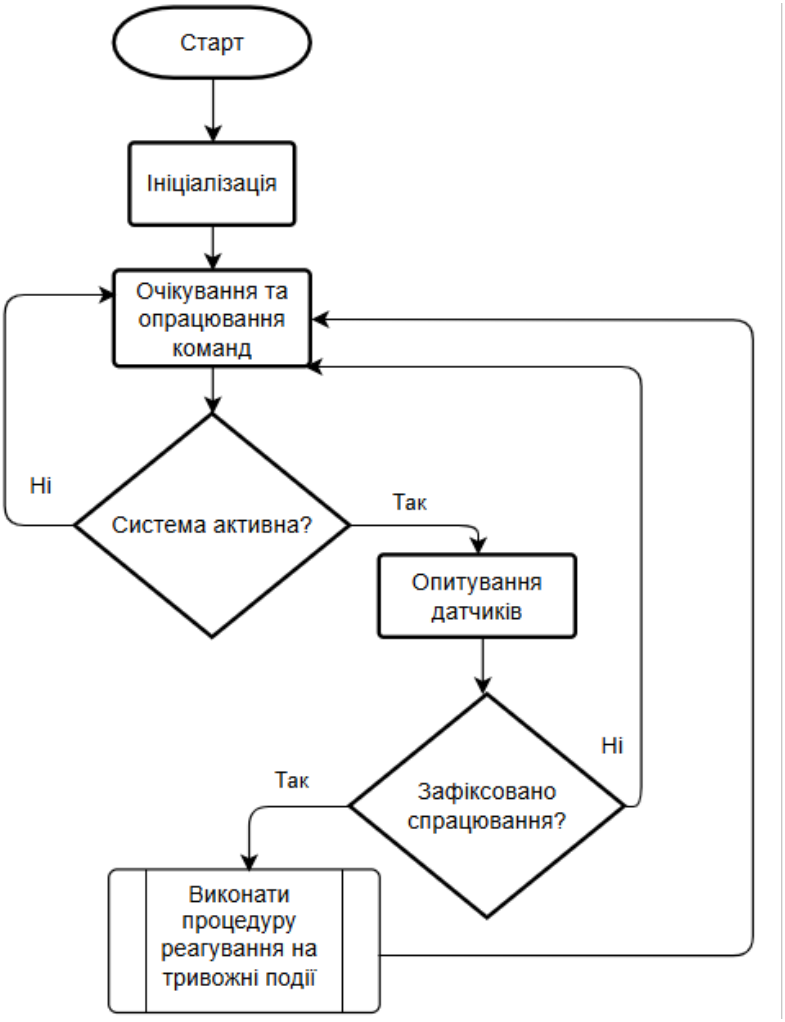


Рис. 3.1. Блок-схема головного циклу роботи системи

Основним елементом логіки системи є процедура реагування на небезпеку, яка активується при спрацюванні будь-якого з датчиків. Детальний алгоритм цієї процедури представлено на блок-схемі (рис.3.2).

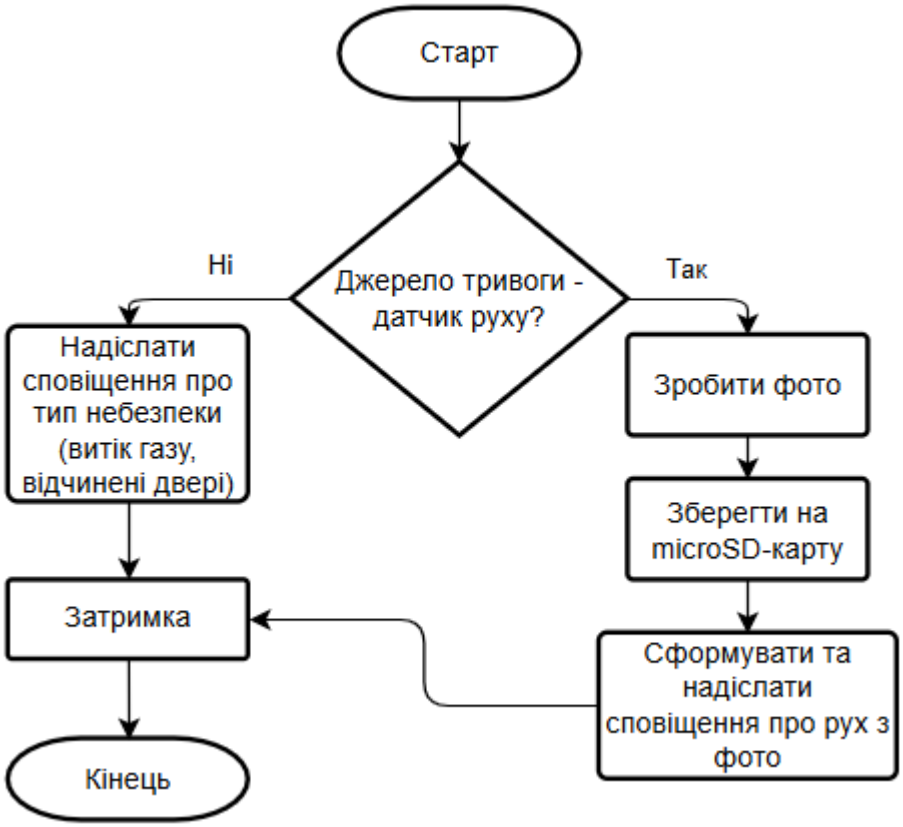


Рис. 3.2. Блок-схема алгоритму реагування на тривожні події

Для забезпечення ефективного та диференційованого реагування на різні типи загроз було розроблено спеціалізований алгоритм. Процедура активується в момент, коли один із сенсорів системи фіксує подію, що виходить за межі норми. Ключовим етапом алгоритму є першочергова перевірка джерела тривоги. Якщо подія спричинена спрацюванням датчика руху (PIR), виконується гілка, що реалізує функцію візуальної верифікації: система робить фотографію, зберігає її на локальний носій, після чого формує та надсилає користувачеві сповіщення, яке містить текстову інформацію про виявлення руху та зроблене фото.

У випадку, якщо джерелом тривоги є інший сенсор, наприклад, датчик витоку газу або геркон на дверях, виконується сценарій, коли система надсилає користувачеві текстове сповіщення із зазначенням конкретного типу небезпеки.

Важливо зазначити, що обидві гілки виконання алгоритму сходяться на фінальному етапі, де реалізовано програмну затримку. Це необхідно для уникнення надсилання повторних повідомлень в разі тривалої тривожної події. Після завершення всіх кроків процедура закінчується, і система повертається в режим очікування наступних подій.

Запропонований алгоритм забезпечує мінімально можливий час затримки між виявленням потенційно небезпечної події та інформуванням користувача. Це є критично важливим для своєчасного реагування, оскільки дозволяє запобігати збиткам та мінімізувати негативні наслідки інцидентів. Швидке сповіщення надає можливість оперативно вжити необхідні заходи, що суттєво підвищує загальну ефективність та надійність системи.

3.1.2. Схема підключення компонентів

Ефективність функціонування системи безпеки безпосередньо залежить від коректного підключення та взаємодії всіх її апаратних складових. На рисунку 3.3 представлена структурна схема системи безпеки, що детально ілюструє фізичні з'єднання між центральним мікроконтролером ESP32-CAM та периферійними сенсорними модулями.

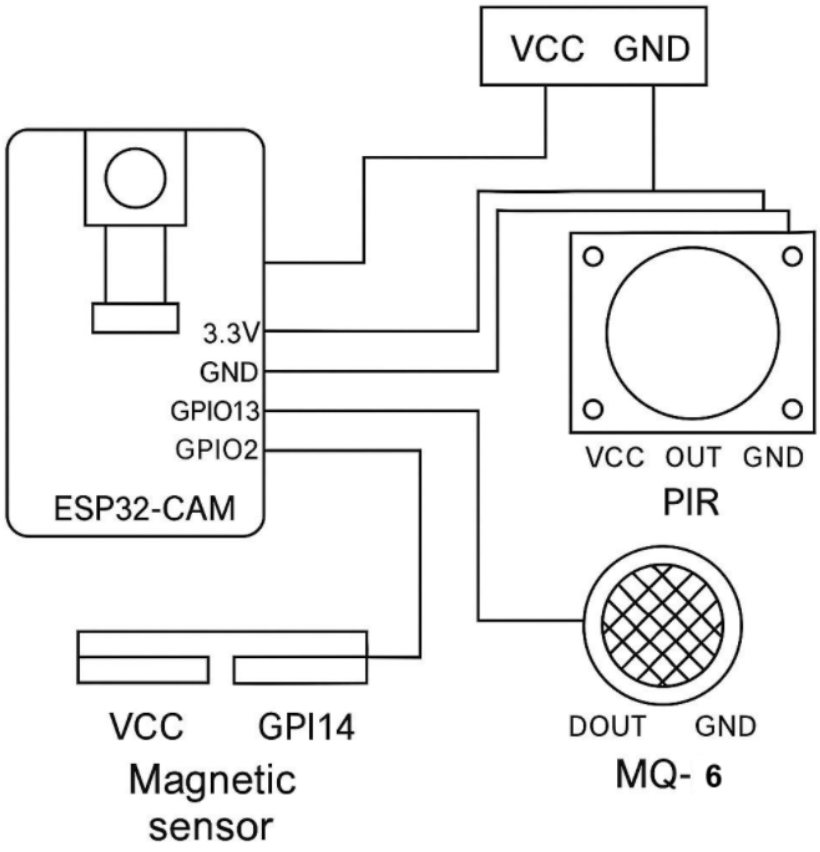


Рис. 3.3. Структурна схема системи безпеки

Згідно з представленою схемою, сенсорні модулі інтегровані з ESP32-CAM за допомогою виділених портів вводу-виводу (GPIO).Кожен тип сенсора підключається до відповідного типу порту, що забезпечує оптимальне зчитування даних [13]:

1) пасивний інфрачервоний (PIR) сенсор руху (HC-SR505) підключений до цифрового піна GPIO13. При виявленні руху сенсор змінює логічний рівень, що фіксується мікроконтролером;
2) цифровий вихід газового сенсора MQ-6 підключений до цифрового піна GPIO2. Це дозволяє миттєво фіксувати бінарний сигнал, коли концентрація газу перевищує порогове значення. Сам поріг чутливості попередньо налаштовується вручну за допомогою потенціометра, вбудованого в модуль сенсора;
3) для контролю відкриття дверей або вікон магнітний контактний датчик MC-38 підключений до цифрового піна GPIO14. Зміна стану контакту

геркона (замикання/розмикання), що відбувається при зміні взаємного положення магніту та датчика, фіксується мікроконтролером як бінарна подія.

Живлення всіх компонентів здійснюється через відповідні лінії VCC (напруга живлення) та GND (заземлення). Центральний мікроконтролерний модуль ESP32-CAM працює від джерела постійної напруги 5В, яка через вбудований стабілізатор знижується до необхідних для роботи 3.3В. Оскільки деякі периферійні сенсори, зокрема MQ-6, потребують живлення 5В, а також узгодження логічних рівнів між компонентами з різною напругою, у систему інтегровано модуль перетворювача Logic Level Shifter. Це забезпечує коректну взаємодію та захист усіх елементів апаратного комплексу від електричних перевантажень і потенційних пошкоджень, що могли б виникнути через несумісність рівнів напруги.

3.2. Апаратна та програмна реалізація прототипу

На етапі апаратної реалізації було створено функціональний фізичний прототип системи безпеки. Основою пристрою є мікроконтролер ESP32-CAM, до якого згідно з розробленою структурною схемою(рис.2.1) підключено основні сенсори: PIR HC-SR505 для детекції руху, газовий сенсор MQ-6 для виявлення горючих газів та магнітний контактний датчик MC-38 для контролю відчинення дверей.

Компоненти були змонтовані на макетній платі з урахуванням компактності та зручності подальшого розміщення в захисному корпусі. Для забезпечення стабільного живлення та проведення комплексного тестування системи використовувався лабораторний блок живлення, що дозволило імітувати різні сценарії роботи, включно з автономним режимом від акумулятора.

Вибір компонентів системи здійснювався на основі аналізу технічних характеристик, сумісності з мікроконтролером ESP32-CAM, функціональності

та енергоефективності. Застосування сенсора руху HC-SR505 забезпечує надійне виявлення присутності людини у контрольованій зоні за мінімальних енерговитрат. Газовий сенсор MQ-6 критично важливий для попередження про небезпеку пожежі чи вибуху, а датчик MC-38 є простим та ефективним засобом контролю доступу.

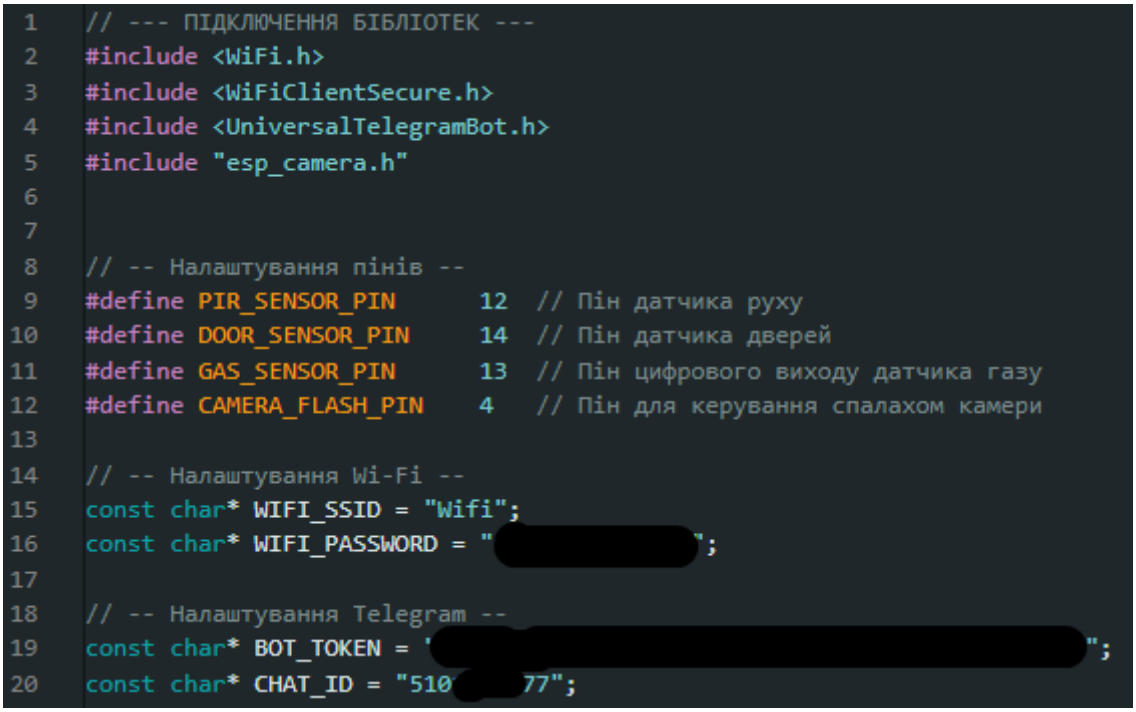
Комплексне використання зазначених сенсорів у поєднанні з обчислювальними та функціональними можливостями ESP32-CAM дозволило створити ефективну, масштабовану та бюджетну систему безпеки.

Основні технічні характеристики використаних апаратних компонентів наведено в таблиці 3.1.

Табл. 3.1. Технічні характеристики апаратних компонентів прототипу

Компонент	Характеристики	Напруга живлення	Інтерфейс	Основна функція
Мікроконтролер ESP32-CAM	Двоядерний процесор Xtensa LX6, вбудовані Wi-Fi, Bluetooth, камера OV2640.	5 В (живлення) 3.3 В (логіка)	GPIO, ADC, UART	Централізоване керування системою, обробка та передача даних з сенсорів.
Датчик руху PIR HC-SR505	Кут огляду ~110°, дальність до 3 м.	5 В	Цифровий вихід	Детектування руху в зоні, що охороняється.
Газовий сенсор MQ-6	Висока чутливість до LPG, пропану, бутану, метану.	5 В	Цифровий вихід (D0)	Виявлення витоку горючих газів для попередження про небезпеку.
Датчик відкриття дверей/вікон MC-38	Магнітний контактний датчик.	3.3 - 5 В	Цифровий вихід	Контроль стану дверей/вікон (відкрито/закрито) .

Програмна частина проєкту була реалізована в середовищі Arduino IDE з використанням мови C++ та об'єктно-орієнтованого підходу. Такий підхід дозволив створити модульну, гнучку та добре читабельну архітектуру. Нижче наведено аналіз основних фрагментів коду.



```
// --- ПІДКЛЮЧЕННЯ БІБЛІОТЕК ---
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include "esp_camera.h"


// -- Налаштування пінів --
#define PIR_SENSOR_PIN      12  // Пін датчика руху
#define DOOR_SENSOR_PIN     14  // Пін датчика дверей
#define GAS_SENSOR_PIN      13  // Пін цифрового виходу датчика газу
#define CAMERA_FLASH_PIN    4   // Пін для керування спалахом камери

// -- Налаштування Wi-Fi --
const char* WIFI_SSID = "Wifi";
const char* WIFI_PASSWORD = "[redacted]";

// -- Налаштування Telegram --
const char* BOT_TOKEN = "[redacted]";
const char* CHAT_ID = "510[redacted]77";
```

Рис. 3.4. Блок конфігурації

На початку програми знаходиться блок, відповідальний за конфігурацію системи. Винесення всіх налаштувань дозволяє легко адаптувати проєкт під різні апаратні конфігурації та умови експлуатації без втручання в основну логіку. У цьому блоці визначаються номери пінів мікроконтролера, до яких підключені датчики, облікові дані для з'єднання з мережею Wi-Fi, а також унікальні ідентифікатори для роботи з API месенджера Telegram.

Також задаються операційні параметри, такі як тривалість періоду сну для перевірки датчика газу та час активного прослуховування команд після пробудження.

Було оголошено три основні класи: CameraHandler, Notifier та SecurityController, кожен з яких виконує чітко визначену роль. CameraHandler інкапсулює всю логіку роботи з камерою та спалахом. Notifier відповідає за всю мережеву взаємодію та відправку сповіщень. SecurityController виступає в ролі головного керуючого класу, що координує роботу інших об'єктів та реалізує основну логіку.

```
// Оголошення класів
class CameraHandler;
class Notifier;
class SecurityController;

// Створення глобальних вказівників на об'єкти
CameraHandler* camera;
Notifier* notifier;
SecurityController* securitySystem;

// --- Клас для керування камерою та спалахом ---
class CameraHandler {
private:
    int _flash_pin;
public:
    CameraHandler(int flash_pin);
    bool setup();
    void setFlash(bool state);
    camera_fb_t* capturePhoto();
    void returnPhotoBuffer(camera_fb_t* fb);
};

// --- Клас для відправки сповіщень ---
class Notifier {
private:
    WiFiClientSecure _secured_client;
    UniversalTelegramBot _bot;
    const char* _chat_id;
public:
    Notifier(const char* token, const char* chat_id);
    bool connectWifi(const char* ssid, const char* password);
    void sendMessage(String message);
    void sendPhotoWithCaption(camera_fb_t* fb, String caption);
    bool getNewMessage(UniversalTelegramBot::telegramMessage &msg)
};

class SecurityController {
private:
    Notifier* _notifier;
    CameraHandler* _camera;
    void handleWakeupEvent();
    void listenForCommands(int seconds);
    void processCommand(String command);
    String getStatus();
    void goToSleep();
    void handleMotionTrigger();
    void handleDoorTrigger();
    void handleGasCheck();
public:
    SecurityController(Notifier* notifier, CameraHandler* camera);
    void setupSystem();
    void handleSystem();
};
```

Рис. 3.5. Оголошення класів

Клас Notifier (рис.3.6) реалізує патерн "Фасад", виступаючи єдиним інтерфейсом для взаємодії з мережею та сервісом сповіщень Telegram. Він повністю інкапсулює складну логіку, пов'язану з підключенням до Wi-Fi, встановленням захищеного HTTPS-з'єднання та формуванням HTTP-запитів до

Telegram Bot API. Такий підхід забезпечує високий рівень абстракції та слабку зв'язаність між компонентами системи. Наприклад, у разі потреби змінити сервіс сповіщень з Telegram на Email, достатньо буде модифікувати лише реалізацію класу Notifier, не вносячи жодних змін до основної логіки системи. Це значно спрощує подальший розвиток та підтримку проєкту.

```
157  Notifier::Notifier(const char* token, const char* chat_id)
158      : _bot(token, _secured_client), _chat_id(chat_id) {}
159
160  bool Notifier::connectWifi(const char* ssid, const char* password) {
161      WiFi.begin(ssid, password);
162      Serial.print("Підключення до Wi-Fi");
163      int attempts = 0;
164      while (WiFi.status() != WL_CONNECTED && attempts < 20) {
165          Serial.print(".");
166          delay(500);
167          attempts++;
168      }
169      if (WiFi.status() == WL_CONNECTED) {
170          Serial.println("\nWi-Fi підключено.");
171          return true;
172      }
173      Serial.println("\nНе вдалося підключитись до Wi-Fi.");
174      return false;
175  }
176
177  void Notifier::sendMessage(String message) {
178      _bot.sendMessage(_chat_id, message, "");
179  }
180
181  void Notifier::sendPhotoWithCaption(camera_fb_t* fb, String caption) {
182      if (!fb) return;
183      _bot.sendPhoto(_chat_id, fb->buf, fb->len, caption, "image/jpeg");
184  }
185
186  bool Notifier::getNewMessage(UniversalTelegramBot::telegramMessage &msg) {
187      int numNewMessages = _bot.getUpdates(_bot.last_message_received + 1);
188      if (numNewMessages > 0) {
189          msg = _bot.messages[0];
190          _bot.last_message_received = msg.update_id;
191          return true;
192      }
193      return false;
194  }
```

Рис. 3.6. Клас Notifier

Клас SecurityController (рис.3.7) є центральним керуючим вузлом системи, що реалізує основну бізнес-логіку та патерн "Стан". Він відповідає за керування життєвим циклом пристрою: сон, пробудження, аналіз причини пробудження, виконання відповідних дій та повернення до сну. Цей клас виступає в ролі оркестратора, координуючи роботу інших об'єктів.

Особливістю є імплементація енергоефективної логіки на основі режиму глибокого сну (Deep Sleep). Це забезпечує довготривалу автономну роботу пристрою, що є критичною вимогою для систем безпеки. Він аналізує причину пробудження системи — чи це був сигнал від одного з датчиків, чи спрацювання таймера і делегує обробку відповідному методу.

```
207 > void SecurityController::handleSystem() {···
221   }
222
223   void SecurityController::handleWakeupEvent() {
224       Serial.println("Пробудження від події...");
225       if (!_notifier->connectWifi(WIFI_SSID, WIFI_PASSWORD)) {
226           Serial.println("Немає Wi-Fi, засинаю...");
227           goToSleep();
228           return;
229       }
230
231       if (!is_system_armed) {
232         Serial.println("Система не активована. Подію проігноровано.");
233         return;
234       }
235
236       esp_sleep_wakeup_cause_t wakeup_reason = esp_sleep_get_wakeup_cause();
237
238       if (wakeup_reason == ESP_SLEEP_WAKEUP_EXT1) {
239           uint64_t wakeup_pin_info = esp_sleep_get_ext1_wakeup_status();
240           if (wakeup_pin_info & (1ULL << PIR_SENSOR_PIN)) {
241               handleMotionTrigger();
242           }
243           if (wakeup_pin_info & (1ULL << DOOR_SENSOR_PIN)) {
244               handleDoorTrigger();
245           }
246       } else if (wakeup_reason == ESP_SLEEP_WAKEUP_TIMER) {
247           handleGasCheck();
248       }
249   }
250
251   void SecurityController::handleMotionTrigger(){
252       Serial.println("Подія: Виявлено рух!");
253 >     if(!_camera->setup()){···
256       }
257       _notifier->sendMessage("УВАГА! Виявлено рух!");
258       _camera->setFlash(true);
259       delay(500);
260       camera_fb_t* fb = _camera->capturePhoto();
261       _camera->setFlash(false);
262       if (fb) {
263           _notifier->sendPhotoWithCaption(fb, "Фото з місця події");
264           _camera->returnPhotoBuffer(fb);
265 >     } else {···
267       }
268   }
269
270   void SecurityController::handleDoorTrigger(){
271       Serial.println("Подія: Відчинено двері!");
272       _notifier->sendMessage("УВАГА! Спрацював датчик відчинення дверей!");
```

Рис. 3.7. Фрагмент класу SecurityController

Стандартні для середовища Arduino IDE [14] функції setup() та loop() використовуються для ініціалізації та запуску системи. У функції setup()

відбувається створення екземплярів основних класів та виклик методів для початкового налаштування апаратних пінів та запуску головної логіки контролера. Порожня функція loop() є ключовою особливістю реалізованого енергоефективного підходу. Система не виконує постійне опитування датчиків у циклі. Замість цього, після виконання всіх дій при пробудженні, вона негайно переходить у режим глибокого сну. Це кардинально знижує енергоспоживання та є головною вимогою для автономних пристроїв.

```
92  void setup() {
93      //  Екземпляри класів
94      camera = new CameraHandler(CAMERA_FLASH_PIN);
95      notifier = new Notifier(BOT_TOKEN, CHAT_ID);
96      securitySystem = new SecurityController(notifier, camera);
97
98      // Запуск налаштування системи
99      securitySystem->setupSystem();
00      // Запуск головної логіки
01      securitySystem->handleSystem();
02  }
03
04  void loop() {
05
06  }
```

Рис. 3.8. Життєвий цикл системи

3.3. Інструкція з користування та сценарії використання

Взаємодія користувача з розробленою системою максимально проста та інтуїтивно зрозуміла. Керування системою та отримання сповіщень здійснюється через месенджер Telegram, що є звичним для більшості користувачів. Нижче описано основні команди та алгоритми, доступні для керування системою та отримання інформації.

Ініціалізація системи

Для першого запуску та активації системи достатньо підключити пристрій до джерела живлення 5В, яким може слугувати стандартний мережевий адаптер для смартфона або портативний акумулятор. Після цього система автоматично ініціалізується, підключається до мережі Wi-Fi, надсилає в Telegram вітальне

повідомлення про готовність до роботи та переходить у режим очікування прийому команд.

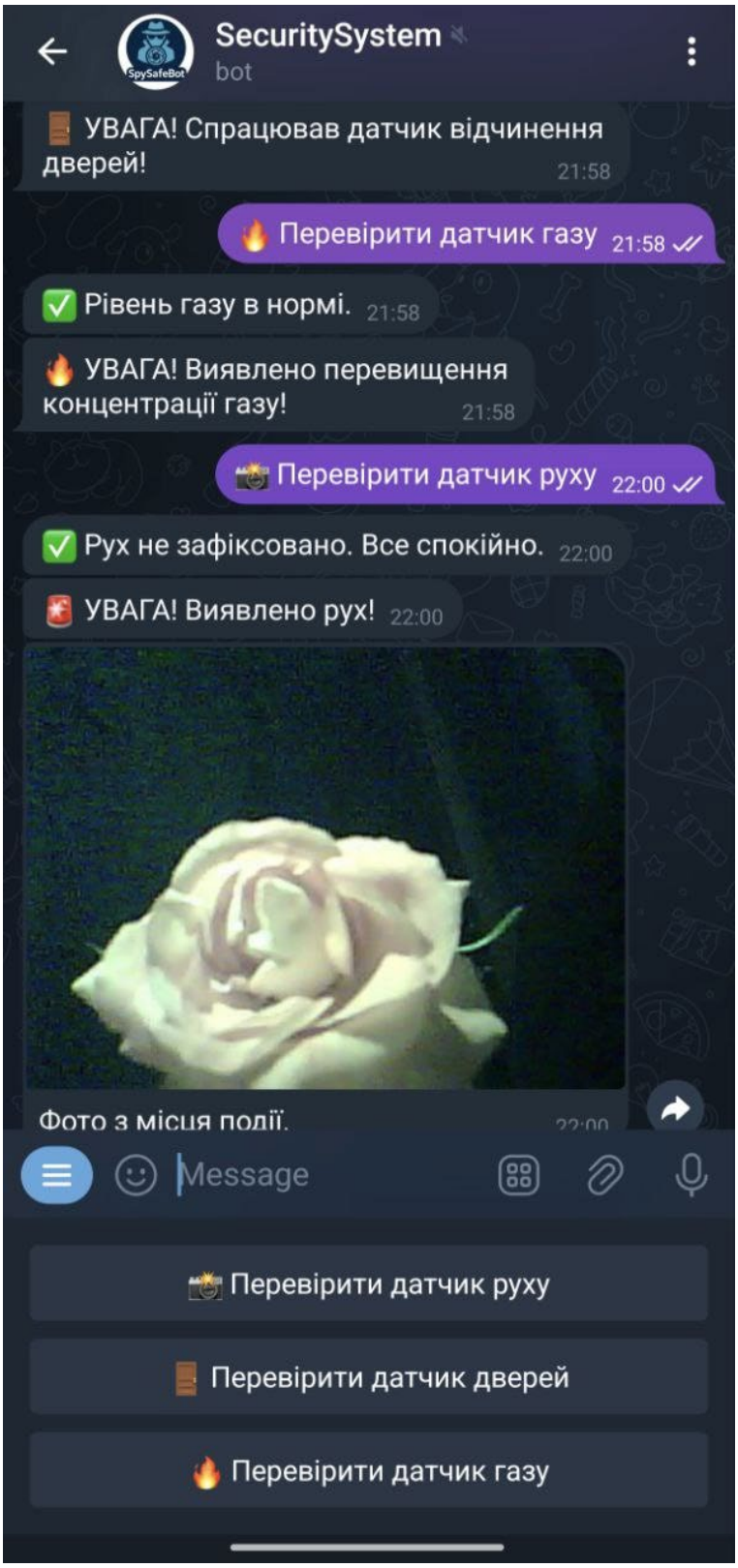


Рис. 3.9. Взаємодія користувача з Telegram-ботом системи безпеки

Активація та деактивація моніторингу:

а) для переведення системи в активний режим охорони необхідно надіслати боту команду /on. Після цього система розпочне моніторинг підключених сенсорів;

б) щоб вимкнути режим охорони (наприклад, під час перебування власника на об'єкті) потрібно надіслати команду /off.

Перевірка поточного стану

Щоб отримати інформацію про поточний режим роботи системи, користувачу достатньо надіслати команду /status.

Отримання сповіщень

Коли система знаходиться в активному режимі, вона автоматично надсилає сповіщення у разі виявлення загроз:

1) рух: при спрацюванні PIR-сенсора користувач отримує текстове повідомлення про детекцію руху та фото з місця події.
2) витік газу: у разі фіксації сенсором MQ-6 перевищення безпечної концентрації горючих газів, користувач отримує текстове попередження;
3) несанкціонований доступ: при розмиканні герконів, встановлених на дверях або вікнах, користувач отримує сповіщення, що сигналізує про спробу несанкціонованого вторгнення.

Розроблена система безпеки є гнучким рішенням, що може бути застосоване в широкому спектрі реальних сценаріїв, де потрібен базовий рівень моніторингу та оперативного оповіщення. Нижче представлено типові приклади використання, що ілюструють практичну цінність та адаптивність системи:

1) Забезпечення безпеки віддалених об’єктів:

   Приклад: Дачні будинки або заміські котеджі. У період відсутності власників система контролює приміщення, виявляючи рух всередині та

миттєво сповіщає власника фотографією. Це дозволяє оперативно реагувати на загрози, навіть перебуваючи далеко від об’єкта.

2) Контроль за об’єктами під час тимчасової відсутності власників:

   Приклад: Квартири або будинки під час ремонту. Власник, який тимчасово проживає в іншому місці, може використовувати систему для моніторингу активності будівельної бригади в робочий час та для запобігання несанкціонованому доступу у позаробочий час.

3) Захист невеликих комерційних або громадських об’єктів:

   Приклад: Невеликі офіси, магазини, склади або майстерні. У неробочий час або у період відсутності персоналу система може фіксувати спроби несанкціонованого доступу та виявляти витік газу, оперативно повідомляючи власника про нештатні ситуації, що дозволяє уникнути значних фінансових втрат.

4) Моніторинг допоміжних споруд:

   Приклад: Літні кухні, гаражі, господарські будівлі. Система може забезпечити базовий рівень захисту для таких об’єктів, які часто залишаються поза зоною уваги основних систем безпеки, але нерідко є сховищем для цінного майна або можуть представляти потенційні небезпеки (наприклад, витік газу в котельні або виникнення пожежі).

Наведені приклади застосування демонструють універсальність та практичну цінність розробленої автономної системи безпеки, підкреслюючи її здатність надавати ефективні рішення для різних потреб у сфері моніторингу та оперативного оповіщення про небезпеку.

3.4. Випробування та результати тестування

Для перевірки працездатності розробленого прототипу було проведено комплексне тестування, спрямоване на оцінку стабільності роботи системи, коректності функціонування при виявленні різних типів подій, а також

надійності та швидкості передачі сповіщень через месенджер Telegram у режимі реального часу.

Методика випробувань:

1) PIR-сенсор (HC-SR505): перевірка чутливості сенсора здійснювалася шляхом фіксації спрацювання при появі людини на відстані до 3 метрів. Додатково фіксувалася стабільність виявлення руху при різній швидкості переміщення об’єкта;
2) газовий сенсор MQ-6: тестування проводилося з використанням контрольованого джерела горючого газу (пропан з портативної запальнички на безпечній відстані). Фіксувався час реакції від моменту появи газу до спрацювання цифрового виходу модуля та генерації відповідного сигналу тривоги мікроконтролером;
3) магнітоконтактний сенсор MC-38: перевірялася миттєвість спрацювання при фізичному роз'єднанні магнітної та контактної частин датчика, що імітувало відчинення дверей;
4) передача сповіщень: для кожного тестового сценарію фіксувався час між моментом фіксації події системою та отриманням відповідного повідомлення в Telegram при стабільному Wi-Fi з'єднанні.

Результати тестування:

1) час реакції системи: середній час від моменту події до отримання сповіщення в Telegram склав 2–4 секунди, що є прийнятним показником для систем безпеки базового рівня. Затримка переважно зумовлена часом захоплення зображення та його подальшою передачею;
2) надійність передачі сповіщень: при стабільному Wi-Fi з'єднанні надійність доставки повідомлень (як текстових, так і фотографій) склала 100% у всіх проведених тестах. Це підкреслює критичну важливість стабільного інтернет-підключення для повноцінного та безперебійного функціонування системи;

3) відсутність хибних спрацювань: протягом серії з понад 20 тестових запусків хибних детекцій загроз не було зафіксовано. Це свідчить про коректну роботу алгоритмів обробки даних з сенсорів та адекватно налаштовані пороги чутливості.

Отримані результати підтверджують працездатність розробленого прототипу системи безпеки та його відповідність заявленим вимогам і функціоналу. Система продемонструвала стабільну роботу, прийнятний час реакції на події та надійну передачу сповіщень. Це дозволяє рекомендувати її для використання в побутових умовах та для потреб малого бізнесу як доступне та ефективне рішення для базового моніторингу та охорони.

3.5. Перспективи розвитку

На основі проведеного аналізу та з урахуванням сучасних тенденцій розвитку IoT-систем можна визначити такі пріоритетні напрямки для подальшої роботи:

1. Підвищення рівня кібербезпеки:

   а) перехід на HTTPS: використання бібліотек, що підтримують TLS-шифрування для всіх запитів до Telegram API, щоб унеможливити перехоплення даних;

   б) шифрування даних на microSD-карті: впровадження алгоритмів шифрування для файлів, що зберігаються на карті пам'яті. Це захистить дані навіть у разі крадіжки пристрою;

   в) безпечне оновлення ПЗ (Secure OTA): використання механізмів з цифровим підписом прошивки для запобігання завантаженню шкідливого коду під час віддаленого оновлення.
2. Впровадження інтелектуального аналізу даних:

   а) локальне розпізнавання об’єктів: впровадження моделей комп’ютерного зору з використанням фреймворків TensorFlow або Edge Impulse [15]. Це дозволить значно зменшити кількість хибних спрацювань, наприклад від домашніх тварин;

б) аналіз аномальної поведінки: використання алгоритмів машинного навчання для аналізу послідовності спрацювань сенсорів з метою виявлення нетипових патернів, що можуть свідчити про загрозу.

3. Інтеграція з системами "розумного дому":

   Впровадження підтримки протоколу MQTT: інтеграція MQTT-клієнта забезпечить взаємодію з популярними платформами "розумного дому", як Home Assistant, Domoticz чи openHAB для створення складних сценаріїв автоматизації. Наприклад, при виявленні витоку газу система дає команду "розумному" реле перекрити газовий клапан.

4. Покращення апаратної частини:

   Додавання невеликої сонячної панелі та контролера заряду для забезпечення повної енергетичної автономності пристрою.

Реалізація цих удосконалень дозволить перетворити розроблений прототип на повноцінний, конкурентоспроможний та безпечний інтелектуальний продукт, що відповідає сучасним вимогам ринку Інтернету речей.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було успішно спроєктовано, реалізовано та протестовано прототип автономної системи безпеки на базі мікроконтролера ESP32-CAM з інтеграцією через месенджер Telegram. Створений програмно-апаратний комплекс продемонстрував свою ефективність у виконанні ключових функцій: детекції руху, контролю несанкціонованого доступу, моніторингу витоку горючих газів, а також візуальної верифікації подій з подальшим миттєвим сповіщенням користувача.

Основними перевагами розробленого рішення є економічна доступність компонентів, низьке енергоспоживання, що дозволяє тривалу автономну роботу та архітектурна гнучкість. Проведений порівняльний аналіз підтверджує конкурентоспроможність прототипу відносно комерційних та DIY-аналогів, особливо за критеріями незалежності від сторонніх хмарних інфраструктур та можливості розширеної кастомізації.

Експериментальні випробування підтвердили функціональну надійність систем. Прототип продемонстрував стабільну роботу та прийнятний час реакції, що в середньому становить 2-4 секунди від моменту фіксації події до отримання сповіщення. Ці результати обґрунтовують практичну значущість розробки для забезпечення базового рівня безпеки житлових приміщень, невеликих офісів, гаражів та інших об'єктів, де пріоритетом є економічність та простота експлуатації.

Науково-технічний внесок роботи полягає в розробці та практичному обґрунтуванні гнучкої архітектури для створення компактних систем безпеки, що поєднують низьке енергоспоживання з економічною доступністю. На відміну від існуючих рішень, запропонований підхід забезпечує повний контроль над даними та логікою роботи, незалежність від платних хмарних сервісів та легку масштабованість.

Водночас, створена система має значний потенціал для подальшого вдосконалення. У межах роботи було проаналізовано та визначено перспективні напрями розвитку, що включають підвищення рівня кібербезпеки, впровадження алгоритмів інтелектуального аналізу даних, подальшу оптимізацію енергоспоживання та інтеграцію з системами "розумного дому".

Таким чином, запропонована система на базі ESP32-CAM є ефективним та перспективним рішенням для побудови доступних та гнучких засобів захисту відповідно до індивідуальних потреб користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sheikh R. U., Kale P. A., Sarfaraj S., Sukhdeve S. A Review on IoT Based Smart Security And Home Automation // *International Journal of Scientific Research in Science and Technology.* – 2021. – Т. 8, № 3. – С. 700–705. DOI: 10.32628/IJSRST2183159.
2. Improvement in Electro-Magnetic Alarms: пат. 9802 США / винахідник A. R. Pope. – Опубл. 21.06.1853. [Електронний ресурс]. – Режим доступу: https://patents.google.com/patent/US9802A/en (дата звернення: 13.05.2025).
3. Kirschenbaum K. Alarm Industry History: Who Was Augustus Russel Pope? // Kirschenbaumesq.com. – 2020. – [Електронний ресурс]. – Режим доступу: https://www.kirschenbaumesq.com/article/alarm-industry-history-who-was-augustus-russel-pope-part-6-november-18-2020 (дата звернення: 13.05.2025).
4. Головчук Ю.Д., Каштан С.С. Цифрова трансформація промисловості. Інформаційні технології і автоматизація – 2024: Матеріали XVII Міжнародної науково-практичної конференції (31 жовтня – 1 листопада 2024 р., м.Одеса). Одеса: ОНТУ, 2024. С. 595-598. URL: https://www.ontu.edu.ua/download/konfi/2024/Collection-of-abstracts-of-the-conference-ITIA-2024.pdf
5. ESP32-S Datasheet, Version 1.1 // Espressif Systems. – 2021. – [Електронний ресурс]. – Режим доступу: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (дата звернення: 12.04.2025).
6. How PIRs Work // Adafruit Learning System [Електронний ресурс]. – Режим доступу: https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work (дата звернення: 12.04.2025).
7. MC-38 Magnetic Switch Sensor // WAT Electronic [Електронний ресурс]. – Режим доступу: https://www.watelectronics.com/mc38-magnetic-switch-sensor/ (дата звернення: 12.04.2025).

8. MQ-6 LPG Sensor Module. Datasheet. [Електронний ресурс]. – Режим доступу: https://www.handsontec.com/dataspecs/sensor/MQ6-LPG%20Sensor%20Module.pdf (дата звернення: 15.04.2025).
9. MQ-6 Semiconductor Sensor for LPG Datasheet. [Електронний ресурс]. – Режим доступу: https://www.winsen-sensor.com/d/files/semiconductor/mq-6.pdf (дата звернення: 15.04.2025).
10. Rahaman M. M. A Review on Internet of Things (IoT): Architecture, Technologies, Future Applications & Challenges // *International Journal of Science and Business*. – 2024. – Т. 14, № 1. – С. 80–92. DOI: 10.5281/zenodo.7066810.
11. Ajax Systems. Технологія Jeweller. [Електронний ресурс]. – Режим доступу: https://ajax.systems/ua/jeweller/ (дата звернення: 13.05.2025).
12. Telegram Bot API [Електронний ресурс]. – Режим доступу: https://core.telegram.org/bots/api (дата звернення: 10.05.2025).
13. ESP32-CAM Official Documentation [Електронний ресурс]. – Режим доступу: https://randomnerdtutorials.com (дата звернення: 10.05.2025).
14. Офіційний сайт Arduino. [Електронний ресурс]. – Режим доступу: https://www.arduino.cc/ (дата звернення: 13.05.2025).
15. Al-Mutawa R. F., Albouraey F. A Smart Home System based on Internet of Things // *International Journal of Advanced Computer Science and Applications*. – 2020. – Т. 11, № 2. – С. 24–29. DOI: 10.14569/IJACSA.2020.0110234.