

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Таврійський національний університет

імені В.І. Вернадського

Навчально-науковий інститут муніципального управління

та міського господарства

Кафедра інженерних систем та технологій

На правах рукопису

Кваліфікаційна робота на здобуття ступеня вищої освіти «бакалавр»

ТЕМА РОБОТИ

АВТОМАТИЗОВАНА СИСТЕМА УПРАВЛІННЯ БЕЗПЕКОЮ БУДИНКУ

Здобувача вищої освіти
Федоренко Івана Сергійовича
Освітня програма «Автоматизація та
комп'ютерно-інтегровані технології»
(Спеціальність 174 «Автоматизація та
комп'ютерно-інтегровані технології»)

(підпис)

Науковий керівник: старший
викладач Вишемірська Я.С.

(підпис)

Національна шкала _____
Кількість балів _____
Оцінка: ECTS _____

Таврійський національний університет імені В. І. Вернадського
Навчально-науковий інститут муніципального управління
та міського господарства
Кафедра інженерних систем та технологій
Рівень вищої освіти перший (бакалаврський)
Спеціальність 174 «Автоматизація та комп'ютерно-інтегровані технології»

«ЗАТВЕРДЖУЮ»
Завідувач кафедри
Наталія ОМЕЦІНСЬКА
«_____» _____ 2026 року

ЗАВДАННЯ НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

1. Тема роботи: Автоматизована система управління безпекою будинку
керівник роботи: старший викладач Вишемірська Я.С.
затверджені Наказом ТНУ імені В.І. Вернадського:
від «7» листопада 2025 № 138/1-ОД

1. Строк подання студентом роботи: _____

0. Вихідні дані до роботи: нормативні вимоги до побудови та експлуатації систем безпеки житлових будинків, електробезпеки, пожежної безпеки та захисту в надзвичайних ситуаціях; теоретичні відомості щодо концепції «Розумний будинок» і технологій Інтернету речей; дані про сучасні охоронні системи, системи контролю доступу, пожежного сповіщення, відеоспостереження та моніторингу інженерних мереж; технічні характеристики мікроконтролера ESP32, датчиків руху, відкриття, диму, газу, затоплення, температури та вологості, а також виконавчих пристроїв системи; методи структурного, функціонального та UML-моделювання автоматизованих систем; алгоритми зчитування й оброблення сигналів із датчиків, виявлення охоронних та аварійних подій, керування виконавчими пристроями, формування повідомлень користувачу та журналювання подій; програмні засоби Arduino IDE / PlatformIO, мова програмування C/C++, засоби Wi-Fi та GSM-зв'язку для розроблення автоматизованої системи управління безпекою будинку.

4. Зміст розрахунково-пояснювальної записки: ВСТУП, РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ АВТОМАТИЗОВАНИХ СИСТЕМ БЕЗПЕКИ БУДИНКУ, РОЗДІЛ 2. ПРОЄКТУВАННЯ СТРУКТУРИ ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ БЕЗПЕКОЮ БУДИНКУ, РОЗДІЛ 3. РОЗРОБЛЕННЯ АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ БЕЗПЕКИ, РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ, ВИСНОВКИ, СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ, ДОДАТКИ.

0. Перелік графічного матеріалу: Графічна частина роботи виконана у вигляді мультимедійної презентації

0. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 1	Ст. викладач Вишемірська Я.С.		
Розділ 2	Ст. викладач Вишемірська Я.С.		
Розділ 3	Ст. викладач Вишемірська Я.С.		
Розділ 4	Ст. викладач Вишемірська Я.С.		

0. Дата видачі завдання 11 листопада 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи		Термін виконання етапів роботи	Заключний документ етапу
1	Вибір теми бакалаврської роботи, призначення керівника		до 31.10.2025	Затвердження переліку тем бакалаврських робіт
2	Пошук і відбір літератури по темі роботи, складання плану бакалаврської роботи		до 14.12.2025	Список літературних джерел, план роботи
3	Визначення об'єкта, предмета, мети, завдань та методів дослідження, написання вступу до теми бакалаврського дослідження		до 25.01.2026	Текст вступу
4	Написання тексту бакалаврської роботи відповідно до її структури:		26.01.2026-17.05.2026	Текст розділів
	4.1	I розділ	26.01.2026-6.03.2026	Текст розділу
	4.2	II розділ	9.03.2026-27.03.2026	Текст розділу
	4.3	III розділ	30.03.2026-17.04.2026	Текст розділу
	4.4	IV розділ	20.04.2026-8.05.2026	
5	Підготовка графічних матеріалів чи іншого уточнення		11.05.2026-15.05.2026	Роздатковий матеріал, презентація
6	Оформлення кінцевого списку використаних джерел та додатків		11.05.2026-15.05.2026	Список літературних джерел
7	Оформлення та попередній захист бакалаврської роботи		18.05.2026-22.05.2026	Бакалаврська робота, рішення кафедри про допуск до захисту
8	Внесення коректив та кінцеве редагування бакалаврської роботи		25.05.2026-29.05.2026	Бакалаврська робота
9	Реєстрація бакалаврської роботи на кафедрі		1.06.2026-5.06.2026	Бакалаврська робота внесена до журналу реєстрації випускових робіт
10	Захист бакалаврської роботи		10.06.2026-28.06.2026	Рішення Екзаменаційної комісії про захист

АНОТАЦІЯ

Федоренко І. С. Автоматизована система управління безпекою будинку – Рукопис.

Кваліфікаційна бакалаврська робота за спеціальністю 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка». – Таврійський національний університет імені В. І. Вернадського, Київ, 2026.

У бакалаврській роботі розглянуто процес проектування автоматизованої системи управління безпекою будинку на основі концепції «Розумний будинок» та технологій Інтернету речей. У роботі проведено аналіз предметної області автоматизованих систем безпеки житлових будинків, розглянуто сучасні охоронні системи, системи контролю доступу, відеоспостереження, пожежного сповіщення та багаторівневі підходи до забезпечення безпеки об'єкта.

У дослідженні обґрунтовано вибір технічних засобів системи, зокрема мікроконтролера ESP32, датчиків руху, відкриття дверей і вікон, диму, газу, затоплення, температури та вологості, а також виконавчих пристроїв для локального реагування на небезпечні події. Розроблено структурну схему автоматизованої системи безпеки, визначено її основні рівні, потоки даних і керуючих сигналів. Для формалізації взаємодії користувача та системи використано засоби UML-моделювання.

Практична частина роботи зосереджена на розробленні алгоритмічного та програмного забезпечення системи. Визначено функціональні вимоги до програмної частини, розроблено функціональну структуру, алгоритми оброблення охоронних та аварійних подій, структуру даних і програмну реалізацію для мікроконтролера ESP32. Програмне забезпечення забезпечує зчитування сигналів із датчиків, аналіз отриманих даних, визначення режиму роботи системи, активацію виконавчих пристроїв, формування повідомлень користувачу та журналювання подій.

Окрему увагу приділено питанням охорони праці, електробезпеки, пожежної безпеки та безпеки в надзвичайних ситуаціях під час монтажу й експлуатації автоматизованої системи безпеки будинку. Запропонована система дозволяє

підвищити рівень захищеності житлового об'єкта, забезпечити своєчасне виявлення небезпечних ситуацій і скоротити час реагування на охоронні та аварійні події.

Ключові слова: автоматизована система безпеки, розумний будинок, Інтернет речей, ESP32, датчики, охоронна сигналізація, пожежна безпека, GSM, Wi-Fi, UML-моделювання, програмне забезпечення, мікроконтролер.

ABSTRACT

Fedorenko I. S. Automated Home Security Management System – Manuscript.

Bachelor's qualification thesis in specialty 174 "Automation, Computer-Integrated Technologies and Robotics". – V. I. Vernadsky Tavria National University, Kyiv, 2026.

The bachelor's thesis examines the process of designing an automated home security management system based on the Smart Home concept and Internet of Things technologies. The paper analyzes the subject area of automated security systems for residential buildings, considers modern alarm systems, access control systems, video surveillance, fire alarm systems, and multi-level approaches to ensuring facility security.

The study substantiates the selection of technical components of the system, including the ESP32 microcontroller, motion sensors, door and window opening sensors, smoke, gas, flooding, temperature and humidity sensors, as well as actuators for local response to dangerous events. A structural diagram of the automated security system was developed, its main levels, data flows, and control signals were defined. UML modeling tools were used to formalize the interaction between the user and the system.

The practical part of the thesis focuses on the development of the algorithmic and software components of the system. The functional requirements for the software part were defined, and the functional structure, algorithms for processing security and emergency events, data structure, and software implementation for the ESP32 microcontroller were developed. The software provides sensor data acquisition, analysis of the obtained data, determination of the system operating mode, activation of actuators, generation of user notifications, and event logging.

Special attention is paid to occupational safety, electrical safety, fire safety, and emergency safety during the installation and operation of the automated home security system. The proposed system makes it possible to increase the level of protection of a residential facility, ensure timely detection of dangerous situations, and reduce the response time to security and emergency events.

Keywords: automated security system, smart home, Internet of Things, ESP32, sensors, alarm system, fire safety, GSM, Wi-Fi, UML modeling, software, microcontroller.

ЗМІСТ

ОСНОВНІ УМОВНІ ПОЗНАЧЕННЯ ТА СКОРОЧЕННЯ	9
ВСТУП	11
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ АВТОМАТИЗОВАНИХ СИСТЕМ БЕЗПЕКИ БУДИНКУ	15
1.1 Аналіз концепції та технологічних рішень системи «Розумний будинок»	15
1.2 Аналіз сучасних охоронних систем для житлових будинків	17
1.3 Дослідження багаторівневих систем забезпечення безпеки будинку	22
1.4 Огляд та порівняльний аналіз існуючих систем безпеки	25
1.5 Аналіз датчиків та виконавчих пристроїв автоматизованої системи безпеки будинку	29
Висновки до розділу 1	32
2 ПРОЄКТУВАННЯ СТРУКТУРИ ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ БЕЗПЕКОЮ БУДИНКУ	34
2.1 Обґрунтування вибору технічних засобів системи безпеки	34
2.2 Проєктування автоматизованої системи безпеки засобами UML- моделювання	49
2.3 Розроблення структурної схеми автоматизованої системи безпеки	59
Висновки до розділу 2	67
3 РОЗРОБЛЕННЯ АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ БЕЗПЕКИ	69
3.1 Визначення функціональних вимог до алгоритмічного та програмного забезпечення системи	69
3.2 Розроблення функціональної структури автоматизованої системи безпеки	76
3.3 Розроблення алгоритмів функціонування автоматизованої системи безпеки	83
3.4 Організація та структура даних в автоматизованій системі безпеки	93
3.5 Розроблення програмного забезпечення автоматизованої системи управління безпекою будинку	103
Висновки до розділу 3	117

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	119
4.1 Аналіз небезпечних і шкідливих виробничих факторів під час розроблення та експлуатації автоматизованої системи безпеки будинку	120
4.2 Заходи з охорони праці під час монтажу та експлуатації системи безпеки	121
4.3 Пожежна безпека об'єкта та заходи щодо запобігання надзвичайним ситуаціям	122
Висновки до розділу 4	124
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	125
ДОДАТКИ	129
Додаток А. Програмний код автоматизованої системи управління безпекою будинку	129

ОСНОВНІ УМОВНІ ПОЗНАЧЕННЯ ТА СКОРОЧЕННЯ

АСУББ — автоматизована система управління безпекою будинку.

IoT — Internet of Things, Інтернет речей.

ESP32 — мікроконтролерна платформа з вбудованими засобами бездротового зв'язку Wi-Fi та Bluetooth.

Wi-Fi — технологія бездротового передавання даних у локальних мережах.

GSM — Global System for Mobile Communications, стандарт мобільного зв'язку.

SMS — Short Message Service, служба коротких текстових повідомлень.

UML — Unified Modeling Language, уніфікована мова моделювання.

GPIO — General Purpose Input/Output, універсальний цифровий вхід/вихід мікроконтролера.

ADC / АЦП — Analog-to-Digital Converter, аналогово-цифровий перетворювач.

PWM / ШИМ — Pulse Width Modulation, широтно-імпульсна модуляція.

PIR — Passive Infrared Sensor, пасивний інфрачервоний датчик руху.

DHT22 — цифровий датчик температури та вологості.

MQ-2 — напівпровідниковий датчик газу та диму.

MQ-5 — напівпровідниковий датчик газу.

LM393 — компаратор, який використовується у модулях датчиків, зокрема датчиках затоплення.

MC-38 — магнітоконтатний датчик відкриття дверей або вікна.

SIM800L — GSM-модуль для передавання SMS-повідомлень і здійснення дзвінків.

JSON — JavaScript Object Notation, текстовий формат обміну структурованими даними.

MQTT — Message Queuing Telemetry Transport, протокол передавання повідомлень для IoT-систем.

API — Application Programming Interface, програмний інтерфейс взаємодії.

HMI — Human-Machine Interface, людино-машинний інтерфейс.

БД — база даних.

ПЗ — програмне забезпечення.

НС — надзвичайна ситуація.

ДБН — державні будівельні норми.

ДСТУ — державний стандарт України.

НПАОП — нормативно-правовий акт з охорони праці.

ОП — охорона праці.

HIGH — високий логічний рівень сигналу.

LOW — низький логічний рівень сигналу.

ВСТУП

Стрімкий розвиток інформаційних технологій, мікропроцесорної техніки, засобів автоматизації та бездротових комунікацій суттєво вплинув на підходи до організації безпеки житлових об'єктів. Сучасні будинки дедалі частіше оснащуються інтелектуальними системами керування, які забезпечують автоматичний моніторинг стану приміщень, контроль доступу, відеоспостереження, пожежну безпеку та оперативне інформування власників про виникнення небезпечних ситуацій. У таких умовах особливого значення набуває концепція «Розумний будинок», яка передбачає інтеграцію різноманітних технічних засобів у єдину автоматизовану систему управління [1].

Актуальність теми зумовлена постійним зростанням вимог до безпеки житлових будинків, необхідністю підвищення надійності виявлення загроз та забезпечення оперативного реагування на надзвичайні ситуації. Традиційні охоронні системи здебільшого орієнтовані на виконання окремих функцій і не забезпечують комплексного контролю за станом об'єкта. Водночас сучасні автоматизовані системи безпеки дозволяють об'єднати в єдиному інформаційному середовищі охоронну сигналізацію, відеоспостереження, контроль доступу, пожежну безпеку та засоби моніторингу інженерних мереж будинку. Такий підхід забезпечує підвищення ефективності функціонування системи, скорочення часу реагування на небезпечні події та створення комфортних умов експлуатації житла [12].

Особливої актуальності проблема набуває в умовах широкого впровадження технологій Інтернету речей (IoT), хмарних сервісів та мобільних платформ. Використання сучасних засобів передавання даних і програмних технологій дозволяє здійснювати дистанційний контроль за станом об'єкта та керування його підсистемами незалежно від місця перебування користувача. В умовах воєнного стану та зростання ризиків виникнення аварійних ситуацій потреба у впровадженні ефективних автоматизованих систем безпеки для житлових будинків стає ще більш актуальною [17].

Питанням розроблення автоматизованих систем керування, інтелектуальних інформаційних технологій, кібернетики та систем моніторингу присвячені праці багатьох вітчизняних і зарубіжних учених. Значний внесок у розвиток теорії автоматизації та комп'ютерно-інтегрованих технологій зробили В. М. Глушков, М. З. Згуровський, А. О. Морозов, Н. Вінер та інші дослідники. Вагоме значення для розвитку концепції Інтернету речей і сучасних інтелектуальних систем мають роботи Кевіна Ештона та інших науковців, присвячені питанням інтеграції фізичних пристроїв у єдині інформаційні мережі. Разом з тим постійний розвиток технічних засобів і технологій потребує вдосконалення існуючих рішень та створення нових автоматизованих систем безпеки, здатних забезпечувати комплексний захист житлових об'єктів [23].

Метою кваліфікаційної роботи є підвищення ефективності забезпечення безпеки житлового будинку шляхом розроблення автоматизованої системи управління безпекою на базі концепції «Розумний будинок», яка забезпечує комплексний моніторинг стану об'єкта, своєчасне виявлення небезпечних ситуацій та автоматизоване формування керуючих впливів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз концепції та технологічних рішень системи «Розумний будинок»;
- дослідити сучасні охоронні системи для житлових будинків та визначити їх функціональні особливості;
- виконати аналіз багаторівневих систем забезпечення безпеки будинку;
- здійснити огляд та порівняльний аналіз існуючих систем безпеки;
- обґрунтувати вибір технічних засобів автоматизованої системи управління безпекою будинку;
- виконати UML-моделювання та розробити структуру автоматизованої системи безпеки;
- визначити функціональні вимоги до алгоритмічного та програмного забезпечення системи;

- розробити функціональну структуру та алгоритми роботи автоматизованої системи;
- сформулювати структуру даних автоматизованої системи управління безпекою;
- розробити програмне забезпечення автоматизованої системи управління безпекою будинку.

Об'єктом дослідження є процес автоматизованого моніторингу та управління безпекою житлового будинку.

Предметом дослідження є методи, алгоритми, технічні та програмні засоби побудови автоматизованих систем управління безпекою житлових об'єктів на основі технологій Інтернету речей та концепції «Розумний будинок».

Для досягнення поставленої мети у роботі використано такі методи дослідження: системний аналіз – для дослідження предметної області та аналізу існуючих систем безпеки; методи порівняльного аналізу – для оцінювання сучасних технічних рішень; методи структурного та функціонального моделювання – для проектування архітектури автоматизованої системи; UML-моделювання – для формалізації структури та взаємодії компонентів системи; методи алгоритмізації та програмування – для розроблення програмного забезпечення; методи інформаційного моделювання – для формування структури даних автоматизованої системи управління [26].

Практична цінність одержаних результатів полягає у розробленні структури та програмного забезпечення автоматизованої системи управління безпекою будинку, яка може бути використана для створення сучасних інтегрованих систем моніторингу та керування житловими об'єктами. Запропоновані технічні та програмні рішення забезпечують можливість інтеграції охоронної сигналізації, систем відеоспостереження, контролю доступу та моніторингу аварійних ситуацій у межах єдиного програмно-апаратного комплексу.

Результати роботи можуть бути використані під час проектування та модернізації систем типу «Розумний будинок», а також у практичній діяльності

підприємств, що займаються впровадженням автоматизованих систем безпеки та комп'ютерно-інтегрованих технологій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ АВТОМАТИЗОВАНИХ СИСТЕМ БЕЗПЕКИ БУДИНКУ

1.1 Аналіз концепції та технологічних рішень системи «Розумний будинок»

Стрімкий розвиток інформаційних технологій, засобів автоматизації та бездротового зв'язку сприяв появі сучасних інтелектуальних систем керування житловими об'єктами. Одним із найбільш перспективних напрямів у цій галузі є концепція «Розумний будинок» (Smart Home), яка передбачає інтеграцію різноманітних пристроїв, датчиків та виконавчих механізмів у єдину автоматизовану систему керування. Основною метою впровадження таких систем є підвищення рівня комфорту, енергоефективності та безпеки житлових приміщень.

Концепція «Розумний будинок» базується на принципі централізованого або розподіленого керування інженерними системами будинку за допомогою сучасних засобів автоматизації. До складу такої системи можуть входити датчики руху, температури, вологості, освітленості, диму, витоку газу та затоплення, а також різноманітні виконавчі пристрої, що забезпечують автоматичне виконання заданих функцій. Отримуючи інформацію від датчиків, система аналізує поточний стан об'єкта та формує відповідні керуючі впливи [2].

Особливістю сучасних систем «Розумний будинок» є їхня здатність функціонувати в режимі реального часу. Завдяки цьому користувач має можливість контролювати стан житла та керувати обладнанням як локально, так і дистанційно через мережу Інтернет. Такий підхід дозволяє оперативно реагувати на виникнення нештатних ситуацій, зокрема спроб несанкціонованого проникнення, виникнення пожежі, витоку газу або затоплення приміщення.

Архітектура системи «Розумний будинок» зазвичай включає три основні рівні: рівень збору даних, рівень обробки інформації та рівень взаємодії з користувачем. На нижньому рівні розташовуються різноманітні датчики та вимірювальні пристрої, що здійснюють безперервний моніторинг параметрів навколишнього середовища. Середній рівень представлений контролером або

мікропроцесорною системою, яка виконує обробку отриманих даних та формує команди керування. Верхній рівень забезпечує взаємодію користувача із системою за допомогою мобільних застосунків, вебінтерфейсів або спеціалізованих панелей керування [12].

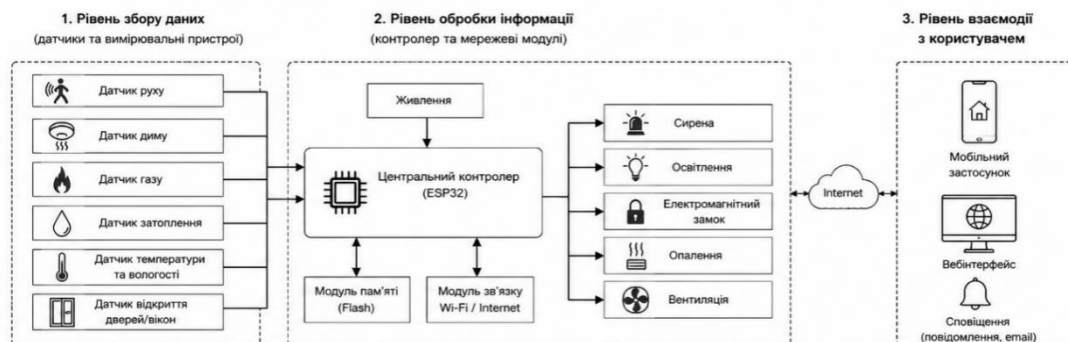


Рисунок 1.1 – Узагальнена архітектура системи «Розумний будинок»

Важливе значення для побудови систем «Розумний будинок» мають технології передавання даних. На сучасному етапі розвитку найбільшого поширення набули бездротові технології зв'язку, які забезпечують простоту монтажу та гнучкість розширення системи. Для обміну даними можуть використовуватись мережі Wi-Fi, Bluetooth, ZigBee, Z-Wave та інші протоколи бездротової комунікації. Вибір конкретної технології залежить від вимог до дальності зв'язку, енергоспоживання та швидкості передавання даних.

Одним із ключових напрямів розвитку інтелектуальних систем є застосування технологій Інтернету речей (IoT). Концепція IoT передбачає об'єднання фізичних пристроїв у єдину інформаційну мережу, що дозволяє здійснювати централізований моніторинг та керування різними компонентами системи. Завдяки використанню хмарних сервісів забезпечується доступ до інформації про стан об'єкта незалежно від місця перебування користувача.

Суттєву роль у функціонуванні систем «Розумний будинок» відіграють мікроконтролерні платформи. Серед найбільш поширених рішень можна виділити Arduino та ESP32. Використання таких платформ дозволяє реалізувати широкий спектр функцій автоматизації при відносно невисокій вартості обладнання. Особливої популярності набувають контролери ESP32, які поєднують достатню

обчислювальну потужність із вбудованими засобами бездротового зв'язку Wi-Fi та Bluetooth.

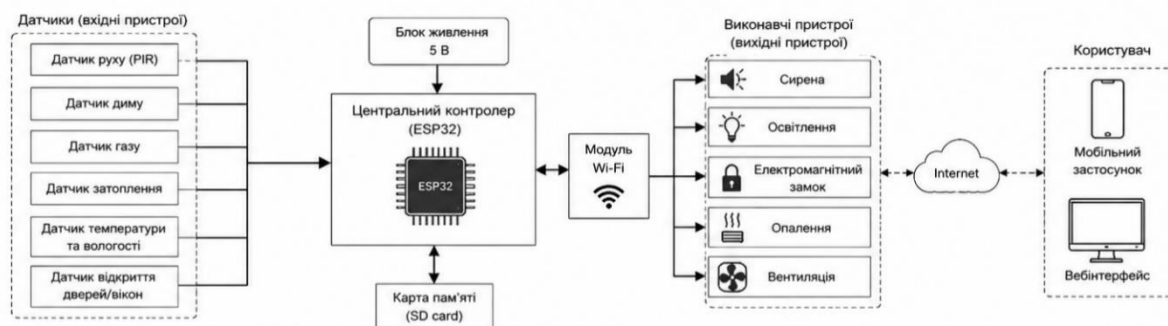


Рисунок 1.2 – Структурна схема взаємодії компонентів системи «Розумний будинок»

Одним із найбільш актуальних напрямів використання технології «Розумний будинок» є забезпечення комплексної безпеки житлових об'єктів. Інтеграція охоронних, пожежних та аварійних підсистем у межах єдиного програмно-апаратного комплексу дозволяє підвищити ефективність виявлення небезпечних ситуацій та скоротити час реагування на них. Автоматичне формування повідомлень про тривогу та передавання їх власнику житла значно підвищує рівень захищеності об'єкта [15].

Таким чином, концепція «Розумний будинок» є сучасним підходом до автоматизації житлових приміщень, який поєднує технології збору даних, їх обробки та дистанційного керування обладнанням. Застосування сучасних мікроконтролерів, бездротових мереж та технологій Інтернету речей створює основу для побудови ефективних автоматизованих систем управління безпекою будинку, що обумовлює доцільність їх використання в рамках даної роботи [25].

1.2 Аналіз сучасних охоронних систем для житлових будинків

Забезпечення безпеки житлових будинків є одним із ключових завдань сучасних автоматизованих систем. Розвиток електроніки, засобів зв'язку та мікропроцесорної техніки сприяв появі великої кількості технічних рішень, призначених для захисту майна та мешканців від різноманітних загроз. Сучасні охоронні системи здатні не лише виявляти небезпечні події, а й автоматично інформувати власника або відповідні служби про їх виникнення.

Залежно від функціонального призначення та принципу роботи системи безпеки для житлових будинків можна умовно поділити на системи охоронної сигналізації, системи контролю доступу, комплекси аудіо- та відеоспостереження, автономні охоронні системи та інтегровані системи типу «Розумний будинок». Кожна з наведених категорій вирішує власне коло завдань, проте на практиці вони часто об'єднуються в єдиний програмно-апаратний комплекс [3].

1.2.1 Система охоронної і тривожної сигналізації

Охоронна сигналізація є одним із найпоширеніших засобів захисту житлових об'єктів. Її основним призначенням є своєчасне виявлення несанкціонованого проникнення на територію будинку або виникнення інших небезпечних ситуацій. Типова система складається з набору датчиків, центрального контролера та пристроїв оповіщення.

Для контролю стану об'єкта використовуються датчики руху, відкриття дверей і вікон, розбиття скла, диму, витоку газу та затоплення. Інформація від них надходить до центрального блоку керування, де здійснюється аналіз отриманих даних відповідно до закладеного алгоритму роботи. У разі виявлення загрози система формує сигнал тривоги та передає його на виконавчі пристрої або канали зв'язку.

Датчики можуть підключатися до контролера як за допомогою кабельних ліній, так і через бездротові канали зв'язку. Вибір способу підключення залежить від особливостей об'єкта, складності монтажу та вимог до надійності системи [18].

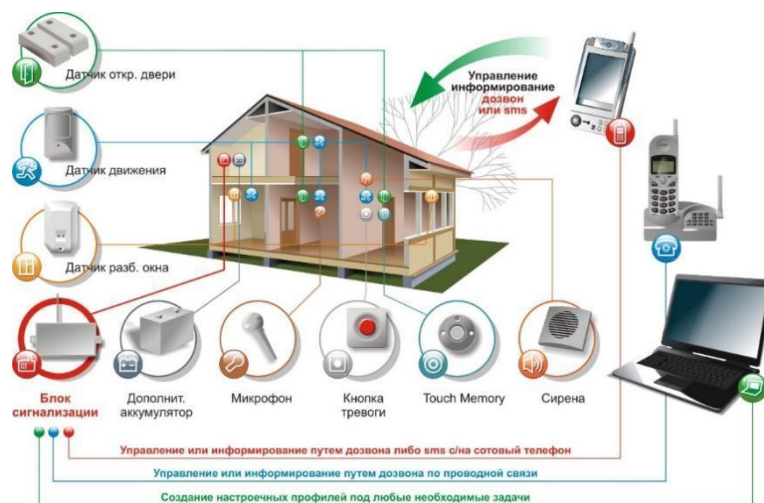


Рисунок 1.2 – Схема системи охоронної і тривожної сигналізації

Важливою особливістю сучасних систем є можливість дистанційного сповіщення користувача. Для цього використовуються GSM-модулі або мережеві інтерфейси, які забезпечують передачу повідомлень на мобільний телефон власника чи пульт централізованого спостереження. Завдяки цьому значно скорочується час реагування на небезпечні події та підвищується загальний рівень безпеки об'єкта.

1.2.2 Система контролю доступу

Системи контролю доступу призначені для обмеження та регулювання доступу до будинку або його окремих приміщень. Використання таких систем дозволяє не лише запобігати проникненню сторонніх осіб, але й здійснювати облік відвідувань та контролювати переміщення користувачів у межах об'єкта.

Основу системи становлять ідентифікаційні пристрої, виконавчі механізми та контролери керування. Для підтвердження права доступу можуть використовуватися персональні коди, електронні ключі, RFID-картки або біометричні засоби ідентифікації. Після успішного проходження процедури перевірки система надсилає команду на відкриття електромеханічного замка, турнікета або іншого виконавчого пристрою [16].

Особливістю таких систем є можливість гнучкого налаштування прав доступу. Наприклад, власник будинку може обмежити доступ до окремих приміщень або встановити різні режими роботи для членів сім'ї та обслуговуючого персоналу.

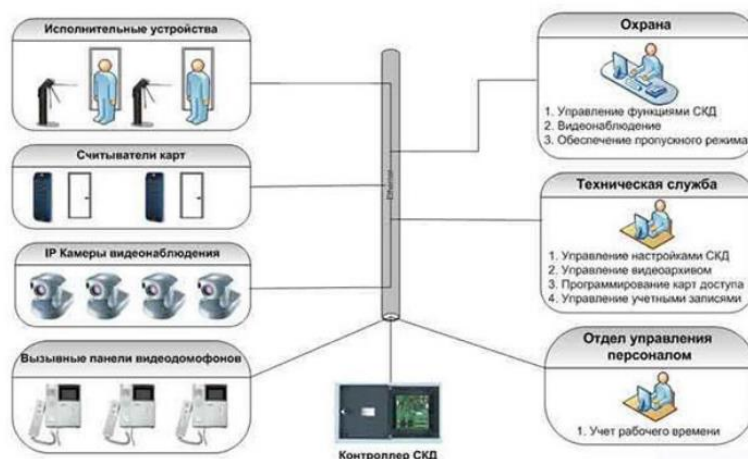


Рисунок 1.3 – Схема системи контролю доступу

Застосування систем контролю доступу дозволяє підвищити рівень захисту об'єкта та мінімізувати ризик несанкціонованого проникнення.

1.2.3 Система аудіо- та відеоспостереження

Важливе місце серед сучасних засобів безпеки займають системи відеоспостереження. Їх використання забезпечує безперервний візуальний контроль за територією будинку та дозволяє оперативно виявляти потенційно небезпечні ситуації.

До складу таких систем входять відеокамери, пристрої запису та зберігання інформації, мережеве обладнання та програмні засоби моніторингу. Камери можуть встановлюватися як усередині приміщень, так і по периметру території. Для зовнішнього монтажу застосовуються моделі із захистом від пилу, вологи та перепадів температур.

Сучасні відеокамери часто оснащуються інфрачервоним підсвічуванням, що забезпечує спостереження в умовах недостатнього освітлення. Додатково можуть використовуватися поворотні механізми, які збільшують сектор огляду та дозволяють контролювати значну площу за допомогою меншої кількості камер.



Рисунок 1.4 – Схема системи відеоспостереження

Перевагою сучасних систем є можливість віддаленого перегляду відеопотоку через мережу Інтернет, що дозволяє власнику контролювати стан об'єкта незалежно від свого місцезнаходження.

1.2.4 Автономні охоронні системи

Автономні системи безпеки призначені для функціонування без постійного втручання користувача. Вони здійснюють автоматичний моніторинг стану об'єкта та самостійно реагують на виникнення небезпечних ситуацій відповідно до заданих алгоритмів.

На відміну від традиційної охоронної сигналізації, автономні комплекси можуть контролювати не лише факти проникнення, але й роботу інженерних мереж будинку. Такі системи здатні відстежувати параметри електропостачання, водопостачання, газопостачання та інші критично важливі показники [23].



Рисунок 1.5 – Приклад автономної охоронної системи

Для забезпечення безперервної роботи більшість автономних систем оснащуються резервними джерелами живлення. Використання акумуляторних батарей дозволяє зберігати працездатність комплексу навіть у разі відключення електроенергії, що є важливою перевагою з точки зору надійності.

1.2.5 Системи типу «Розумний будинок»

Найбільш функціональним напрямом розвитку засобів безпеки є інтегровані системи «Розумний будинок». На відміну від окремих охоронних рішень, вони поєднують у собі функції моніторингу, керування та автоматизації інженерних систем будинку.

До складу таких комплексів можуть входити підсистеми охоронної сигналізації, відеоспостереження, контролю доступу, пожежної безпеки та

керування енергоспоживанням. Усі компоненти працюють як єдина система, що забезпечує комплексний підхід до захисту житла.

Наприклад, при виявленні диму система може автоматично вимкнути електроживлення певних пристроїв, перекрити подачу газу, активувати вентиляцію та передати повідомлення власнику. Подібна інтеграція дозволяє не лише підвищити рівень безпеки, а й скоротити час реагування на аварійні ситуації.



Рисунок 1.6 – Структурна схема системи «Розумний будинок»

Таким чином, сучасні охоронні системи відрізняються рівнем функціональності, способом організації та складністю реалізації. Найбільш перспективним напрямом є інтегровані комплекси типу «Розумний будинок», які забезпечують поєднання охоронних функцій, моніторингу та автоматизованого керування інженерними системами житлового будинку.

1.3 Дослідження багаторівневих систем забезпечення безпеки будинку

Підвищення вимог до захисту житлових об'єктів зумовило перехід від використання окремих охоронних пристроїв до комплексних багаторівневих систем безпеки. Практика показує, що застосування лише одного засобу захисту не забезпечує достатнього рівня надійності, оскільки кожна технічна система має певні обмеження та не здатна ефективно реагувати на всі можливі загрози. Саме тому сучасні системи безпеки будуються за принципом багаторівневого захисту,

коли декілька незалежних підсистем спільно забезпечують контроль стану об'єкта та реагування на небезпечні події.

Основна ідея багаторівневої системи полягає у створенні декількох рубежів контролю, кожен з яких виконує власні функції. У разі якщо певна загроза не була виявлена одним рівнем захисту, її може зафіксувати інший рівень системи. Такий підхід дозволяє суттєво підвищити загальну надійність охорони будинку та знизити ймовірність виникнення небезпечних ситуацій [6].

У більшості сучасних рішень перший рівень захисту формується на зовнішньому периметрі об'єкта. Його завданням є виявлення спроб проникнення на територію домоволодіння ще до безпосереднього доступу до будинку. Для цього можуть використовуватися датчики руху, камери відеоспостереження, освітлювальні пристрої з автоматичним увімкненням та інші засоби контролю території.

Наступний рівень безпеки забезпечує захист безпосередньо будівлі. На цьому етапі контролюється стан дверей, вікон та інших можливих точок проникнення. У разі спроби несанкціонованого відкриття або пошкодження конструкцій система формує сигнал тривоги та передає його користувачеві або охоронній службі.

Особливе значення має внутрішній рівень контролю, який відповідає за моніторинг ситуації всередині приміщень. Для цього використовуються датчики руху, системи відеоспостереження та інші пристрої, здатні виявляти присутність сторонніх осіб після проникнення до будинку. Такий підхід дозволяє забезпечити додатковий рівень захисту навіть у випадках, коли зовнішні рубежі були подолані.

Важливою складовою сучасних багаторівневих систем є підсистема техногенної безпеки. Її призначення полягає у своєчасному виявленні аварійних ситуацій, пов'язаних із виникненням пожежі, витоком газу, затопленням приміщень або порушенням параметрів електроживлення. На відміну від класичних охоронних систем, які спрямовані переважно на протидію зовнішнім загрозам, техногенні підсистеми дозволяють захищати будинок від внутрішніх факторів ризику.

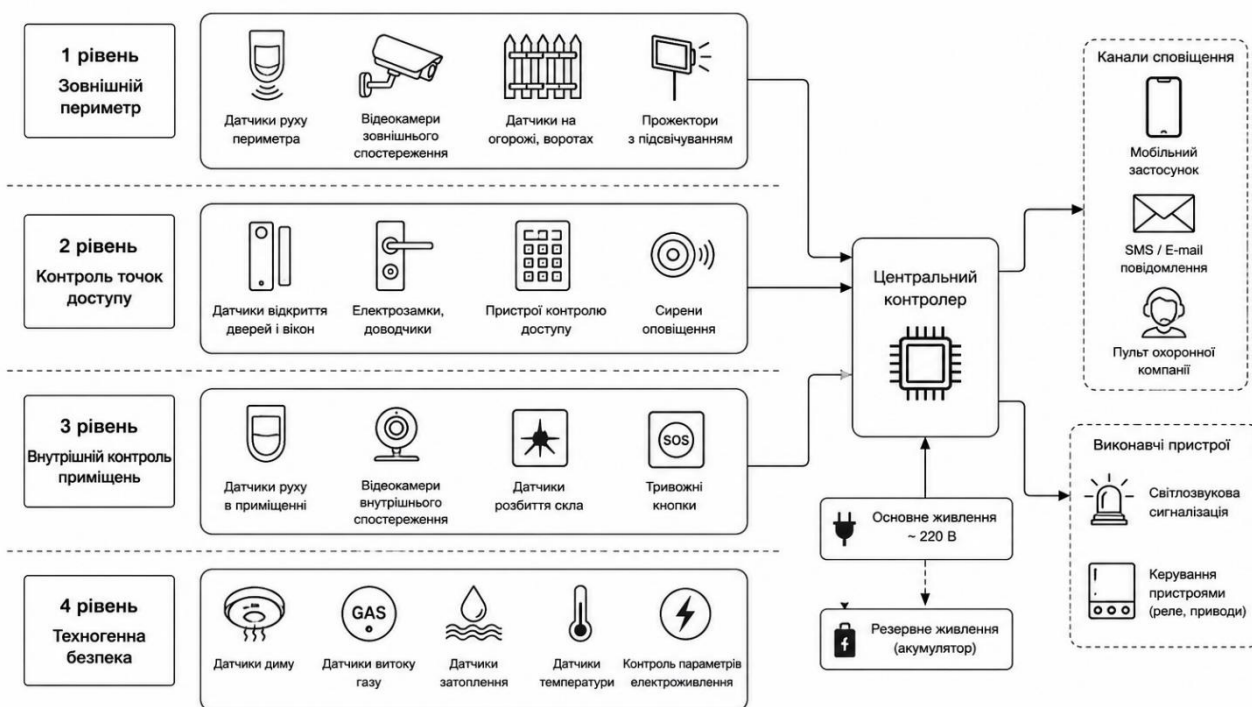


Рисунок 1.7 – Структура багаторівневої системи забезпечення безпеки будинку

Розвиток мікропроцесорної техніки та мережевих технологій сприяв появі інтегрованих систем, у яких усі рівні безпеки функціонують як єдиний комплекс. Центральний контролер здійснює збір інформації від датчиків, аналізує отримані дані та формує необхідні керуючі впливи. У разі виникнення небезпечної події система може одночасно активувати звукову сигналізацію, надіслати повідомлення власнику, виконати запис відео та задіяти інші механізми реагування [8].

Однією з переваг багаторівневого підходу є можливість масштабування системи. Залежно від особливостей об'єкта та вимог користувача до складу комплексу можуть додаватися нові датчики, виконавчі пристрої або програмні модулі без суттєвої зміни загальної структури. Це дозволяє адаптувати систему до конкретних умов експлуатації та забезпечувати необхідний рівень захисту при різних фінансових можливостях власника.

Слід також зазначити, що ефективність багаторівневої системи визначається не лише кількістю встановлених пристроїв, а й правильністю їх взаємодії. Найкращі результати досягаються тоді, коли всі елементи працюють узгоджено та обмінюються інформацією в режимі реального часу. Саме тому сучасні системи безпеки дедалі частіше реалізуються на базі концепції «Розумний будинок», де

охоронні, пожежні та сервісні підсистеми інтегруються в єдине інформаційне середовище.

Таким чином, багаторівневі системи забезпечення безпеки є найбільш ефективним підходом до захисту житлових будинків. Поєднання декількох рівнів контролю дозволяє підвищити надійність виявлення загроз, скоротити час реагування на небезпечні події та забезпечити комплексний захист майна і мешканців будинку.

1.4 Огляд та порівняльний аналіз існуючих систем безпеки

Системи безпеки протягом останніх десятиліть перетворилися з окремих технічних засобів охорони на складні інтегровані комплекси, здатні виконувати широкий спектр завдань. Сучасні рішення забезпечують не лише виявлення несанкціонованого проникнення, але й контроль технічного стану будинку, відеоспостереження, керування доступом, моніторинг пожежної безпеки та дистанційну взаємодію з користувачем. У зв'язку зі зростанням вимог до безпеки житла попит на такі системи постійно збільшується як у світі, так і в Україні [3].

Розвиток ринку систем безпеки безпосередньо пов'язаний із поширенням технологій Інтернету речей, бездротових мереж передачі даних та мобільних платформ. Якщо раніше більшість охоронних комплексів працювали автономно та виконували обмежений набір функцій, то сучасні системи здатні інтегруватися в єдине інформаційне середовище та забезпечувати постійний зв'язок із власником через мережу Інтернет. Завдяки цьому користувач отримує можливість контролювати стан об'єкта практично з будь-якої точки світу.

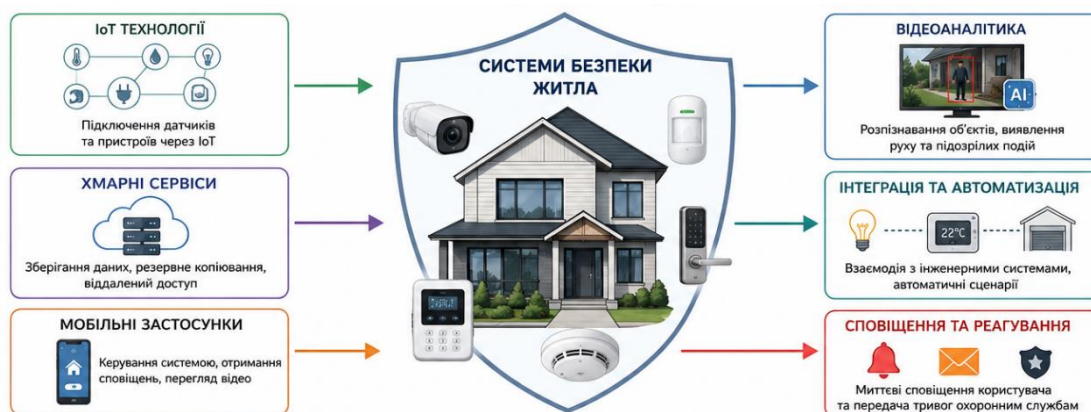


Рисунок 1.8 – Основні напрями розвитку сучасних систем безпеки

Сучасний ринок охоронних систем представлений великою кількістю виробників та програмно-апаратних рішень. Найбільшого поширення набули системи охоронної сигналізації, відеоспостереження, контролю доступу та інтегровані комплекси «Розумний будинок». При цьому спостерігається тенденція до поступового об'єднання різних підсистем у межах єдиної платформи управління.

Особливу роль у розвитку систем безпеки відіграють хмарні технології. Використання віддалених серверів для зберігання інформації дозволяє забезпечити доступ до даних незалежно від місця перебування користувача. Власник будинку може отримувати повідомлення про тривожні події, переглядати відео з камер спостереження та змінювати налаштування системи за допомогою мобільного застосунку. Крім зручності використання, хмарні сервіси забезпечують додатковий рівень надійності зберігання інформації, оскільки дані дублюються на віддалених серверах [13].

Ще одним важливим напрямом розвитку є впровадження технологій штучного інтелекту та машинного навчання. Використання алгоритмів аналізу даних дозволяє системам безпеки розпізнавати типові сценарії поведінки та своєчасно виявляти аномальні події. Особливо ефективно такі технології використовуються у системах відеоспостереження, де програмне забезпечення може автоматично визначати рух людини, проникнення на територію або інші потенційно небезпечні ситуації. Це дозволяє зменшити кількість помилкових спрацювань та підвищити ефективність моніторингу.

Важливою особливістю сучасного ринку є активний розвиток бездротових технологій. Використання Wi-Fi, Bluetooth, ZigBee та інших протоколів передачі даних значно спрощує процес встановлення обладнання та розширення функціональних можливостей системи. Відсутність необхідності прокладання великої кількості кабелів дозволяє скоротити витрати на монтаж і забезпечує більшу гнучкість при модернізації системи.

В Україні розвиток ринку систем безпеки має власні особливості. В умовах воєнного стану значно зросла потреба у засобах моніторингу та оперативного реагування на надзвичайні ситуації. Власники житлових будинків дедалі частіше

використовують системи дистанційного контролю для спостереження за майном під час тривалої відсутності. Крім традиційного захисту від проникнення сторонніх осіб, важливого значення набувають функції контролю електроживлення, виявлення задимлення, витоків газу та аварійних ситуацій, які можуть виникати внаслідок пошкодження інфраструктури.

Серед виробників охоронних систем, представлених на українському ринку, особливу увагу привертають рішення української компанії Ajax Systems. Продукція компанії орієнтована на створення інтегрованих комплексів безпеки та включає широкий спектр датчиків, пристроїв оповіщення, засобів автоматизації та елементів відеоспостереження. Однією з ключових переваг таких рішень є можливість централізованого керування всіма компонентами через мобільний застосунок та підтримка хмарної інфраструктури [28].

Значне місце на ринку займають також системи відеоспостереження. Сучасні ІР-камери забезпечують високу якість зображення, підтримують дистанційний доступ та можуть інтегруватися з іншими підсистемами безпеки. Завдяки використанню мережевих технологій користувач отримує можливість переглядати відеопотік у режимі реального часу та зберігати записи на локальних або віддалених носіях інформації.

Практика використання систем відеоспостереження свідчить про їхню високу ефективність як засобу стримування потенційних правопорушників. Наявність камер не лише дозволяє фіксувати події, але й виконує превентивну функцію, знижуючи ймовірність протиправних дій. Крім того, відеоматеріали можуть використовуватися для встановлення обставин події та ідентифікації осіб, причетних до правопорушення.

Останніми роками все більшої популярності набувають системи безпеки, орієнтовані на самостійне встановлення користувачем. Такі рішення отримали широке поширення завдяки простоті монтажу та можливості налаштування без залучення спеціалізованих організацій. Застосування бездротових технологій дозволяє власнику будинку самостійно розгорнути систему та адаптувати її конфігурацію відповідно до особливостей об'єкта.

Однією з ключових тенденцій сучасного ринку є використання мобільних пристроїв як основного інструменту взаємодії із системою безпеки. Смартфон фактично перетворюється на універсальний центр керування, за допомогою якого можна отримувати повідомлення про тривожні події, переглядати відео з камер, керувати режимами охорони та контролювати роботу виконавчих пристроїв. Такий підхід значно підвищує зручність експлуатації системи та забезпечує оперативність реагування на будь-які зміни її стану.

Аналіз функціональних можливостей сучасних систем безпеки показує, що найбільш ефективними є комплексні рішення, які поєднують декілька підсистем у межах єдиної архітектури. У таблиці 1.1 наведено порівняння основних типів систем безпеки.

Таблиця 1.1 – Порівняння сучасних систем безпеки житлового будинку

Тип системи	Основне призначення	Переваги	Обмеження
Охоронна сигналізація	Виявлення проникнення	Простота та надійність	Обмежений функціонал
Контроль доступу	Обмеження доступу до об'єкта	Високий рівень контролю	Потребує додаткових засобів захисту
Відеоспостереження	Візуальний контроль території	Фіксація подій та дистанційний моніторинг	Значний обсяг даних для зберігання
Автономні системи	Автоматичний моніторинг об'єкта	Незалежність та швидке реагування	Обмежені можливості масштабування
Системи «Розумний будинок»	Комплексне управління безпекою та інженерними системами	Інтеграція всіх функцій в одному середовищі	Вища вартість реалізації

Отримані результати свідчать про те, що найбільш перспективним напрямом розвитку є інтегровані системи типу «Розумний будинок». Вони дозволяють об'єднати охоронну сигналізацію, контроль доступу, відеоспостереження та моніторинг технічного стану будинку в межах єдиного програмно-апаратного комплексу. Саме такий підхід забезпечує максимальний рівень захисту та відповідає сучасним вимогам до автоматизації житлових об'єктів [26].



Рисунок 1.9 – Порівняння функціональних можливостей сучасних систем безпеки

Таким чином, проведений аналіз показав, що сучасні системи безпеки розвиваються у напрямі інтеграції, підвищення автономності та використання інтелектуальних методів обробки інформації. Для забезпечення ефективного захисту житлового будинку доцільно використовувати комплексний підхід, який передбачає поєднання декількох підсистем безпеки в межах єдиної автоматизованої системи управління. Саме такий підхід буде покладений в основу розроблюваної системи безпеки будинку у межах даної кваліфікаційної роботи.

1.5 Аналіз датчиків та виконавчих пристроїв автоматизованої системи безпеки будинку

Ефективність функціонування автоматизованої системи безпеки будинку значною мірою визначається правильним вибором датчиків та виконавчих пристроїв. Саме ці компоненти забезпечують збір інформації про стан об'єкта, виявлення небезпечних ситуацій та реалізацію необхідних керуючих впливів у разі виникнення загроз.

У сучасних системах безпеки датчики виконують роль первинних джерел інформації. Вони здійснюють безперервний моніторинг контрольованих параметрів навколишнього середовища та передають отримані дані до центрального контролера для подальшої обробки. На основі аналізу цих даних система приймає рішення щодо формування повідомлень користувачу або активації відповідних виконавчих пристроїв.

Залежно від функціонального призначення датчики систем безпеки можна умовно поділити на датчики охоронного контролю, пожежної безпеки, техногенного контролю та датчики контролю доступу [7].



Рисунок 1.10 Класифікація датчиків та виконавчих пристроїв автоматизованої системи безпеки будинку

До найбільш поширених засобів охоронного контролю належать датчики руху. Їх основним призначенням є виявлення присутності людини у контрольованій зоні. Найчастіше використовуються інфрачервоні датчики, які реагують на зміну теплового випромінювання в зоні спостереження. Такі пристрої широко застосовуються для контролю внутрішніх приміщень, коридорів та підходів до будинку.

Для контролю стану дверей і вікон використовуються магнітоконтатні датчики відкриття. Вони складаються з двох частин — магніту та геркона, які встановлюються на рухомій і нерухомій частинах конструкції. Під час відкривання дверей або вікна контакт розмикається, що фіксується системою як потенційна загроза несанкціонованого проникнення [17].

Важливу роль у забезпеченні пожежної безпеки відіграють датчики диму та температури. Датчики диму реагують на появу продуктів горіння у повітрі та дозволяють виявляти займання на ранніх стадіях розвитку пожежі. Температурні датчики контролюють рівень нагрівання навколишнього середовища та можуть використовуватися як додатковий засіб підтвердження пожежної небезпеки.

Для запобігання аварійним ситуаціям застосовуються датчики витоку газу та затоплення. Газові датчики здійснюють контроль концентрації природного або чадного газу в приміщенні та формують сигнал тривоги у разі перевищення допустимих значень. Датчики затоплення встановлюються поблизу джерел водопостачання та дозволяють своєчасно виявляти появу води на підлозі.

Окрему категорію становлять засоби відеоспостереження. Сучасні IP-камери забезпечують безперервний візуальний контроль території та можуть працювати спільно з іншими датчиками системи. У разі виявлення руху або іншої небезпечної події система може автоматично активувати запис відео та передати відповідне повідомлення користувачу.

Для реалізації керуючих впливів у системах безпеки використовуються виконавчі пристрої. До них належать звукові та світлові сирени, електромеханічні замки, електромагнітні клапани перекриття води або газу, релейні модулі керування та інші технічні засоби.

Сирени є одним із найпростіших та найефективніших способів привернення уваги до небезпечної події. Вони можуть активуватися під час спрацювання охоронної сигналізації, виявлення пожежі або інших аварійних ситуацій. Світлові індикатори додатково забезпечують візуальне оповіщення користувачів.

Електромеханічні та електромагнітні замки використовуються в системах контролю доступу. Вони дозволяють дистанційно керувати відкриванням і блокуванням дверей, а також інтегруються із системами ідентифікації користувачів [13].

Для автоматичного реагування на аварійні ситуації застосовуються електроприводи та керовані клапани. Наприклад, у разі виявлення витoku води система може автоматично перекрити подачу води до будинку, а при виявленні витoku газу — відключити подачу газу та активувати вентиляцію приміщення.



Рисунок 1.11 Структурна схема взаємодії датчиків, контролера та виконавчих пристроїв автоматизованої системи безпеки будинку

Таким чином, датчики та виконавчі пристрої є основними елементами автоматизованої системи безпеки будинку. Їх комплексне використання забезпечує своєчасне виявлення загроз, оперативне реагування на небезпечні ситуації та підвищує загальний рівень захищеності житлового об'єкта.

Таблиця 1.2 – Основні датчики та виконавчі пристрої автоматизованої системи безпеки будинку

Пристрій	Призначення	Контрольований параметр або дія
Датчик руху	Виявлення присутності сторонніх осіб	Рух у контрольованій зоні
Датчик відкриття дверей/вікон	Контроль проникнення	Стан дверей та вікон
Датчик диму	Виявлення пожежі	Концентрація продуктів горіння
Температурний датчик	Контроль температури	Температура середовища
Датчик витоку газу	Виявлення небезпечної концентрації газу	Концентрація газу
Датчик затоплення	Виявлення протікання води	Наявність води
ІР-камера	Візуальний моніторинг	Відеопотік
Сирена	Оповіщення про небезпеку	Звуковий сигнал
Світловий індикатор	Візуальне оповіщення	Світлова індикація
Електромеханічний замок	Контроль доступу	Блокування/відкривання дверей
Клапан перекриття води	Ліквідація аварії	Перекриття водопостачання
Релейний модуль	Керування виконавчими пристроями	Комутація навантаження

Висновки до розділу 1

У першому розділі кваліфікаційної роботи проведено аналіз предметної області автоматизованих систем безпеки будинку та досліджено сучасні підходи до забезпечення захисту житлових об'єктів.

Розглянуто концепцію системи «Розумний будинок», яка є одним із найбільш перспективних напрямів розвитку автоматизованих систем керування житлом. Встановлено, що інтеграція датчиків, виконавчих пристроїв, мікроконтролерних платформ та засобів бездротового зв'язку дозволяє реалізувати ефективні системи моніторингу й управління в режимі реального часу.

Виконано аналіз сучасних охоронних систем для житлових будинків, зокрема систем охоронної сигналізації, контролю доступу, відеоспостереження, автономних охоронних комплексів та інтегрованих систем типу «Розумний

будинку». Визначено їхні переваги, недоліки та особливості практичного застосування. Встановлено, що найбільш ефективними є комплексні рішення, які поєднують функції моніторингу, контролю та автоматизованого реагування на небезпечні події.

Проведене дослідження багаторівневих систем забезпечення безпеки показало, що використання декількох незалежних рубежів захисту дозволяє суттєво підвищити надійність виявлення загроз та зменшити ймовірність виникнення аварійних ситуацій. Визначено, що поєднання зовнішнього, внутрішнього та техногенного рівнів контролю забезпечує комплексний захист житлового об'єкта.

Виконаний огляд і порівняльний аналіз існуючих систем безпеки засвідчив тенденцію до інтеграції різних підсистем у межах єдиного програмно-апаратного комплексу. Встановлено, що сучасні системи безпеки активно використовують технології Інтернету речей, хмарні сервіси, мобільні застосунки та засоби інтелектуального аналізу даних для підвищення ефективності функціонування.

Окрему увагу приділено аналізу датчиків та виконавчих пристроїв автоматизованої системи безпеки. Визначено основні типи датчиків, необхідних для контролю стану об'єкта, а саме датчики руху, відкриття дверей і вікон, диму, температури, витоку газу та затоплення. Також розглянуто виконавчі пристрої, що забезпечують автоматичне реагування системи на небезпечні події, зокрема сирени, електромеханічні замки, релейні модулі та клапани перекриття води.

За результатами проведеного аналізу встановлено, що найбільш доцільним підходом до побудови автоматизованої системи безпеки будинку є використання інтегрованої архітектури на базі сучасного мікроконтролера з підтримкою бездротового зв'язку, яка забезпечує взаємодію датчиків, виконавчих пристроїв та засобів дистанційного керування в межах єдиного інформаційного середовища.

Отримані результати стали підґрунтям для подальшого проектування структури автоматизованої системи управління безпекою будинку, обґрунтування вибору технічних засобів та розроблення алгоритмічного і програмного забезпечення, що розглядаються у наступних розділах кваліфікаційної роботи.

2 ПРОЄКТУВАННЯ СТРУКТУРИ ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ БЕЗПЕКОЮ БУДИНКУ

2.1 Обґрунтування вибору технічних засобів системи безпеки

Для побудови автоматизованої системи управління безпекою будинку необхідно обрати такі технічні засоби, які забезпечуватимуть своєчасне виявлення небезпечних подій, оброблення сигналів від датчиків, формування керуючих дій та оперативне інформування користувача. До таких подій належать несанкціоноване проникнення до приміщення, відкриття дверей або вікон у режимі охорони, поява диму, витік газу, затоплення, а також відхилення параметрів мікроклімату від допустимих значень [25].

Автоматизована система безпеки будинку повинна працювати не лише як набір окремих датчиків, а як єдиний технічний комплекс, у якому всі елементи взаємодіють між собою. У такій системі датчики виконують функцію збору інформації про стан об'єкта, мікроконтролер здійснює аналіз отриманих сигналів і приймає рішення, а виконавчі пристрої реалізують реакцію системи: вмикають сирену, активують світлове оповіщення, перекривають воду або газ, надсилають повідомлення власнику.

Для побудови автоматизованої системи управління безпекою будинку доцільно використовувати архітектуру розподіленої периферії з централізованим обчислювальним ядром. Такий підхід передбачає, що датчики розміщуються у різних зонах будинку, а їхні сигнали надходять до центрального контролера. Контролер виконує оброблення інформації, визначає поточний стан системи та формує керуючі сигнали для виконавчих пристроїв. Така архітектура забезпечує оптимальний баланс між гнучкістю налаштування окремих вузлів і надійністю загальної логіки функціонування системи.

Загальну структуру технічних засобів автоматизованої системи безпеки будинку наведено на рисунку 2.1.

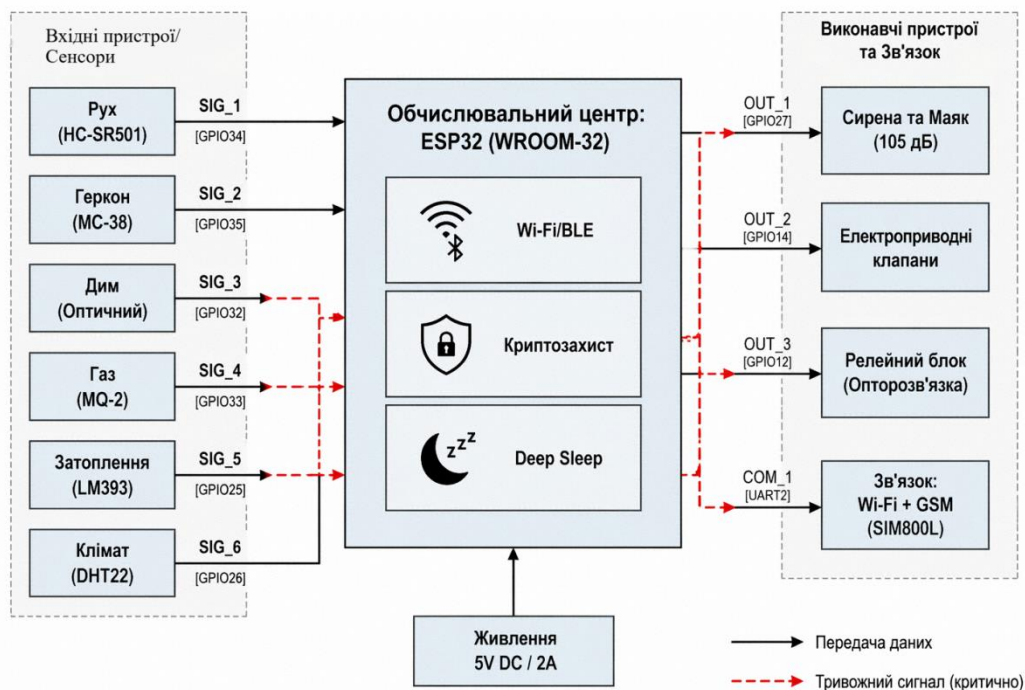


Рисунок 2.1 – Узагальнена структура технічних засобів автоматизованої системи управління безпекою будинку

Під час вибору апаратної бази автоматизованої системи безпеки необхідно враховувати не лише функціональність окремих пристроїв, а й можливість їхньої спільної роботи в межах єдиної системи. Основними критеріями вибору технічних засобів є енергоефективність, швидкодія, надійність, стійкість до завад, простота підключення, сумісність з бездротовими технологіями, економічна доцільність і можливість подальшого розширення системи.

Основні критерії вибору апаратних компонентів системи наведено в таблиці 2.1.

Таблиця 2.1 – Критерії вибору технічних засобів автоматизованої системи безпеки будинку

Критерій вибору	Зміст критерію	Значення для системи безпеки
Енергоефективність	Мінімальне споживання електроенергії датчиками та контролером	Забезпечує можливість автономної роботи системи від резервного джерела живлення
Швидкодія	Здатність швидко обробляти сигнали від датчиків	Дозволяє оперативно реагувати на тривожні події
Надійність	Стабільна робота компонентів у різних умовах експлуатації	Зменшує ризик відмови системи у критичній ситуації

Критерій вибору	Зміст критерію	Значення для системи безпеки
Стійкість до завад	Захист від електромагнітних перешкод і помилкових сигналів	Підвищує достовірність виявлення небезпечних подій
Простота інтеграції	Наявність стандартних інтерфейсів підключення	Спрощує монтаж, налаштування та обслуговування системи
Протокольна підтримка	Підтримка Wi-Fi, Bluetooth, UART, I2C, SPI, ADC, GPIO	Забезпечує можливість підключення різних датчиків і модулів зв'язку
Економічна доцільність	Доступна вартість компонентів	Дозволяє створити систему без значного збільшення бюджету
Масштабованість	Можливість додавання нових датчиків і виконавчих пристроїв	Дає змогу розширювати систему відповідно до потреб користувача

З урахуванням зазначених критеріїв технічне забезпечення автоматизованої системи управління безпекою будинку доцільно поділити на такі основні групи: обчислювальний центр, сенсорний рівень, виконавчі пристрої, комунікаційні засоби та система живлення.

2.1.1 Вибір обчислювального центру системи

Центральним елементом автоматизованої системи управління безпекою будинку є мікроконтролер, який виконує функції збору, оброблення та передавання даних. Саме мікроконтролер отримує сигнали від датчиків, аналізує їх відповідно до заданого алгоритму, визначає наявність загрози та формує команди для виконавчих пристроїв.

Як обчислювальне ядро системи доцільно використати мікроконтролер ESP32, зокрема модуль серії ESP32-WROOM-32. Його вибір обґрунтовується поєднанням достатньої обчислювальної продуктивності, наявністю вбудованих бездротових інтерфейсів, великою кількістю портів введення/виведення та невисокою вартістю [28].

ESP32 є 32-бітним двоядерним мікроконтролером, який працює на базі процесорного ядра Xtensa LX6 з тактовою частотою до 240 МГц. Така продуктивність є достатньою для реалізації алгоритмів моніторингу стану будинку, опрацювання сигналів від декількох датчиків, керування виконавчими пристроями та передавання повідомлень користувачу. У межах розроблюваної системи ESP32

виконує роль центрального вузла, який об'єднує всі апаратні компоненти в єдину автоматизовану систему.

Однією з основних переваг ESP32 є наявність вбудованих модулів Wi-Fi та Bluetooth. Завдяки цьому контролер може безпосередньо підключатися до домашньої бездротової мережі та передавати дані до мобільного застосунку, хмарного сервісу або сервісу повідомлень. Це усуває потребу у використанні додаткового Wi-Fi-модуля та спрощує апаратну структуру системи.

Важливою перевагою ESP32 є також підтримка значної кількості периферійних інтерфейсів. Контролер має цифрові входи та виходи GPIO, аналогово-цифровий перетворювач для зчитування аналогових сигналів, підтримує інтерфейси UART, I2C, SPI, а також широтно-імпульсну модуляцію PWM. Це дозволяє підключати до нього як прості дискретні датчики, так і складніші цифрові модулі, релейні блоки, GSM-модулі та інші пристрої.

Для системи безпеки важливим є також питання захисту інформації. ESP32 підтримує апаратне прискорення окремих криптографічних операцій, що може використовуватися для підвищення рівня захисту даних при передаванні інформації через мережу. У межах бакалаврської роботи це можна розглядати як додаткову перевагу платформи, оскільки система безпеки працює з даними про стан приватного житлового об'єкта [18].

Ще однією важливою особливістю ESP32 є підтримка енергоощадних режимів роботи. У режимі зниженого енергоспоживання контролер може тимчасово вимикати частину своїх функціональних блоків і переходити в очікування. Це особливо важливо для систем, які повинні зберігати працездатність при короткочасному зникненні основного живлення. У реальних умовах рівень енергоспоживання залежить від конкретної плати, підключених модулів і режиму роботи, однак сама можливість енергоощадного функціонування є суттєвою перевагою ESP32.

Для обґрунтування вибору ESP32 проведено порівняння з іншими поширеними мікроконтролерними платформами, наприклад Arduino Uno та ESP8266. Порівняння наведено в таблиці 2.2.

Таблиця 2.2 – Порівняння можливих платформ для реалізації системи безпеки

Характеристика	Arduino Uno	ESP8266	ESP32
Наявність Wi-Fi	Немає, потрібен додатковий модуль	Є	Є
Наявність Bluetooth	Немає	Немає	Є
Кількість доступних інтерфейсів	Обмежена	Середня	Висока
Продуктивність	Невисока	Середня	Вища
Можливість роботи з аналоговими сигналами	Є	Обмежена	Є, декілька каналів ADC
Придатність для IoT-систем	Обмежена	Добра	Висока
Вартість	Низька	Низька	Низька або середня
Доцільність використання в системі безпеки	Потребує додаткових модулів	Можлива	Найбільш доцільна

Отже, використання ESP32 є обґрунтованим, оскільки цей мікроконтролер дозволяє поєднати функції локального керування, оброблення сигналів і бездротової передачі даних без значного ускладнення апаратної частини.

2.1.2 Вибір датчиків сенсорного рівня системи

Сенсорний рівень автоматизованої системи безпеки будинку призначений для збору інформації про стан контролюваного об'єкта. Саме датчики забезпечують первинне виявлення подій, які можуть свідчити про небезпеку. До складу сенсорного рівня доцільно включити датчики руху, відкриття дверей і вікон, диму, газу, затоплення, температури та вологості.

Для забезпечення комплексного контролю стану житлового будинку було обрано набір датчиків, які дозволяють контролювати як охоронні, так і аварійні параметри. До охоронних параметрів належать рух у приміщенні та відкриття дверей або вікон. До аварійних параметрів належать дим, газ, затоплення, температура і вологість.

Інфрачервоний датчик руху HC-SR501.

Для виявлення руху в приміщенні доцільно використати пасивний інфрачервоний датчик руху HC-SR501. Принцип його роботи ґрунтується на фіксації змін інфрачервоного випромінювання, яке випромінюють об'єкти з температурою, відмінною від температури навколишнього середовища. Якщо в

зоні контролю з'являється людина, датчик фіксує зміну теплового випромінювання і формує цифровий сигнал для мікроконтролера.

Перевагами датчика HC-SR501 є простота підключення, невисока вартість, достатня дальність виявлення та можливість регулювання чутливості. Датчик має цифровий вихід, тому його можна безпосередньо підключати до одного з GPIO-входів ESP32. У системі безпеки такий датчик доцільно встановлювати у коридорах, біля входу, у кімнатах або інших зонах, де поява руху в режимі охорони повинна розглядатися як потенційна тривожна подія [13].

Для зменшення кількості хибних спрацювань датчик руху потрібно правильно розміщувати. Його не слід спрямовувати безпосередньо на джерела тепла, вікна з прямим сонячним освітленням або вентиляційні отвори. Також у програмному забезпеченні доцільно передбачити коротку затримку або повторну перевірку сигналу перед формуванням тривоги.

Магнітоконттактний датчик МС-38.

Для контролю стану дверей і вікон доцільно використати магнітоконттактний датчик, або геркон, моделі МС-38. Цей датчик складається з двох частин: магніту та герметичного контакту. Одна частина встановлюється на нерухому частину дверної або віконної рами, а інша — на рухому частину. У зачиненому стані магніт утримує контакт у певному положенні, а при відкритті дверей або вікна контакт змінює свій стан.

Перевагами герконового датчика є простота конструкції, висока надійність, низьке енергоспоживання та стабільність роботи. На відміну від активних електронних датчиків, геркон не потребує складної обробки сигналу і може працювати тривалий час без обслуговування. У системі безпеки він використовується для контролю периметра будинку, тобто входів, балконних дверей, вікон і технічних люків.

Магнітоконттактний датчик доцільно підключати до цифрового входу ESP32 з використанням підтягувального резистора. У програмній частині необхідно враховувати можливе короткочасне дрижання контактів, тому для такого датчика доцільно застосовувати програмну фільтрацію сигналу.

Оптико-електронний датчик диму.

Для раннього виявлення пожежної небезпеки до складу системи доцільно включити оптико-електронний датчик диму. Принцип його роботи полягає у фіксації розсіювання світла на частинках диму. Якщо в камеру датчика потрапляють частинки диму, змінюється інтенсивність світлового потоку, що реєструється електронною схемою датчика. Після цього формується сигнал тривоги.

Оптико-електронні датчики диму є ефективними для виявлення займання на ранній стадії, особливо у випадках тління матеріалів, коли відкритого полум'я ще немає, але вже утворюється дим. У системі безпеки будинку такий датчик виконує функцію пожежного сповіщувача. Його доцільно встановлювати у коридорах, поблизу кухонної зони, у житлових кімнатах або технічних приміщеннях.

Для інтеграції з мікроконтролерною системою зручно використовувати датчики диму з релейним або цифровим виходом. Такий вихід дозволяє підключити датчик до ESP32 і використовувати його сигнал як умову для активації тривожного режиму. У разі виявлення диму система може увімкнути сирену, надіслати повідомлення власнику та, за потреби, активувати додаткові виконавчі пристрої.

Датчик витоку газу MQ-2 або MQ-5.

Для виявлення витоку газу доцільно використати датчик серії MQ, зокрема MQ-2 або MQ-5. Датчики цього типу базуються на напівпровідниковому чутливому елементі, опір якого змінюється при контакті з певними газами. Модель MQ-2 є універсальною і може реагувати на зріджений газ, пропан, водень, дим та інші гази. Модель MQ-5 більше орієнтована на виявлення природного газу, зокрема метану.

Датчики MQ мають аналоговий та цифровий вихід. Аналоговий вихід дозволяє зчитувати приблизний рівень концентрації газу за допомогою аналогово-цифрового перетворювача ESP32. Цифровий вихід формується компаратором і може використовуватися для швидкого виявлення перевищення встановленого порогу. У межах розроблюваної системи цифровий вихід можна використовувати

для формування тривожного сигналу, а аналоговий — для додаткового контролю рівня небезпеки.

Водночас слід враховувати, що датчики серії MQ мають нагрівальний елемент, тому споживають більше енергії, ніж, наприклад, геркони або датчики руху. Через це їх доцільно жити від основного джерела живлення системи. Також такі датчики потребують певного часу для прогрівання і не є високоточними вимірювальними приладами. У системі безпеки будинку їх доцільно використовувати саме як засіб виявлення небезпечного перевищення концентрації газу, а не як лабораторний газоаналізатор.

При спрацюванні датчика газу система повинна перейти в аварійний режим. У цьому режимі доцільно передбачити увімкнення звукової та світлової сигналізації, надсилання повідомлення власнику, а також можливість автоматичного перекриття подачі газу за допомогою електроприводного клапана.

Датчик затоплення на базі мікросхеми LM393.

Для виявлення витoku води або затоплення доцільно використати датчик на базі компаратора LM393. Принцип роботи такого датчика ґрунтується на зміні електричного опору між контактами сенсорної пластини або виносного зонда. У сухому стані опір між контактами є великим, а при потраплянні води він різко зменшується. Ця зміна фіксується електронною схемою і перетворюється на сигнал для мікроконтролера.

Датчик затоплення доцільно встановлювати у місцях, де найбільш імовірна поява води: під раковиною, біля пральної машини, біля бойлера, у санвузлі, на кухні або в технічному приміщенні. У разі виявлення води система може не лише повідомити власника, а й автоматично перекрити подачу води за допомогою електроприводного клапана.

Перевагами датчика затоплення є простота конструкції, невисока вартість і легкість підключення. У більшості модулів передбачено як аналоговий, так і цифровий вихід, тому користувач може налаштувати поріг спрацювання відповідно до умов експлуатації. Для підвищення надійності роботи датчик

потрібно встановлювати у нижній точці контрольованої зони, де вода з'явиться першою.

Датчик мікроклімату DHT22/AM2302 або DS18B20.

Для контролю параметрів мікроклімату в будинку доцільно використовувати цифрові датчики температури та вологості. Одним із поширених варіантів є DHT22, також відомий як AM2302. Він дозволяє вимірювати температуру та відносну вологість повітря і передавати дані до мікроконтролера через однопровідний цифровий інтерфейс.

Датчик DHT22 є зручним для побудови систем моніторингу, оскільки поєднує в собі два вимірювальні канали: температуру і вологість. Дані з нього можуть використовуватися не лише для відображення поточного стану приміщення, а й для прийняття рішень у випадку небезпечного підвищення температури або вологості. Наприклад, значне підвищення температури може бути додатковою ознакою пожежної небезпеки, а висока вологість — свідчити про проблеми з вентиляцією або можливе затоплення.

Для вимірювання лише температури може застосовуватися датчик DS18B20. Його перевагою є достатня точність і можливість підключення за цифровою шиною 1-Wire. На відміну від DHT22, DS18B20 не вимірює вологість, проте є зручним для контролю температури у технічних зонах, біля трубопроводів або в місцях, де потрібне надійне температурне вимірювання.

У розроблюваній системі датчик мікроклімату виконує допоміжну, але важливу функцію. Він дозволяє підвищити інформативність системи та забезпечити додатковий контроль стану житлового середовища.

Узагальнені характеристики обраних датчиків наведено в таблиці 2.3.

Таблиця 2.3 – Обґрунтування вибору датчиків автоматизованої системи безпеки будинку

Датчик	Контрольований параметр	Тип сигналу	Призначення в системі	Причина вибору
HC-SR501	Рух у приміщенні	Цифровий	Виявлення несанкціонованого проникнення	Простота підключення, доступність, достатня зона контролю

Датчик	Контрольований параметр	Тип сигналу	Призначення в системі	Причина вибору
МС-38	Відкриття дверей або вікон	Цифровий	Контроль периметра будинку	Надійність, низьке енергоспоживання, простота монтажу
Оптико-електронний датчик диму	Наявність диму	Цифровий або релейний	Раннє виявлення пожежної небезпеки	Можливість виявлення займання на ранній стадії
MQ-2 / MQ-5	Витік газу	Аналоговий / цифровий	Виявлення небезпечної концентрації газу	Доступність, можливість налаштування порогу спрацювання
Датчик затоплення LM393	Наявність води	Аналоговий / цифровий	Виявлення протікання або затоплення	Низька вартість, простота встановлення
DHT22 / AM2302	Температура і вологість	Цифровий	Контроль мікроклімату	Поєднання вимірювання температури та вологості
DS18B20	Температура	Цифровий 1-Wire	Додатковий контроль температури	Стабільність і зручність підключення

2.1.3 Вибір виконавчих пристроїв системи

Виконавчі пристрої автоматизованої системи безпеки призначені для реалізації реакції на виявлену загрозу. Якщо датчики виконують функцію виявлення небезпечної події, то виконавчі пристрої забезпечують практичну дію системи: оповіщення людей, привернення уваги, перекриття небезпечних потоків або керування допоміжним обладнанням.

У розроблюваній системі до виконавчих пристроїв належать звукова сирена, світловий маяк, релейний модуль та електроприводні клапани для перекриття води або газу.

Звукова та світлова сигналізація.

Для оперативного оповіщення про небезпеку доцільно використати звукову сирену та світловий індикатор або стробоскопічний маяк. Звукова сигналізація забезпечує привернення уваги мешканців будинку, сусідів або охоронної служби. Світлова сигналізація є додатковим способом повідомлення про тривогу, особливо

в умовах високого рівня шуму або за потреби швидкої локалізації місця спрацювання.

Для системи безпеки доцільно використовувати сирену з живленням 12 В. Такий варіант є поширеним у охоронних системах і забезпечує достатню гучність сигналу. Світловий маяк може працювати разом із сиреною або окремо, залежно від типу тривожної події. Наприклад, при несанкціонованому проникненні вмикається і сирена, і світловий маяк, а при технічному попередженні може вмикатися лише світлова індикація або надсилатися повідомлення власнику.

Поєднання звукового і світлового оповіщення підвищує ефективність системи, оскільки забезпечує локальну реакцію навіть у випадку відсутності інтернет-зв'язку. Це важливо, оскільки система безпеки не повинна повністю залежати від зовнішніх сервісів.

Електроприводні клапани для перекриття води та газу.

Для автоматичного реагування на аварійні події, пов'язані з витоків води або газу, доцільно використовувати кульові крани з електроприводом. Такі пристрої дозволяють автоматично перекривати подачу води або газу після отримання відповідної команди від контролера.

У порівнянні з електромагнітними клапанами, кульові крани з електроприводом мають низку переваг. По-перше, вони закриваються поступово, що зменшує ризик виникнення гідроудару у водопровідній системі. По-друге, вони споживають електроенергію переважно під час зміни положення, тобто в момент відкриття або закриття. У сталому стані такий клапан не потребує постійного живлення, що є важливим для енергоефективності системи.

У розроблюваній системі електроприводний клапан може використовуватися для перекриття води при спрацюванні датчика затоплення. Аналогічно, при спрацюванні датчика газу система може сформувати команду на перекриття газового клапана. При цьому необхідно враховувати, що керування газовим обладнанням повинно виконуватися з дотриманням відповідних правил безпеки, а практичне встановлення таких пристроїв має здійснюватися кваліфікованими спеціалістами.

Релейний модуль.

Для керування виконавчими пристроями, які працюють від напруги 12 В або 220 В, необхідно використовувати проміжний комутаційний елемент. Таким елементом може бути релейний модуль з оптичною розв'язкою. Його призначення полягає у відокремленні низьковольтної керуючої частини системи від силових кіл виконавчих пристроїв.

ESP32 працює з логічними рівнями 3,3 В, тому безпосереднє підключення потужних навантажень до його виходів є неможливим і небезпечним. Релейний модуль дозволяє мікроконтролеру безпечно керувати сиреною, маяком, клапаном або іншим навантаженням. Наявність оптичної розв'язки додатково підвищує захист контролера від імпульсних завад і можливих пошкоджень.

У системі доцільно використовувати релейні модулі або транзисторні ключі залежно від типу навантаження. Для простих навантажень постійного струму можуть застосовуватися транзисторні ключі, а для комутації пристроїв з окремим живленням — релейні модулі.

Узагальнене обґрунтування вибору виконавчих пристроїв наведено в таблиці 2.4.

Таблиця 2.4 – Виконавчі пристрої автоматизованої системи безпеки будинку

Виконавчий пристрій	Призначення	Спосіб підключення	Реакція системи
Звукова сирена	Оповіщення про тривогу	Через релейний модуль або транзисторний ключ	Вмикається при проникненні, димі, газі або аварії
Світловий маяк	Візуальне оповіщення	Через релейний модуль	Дублює сигнал тривоги
Електроприводний клапан води	Перекриття води	Через релейний модуль	Активується при виявленні затоплення
Електроприводний клапан газу	Перекриття газу	Через релейний модуль	Активується при виявленні витоку газу
Релейний модуль	Комутація навантажень	GPIO ESP32	Забезпечує безпечне керування виконавчими пристроями

2.1.4 Вибір комунікаційних засобів

Комунікаційні засоби є важливою частиною автоматизованої системи безпеки, оскільки забезпечують передавання інформації користувачу. Навіть якщо

система локально виявила небезпеку і активувала сирену, власник будинку повинен отримати повідомлення про подію на смартфон або інший пристрій.

Основним каналом зв'язку в розроблюваній системі доцільно обрати Wi-Fi. Це пояснюється тим, що більшість житлових будинків і квартир мають домашню бездротову мережу, а мікроконтролер ESP32 уже містить вбудований Wi-Fi-модуль. Через Wi-Fi система може передавати повідомлення до мобільного застосунку, MQTT-брокера, Telegram-бота або іншого сервісу повідомлень.

Wi-Fi-канал доцільно використовувати для передавання поточного стану системи, даних з датчиків, журналу подій і команд керування. Наприклад, власник може отримувати повідомлення про спрацювання датчика руху, відкриття дверей, появу диму або виявлення затоплення. Також через Wi-Fi можна реалізувати віддалене ввімкнення або вимкнення режиму охорони.

Однак система безпеки не повинна повністю залежати від одного каналу зв'язку. У разі відключення інтернету, несправності маршрутизатора або зникнення електроживлення домашньої мережі Wi-Fi-повідомлення можуть бути недоступними. Тому доцільно передбачити резервний канал зв'язку на базі GSM/GPRS-модуля, наприклад SIM800L.

GSM-модуль дозволяє надсилати SMS-повідомлення або виконувати тривожний дзвінок на номер власника. Такий канал є корисним у критичних ситуаціях, оскільки він не залежить від домашнього інтернету. У разі втрати Wi-Fi-з'єднання ESP32 може перейти на резервний сценарій інформування і передати повідомлення через GSM-модуль.

Для підключення GSM-модуля до ESP32 доцільно використовувати інтерфейс UART. При цьому необхідно враховувати вимоги модуля до живлення, оскільки під час передавання даних GSM-модулі можуть короткочасно споживати значний струм. Тому для них бажано передбачити окреме стабілізоване живлення.

Комбіноване використання Wi-Fi та GSM підвищує надійність системи. Wi-Fi є основним каналом для зручного повсякденного моніторингу, а GSM — резервним каналом для інформування у разі втрати інтернет-з'єднання.

2.1.5 Вибір системи живлення

Для стабільної роботи автоматизованої системи безпеки важливо правильно організувати живлення всіх її компонентів. Центральний контролер ESP32 зазвичай живиться від напруги 5 В через плату розробника або від стабілізованої напруги 3,3 В безпосередньо на відповідний вхід. Датчики можуть працювати від 3,3 В або 5 В залежно від їхньої моделі. Виконавчі пристрої, зокрема сирена, світловий маяк і електроприводні клапани, часто потребують живлення 12 В.

Тому в системі доцільно передбачити окремі лінії живлення для логічної та силової частини. Логічна частина включає ESP32 і датчики, а силова частина — реле, сирену, клапани та інші виконавчі механізми. Для узгодження різних рівнів напруги можуть використовуватися стабілізатори напруги або DC-DC перетворювачі.

Окрему увагу слід приділити резервному живленню. Система безпеки повинна зберігати працездатність хоча б протягом певного часу після зникнення основного електроживлення. Для цього можна використати акумуляторний блок або джерело безперебійного живлення. У разі зникнення напруги 220 В система повинна залишатися здатною обробляти сигнали основних датчиків, активувати локальну сигналізацію і, за можливості, надіслати повідомлення власнику.

Узагальнений розподіл живлення компонентів системи наведено в таблиці 2.5.

Таблиця 2.5 – Орієнтовний розподіл живлення технічних засобів системи

Компонент системи	Орієнтовна напруга живлення	Особливості живлення
ESP32	5 В або 3,3 В	Потребує стабілізованого живлення
Датчик руху HC-SR501	5 В	Підключається до цифрового входу контролера
Геркон MC-38	3,3 В або 5 В у складі кола	Має низьке енергоспоживання
Датчик газу MQ-2/MQ-5	5 В	Потребує живлення нагрівального елемента
Датчик затоплення LM393	3,3 В або 5 В	Може мати аналоговий і цифровий вихід
DHT22 / AM2302	3,3 В або 5 В	Передає цифрові дані до контролера

Компонент системи	Орієнтовна напруга живлення	Особливості живлення
GSM-модуль SIM800L	Окреме стабілізоване живлення	Потребує достатнього струму під час передавання
Сирена, маяк, клапани	Переважно 12 В	Керуються через реле або силові ключі

Для підвищення надійності живлення доцільно передбачити захист від короткого замикання, перевантаження та переполюсування. Також бажано розділити живлення силових і логічних елементів, щоб спрацювання сирени або клапана не призводило до перезавантаження мікроконтролера.

2.1.6 Узагальнення вибору технічних засобів системи

На основі аналізу вимог до автоматизованої системи управління безпекою будинку було обрано технічні засоби, які забезпечують комплексний контроль стану житлового об'єкта. Центральним елементом системи є мікроконтролер ESP32, який виконує функції збору, оброблення та передавання даних. До сенсорного рівня входять датчики руху, відкриття дверей і вікон, диму, газу, затоплення, температури та вологості. Виконавчий рівень представлений сиреною, світловим маяком, релейним модулем і електроприводними клапанами. Для передавання повідомлень користувачу передбачено використання основного Wi-Fi-каналу та резервного GSM/GPRS-каналу.

Обраний склад технічних засобів дозволяє реалізувати основні функції системи: виявлення несанкціонованого проникнення, контроль аварійних ситуацій, локальне оповіщення, автоматичне реагування на небезпеку та дистанційне інформування власника. При цьому система залишається відносно простою, економічно доступною та придатною для реалізації на базі поширених мікроконтролерних компонентів.

2.2 Проктування автоматизованої системи безпеки засобами UML-моделювання

Для формалізованого опису структури та логіки функціонування автоматизованої системи управління безпекою будинку доцільно використати засоби UML-моделювання. UML, або уніфікована мова моделювання, є одним із

поширених інструментів візуального опису програмних та автоматизованих систем. Її використання дозволяє наочно подати склад системи, визначити основних учасників взаємодії, описати послідовність оброблення тривожних подій та відобразити загальну логіку роботи програмного забезпечення.

Застосування UML-моделювання є доцільним на етапі проєктування автоматизованої системи безпеки, оскільки така система складається з кількох взаємопов'язаних частин: сенсорного рівня, обчислювального центру, виконавчих пристроїв, каналів зв'язку та інтерфейсу користувача. Без попереднього моделювання складно забезпечити узгоджену роботу всіх компонентів, оскільки кожен датчик може формувати окремий тип події, а система повинна правильно визначити її рівень небезпеки та виконати відповідну реакцію [12].

UML-моделювання дає змогу абстрагуватися від безпосереднього написання програмного коду та зосередитися на логіці функціонування системи. На цьому етапі визначаються основні функції системи, способи взаємодії користувача з нею, порядок оброблення сигналів від датчиків, сценарії реагування на аварійні події та структура програмних процесів. Це дозволяє зменшити кількість помилок на етапі програмної реалізації та забезпечити кращу відповідність майбутнього програмного забезпечення поставленим вимогам.

У межах розроблення автоматизованої системи управління безпекою будинку доцільно використати три типи UML-діаграм:

- діаграму варіантів використання, яка визначає функціональні можливості системи та взаємодію з користувачем;
- діаграму послідовності, яка описує порядок обміну повідомленнями між елементами системи під час виникнення тривоги;
- діаграму діяльності, яка відображає загальний алгоритм роботи системи та логіку переходу між станами.

Використання цих діаграм дозволяє комплексно описати автоматизовану систему безпеки як з точки зору користувача, так і з точки зору внутрішньої логіки її роботи.

2.2.1 Діаграма варіантів використання

Діаграма варіантів використання є одним із базових засобів UML-моделювання, який дозволяє описати функціональність системи з позиції її користувачів та зовнішніх сервісів. Така діаграма визначає межі системи, основних акторів, які взаємодіють із нею, а також набір функцій, доступних для виконання.

Для автоматизованої системи управління безпекою будинку основним користувачем є власник будинку. Саме він ініціює більшість дій, пов'язаних із налаштуванням системи, постановкою об'єкта під охорону, переглядом стану датчиків та отриманням повідомлень про тривожні події. Крім користувача, із системою взаємодіє сервіс повідомлень, який забезпечує доставку сповіщень на смартфон або інший пристрій користувача. Також у межах діаграми відображається внутрішня автоматизована логіка системи, яка забезпечує автономне реагування на критичні події без постійного втручання людини [17].

Діаграма варіантів використання автоматизованої системи безпеки будинку наведена на рисунку 2.2.



Рисунок 2.2 – UML-діаграма варіантів використання АСУББ

На діаграмі показано основні варіанти використання системи: постановка під охорону, перевірка датчиків, моніторинг, конфігурування, отримання повідомлень і реагування на аварії. Ці функції відображають базову логіку роботи автоматизованої системи безпеки будинку.

Основним учасником взаємодії (актором) є власник будинку. Він виконує керування системою та отримує інформацію про її стан. До основних дій власника

належать постановка системи під охорону, перегляд поточного стану об'єкта, налаштування параметрів системи та отримання сповіщень про тривогу.

Другим важливим учасником є сервіс повідомлень. Він виступає зовнішнім компонентом, через який користувач отримує інформацію про події, зафіксовані системою. У ролі такого сервісу можуть використовуватися мобільний застосунок, Telegram Bot API, MQTT-брокер, GSM-шлюз або інший засіб передавання повідомлень. Сервіс повідомлень не приймає рішення щодо безпеки об'єкта, але забезпечує оперативне інформування користувача.

У межах роботи системи також враховується автоматизована логіка безпеки, яка реалізується на базі мікроконтролера ESP32. Вона забезпечує автономний контроль датчиків, визначення небезпечних ситуацій та активацію виконавчих пристроїв. Наприклад, якщо у режимі охорони датчик руху фіксує переміщення в приміщенні, система може самостійно активувати сирену та сформулювати повідомлення власнику.

Основні учасники взаємодії (актори) та їхні функції наведено в таблиці 2.6.

Таблиця 2.6 – Учасники взаємодії UML-діаграми варіантів використання

Учасник взаємодії	Роль у системі	Основні дії
Власник будинку	Основний користувач системи	Постановка під охорону, моніторинг, конфігурування, отримання повідомлень
Сервіс повідомлень	Зовнішній засіб передавання інформації	Доставка повідомлень про тривожні та аварійні події
Автоматизована логіка системи	Внутрішній програмно-апаратний механізм реагування	Контроль датчиків, виявлення загроз, активація виконавчих пристроїв

Варіант використання «Постановка під охорону» описує процес переведення системи у режим активного контролю. Після вибору цього режиму система повинна перевірити стан усіх основних датчиків. Якщо двері або вікна зачинені, датчики справні, а аварійних подій не виявлено, система переходить у режим охорони. Якщо ж один із датчиків фіксує небезпечний або некоректний стан, система повинна повідомити користувача про неможливість постановки під охорону або сформулювати попередження.

Варіант використання «Перевірка датчиків» є допоміжною функцією, яка виконується під час постановки системи під охорону або під час технічного

контролю. Його призначення полягає у визначенні поточного стану сенсорного рівня системи. Перевіряються датчики руху, відкриття дверей і вікон, диму, газу, затоплення, температури та вологості.

Варіант використання «Моніторинг» відображає процес постійного спостереження за станом об'єкта. У цьому режимі система циклічно отримує дані від датчиків, порівнює їх із заданими умовами та визначає, чи є підстави для формування тривоги. Моніторинг може виконуватися як у режимі охорони, так і в режимі звичайного спостереження за аварійними параметрами.

Варіант використання «Конфігурування» передбачає зміну налаштувань системи. До таких налаштувань можуть належати вибір режиму роботи, встановлення порогових значень для датчиків газу або температури, налаштування номера телефону для SMS-повідомлень, зміна параметрів Wi-Fi-з'єднання, увімкнення або вимкнення окремих датчиків [24].

Варіант використання «Отримання повідомлень» описує процес інформування користувача про події, зафіксовані системою. Повідомлення можуть надсилатися у випадку спрацювання датчика руху, відкриття дверей або вікна, появи диму, витоку газу, затоплення, втрати зв'язку або зникнення основного живлення.

Варіант використання «Реагування на аварії» описує дії системи у випадку виявлення критичної події. До таких дій належать увімкнення сирени, активація світлового маяка, перекриття подачі води або газу, запис події в журнал та надсилання повідомлення користувачу.

На діаграмі використовуються відношення *include* та *extend*. Відношення *include* означає, що один варіант використання є обов'язковою частиною іншого. Наприклад, постановка системи під охорону обов'язково включає перевірку стану датчиків. Без такої перевірки система не може достовірно визначити, чи готовий об'єкт до охорони.

Відношення *extend* означає, що певний варіант використання виконується лише за наявності відповідної умови. Наприклад, реагування на аварії розширює процес отримання повідомлень, оскільки додаткові дії виконуються тільки у

випадку виявлення критичної події. Якщо загроза не зафіксована, система продовжує працювати у звичайному режимі моніторингу.

Таким чином, діаграма варіантів використання дозволяє визначити основні функції автоматизованої системи безпеки та показати, яким чином користувач і зовнішні сервіси взаємодіють із нею.

2.2.2 Діаграма послідовності

Діаграма послідовності використовується для опису динамічної поведінки системи. Вона показує, у якій послідовності об'єкти системи обмінюються повідомленнями під час виконання певного сценарію. Для автоматизованої системи безпеки доцільно розглянути сценарій виникнення тривожної події, оскільки саме він є одним із найважливіших для оцінювання працездатності системи.

У межах цього сценарію беруть участь такі об'єкти: датчик, мікроконтролер ESP32, реле або сирена, сервіс повідомлень і смартфон користувача. Датчик є джерелом первинного сигналу. ESP32 виконує оброблення сигналу та приймає рішення. Реле або сирена реалізують локальну реакцію системи. Сервіс повідомлень забезпечує передавання інформації, а смартфон користувача є кінцевим пристроєм отримання сповіщення.

Діаграма послідовності роботи системи при виникненні тривоги наведена на рисунку 2.3.

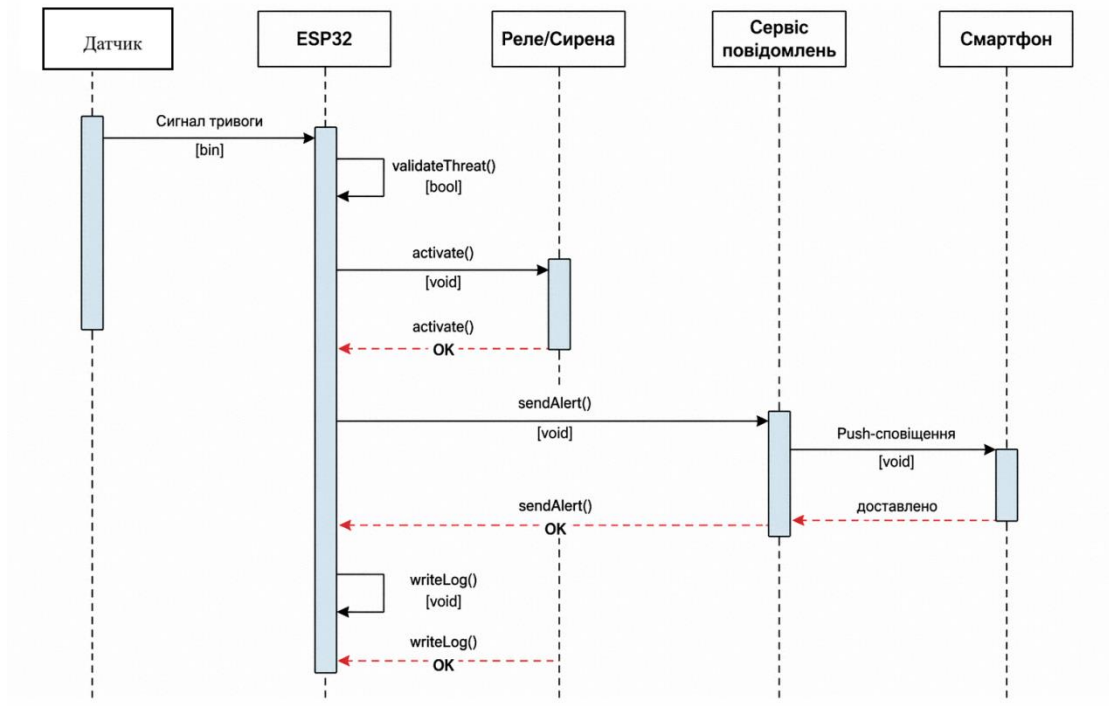


Рисунок 2.3 – UML-діаграма послідовності роботи системи при виникненні тривоги

Сценарій починається з того, що один із датчиків фіксує зміну контрольованого параметра. Це може бути рух у приміщенні, відкриття дверей, поява диму, підвищення концентрації газу або виявлення води в контрольованій зоні. Після цього датчик формує сигнал, який надходить на відповідний вхід мікроконтролера ESP32.

Контролер отримує сигнал і виконує його первинне оброблення. Для цифрових датчиків перевіряється логічний рівень сигналу, наприклад HIGH або LOW. Для аналогових датчиків, зокрема датчика газу, виконується зчитування значення з аналогового входу та порівняння його з установленим порогом. У разі необхідності система може виконати повторну перевірку сигналу, щоб зменшити ймовірність хибного спрацювання.

Після отримання сигналу контролер виконує процедуру підтвердження загрози. У програмній реалізації така процедура може бути представлена окремим методом, наприклад `validateThreat()`. Її призначення полягає у перевірці того, чи справді зафіксована подія є небезпечною. Наприклад, рух у приміщенні є тривожною подією лише тоді, коли система перебуває в режимі охорони. Водночас

поява диму, витік газу або затоплення є аварійними подіями незалежно від режиму охорони.

Якщо загроза не підтверджується, система повертається до режиму моніторингу. Якщо ж небезпечну подію підтверджено, ESP32 формує керуючий сигнал для виконавчого пристрою. У випадку охоронної тривоги вмикається сирена або світловий маяк. У випадку затоплення може активуватися електроприводний клапан перекриття води. У випадку витоку газу може формуватися команда на перекриття газового клапана та надсилання термінового повідомлення користувачу.

На діаграмі цей етап відображено як виклик методу `activate()` для об'єкта виконавчого пристрою. Після отримання команди реле або сирена переходить в активний стан і повертає контролеру підтвердження виконання дії. Це дозволяє системі зафіксувати, що локальна реакція була виконана.

Паралельно або одразу після активації виконавчого пристрою контролер формує повідомлення для користувача. Для цього викликається процедура `sendAlert(msg)`, яка передає інформацію до сервісу повідомлень. Зміст повідомлення може містити тип події, час її виникнення, назву датчика та поточний режим роботи системи. Наприклад, повідомлення може мати такий зміст: «Тривога: зафіксовано рух у коридорі», «Аварія: виявлено витік газу», «Увага: спрацював датчик затоплення».

Сервіс повідомлень після отримання запиту доставляє push-сповіщення або SMS-повідомлення на смартфон користувача. Завдяки цьому власник будинку може оперативно дізнатися про подію навіть тоді, коли він перебуває поза межами об'єкта.

Окремим етапом є запис події до журналу. Для цього контролер може виконувати метод `writeLog()`. У журналі подій доцільно зберігати дату і час події, тип датчика, характер події, результат реагування та спосіб інформування користувача. Наявність журналу дозволяє аналізувати роботу системи, виявляти повторювані спрацювання та перевіряти коректність функціонування датчиків.

Послідовність дій системи у випадку виникнення тривоги можна описати таким чином:

1. датчик фіксує небезпечну подію;
2. сигнал надходить на вхід мікроконтролера ESP32;
3. контролер зчитує сигнал і виконує його первинну обробку;
4. система перевіряє достовірність події;
5. у разі підтвердження загрози активується виконавчий пристрій;
6. формується повідомлення для користувача;
7. повідомлення передається через сервіс сповіщень;
8. користувач отримує повідомлення на смартфон;
9. інформація про подію записується до журналу.

Отже, діаграма послідовності дозволяє описати порядок взаємодії між основними компонентами автоматизованої системи безпеки у випадку тривоги. Вона демонструє, що система повинна не лише виявити подію, а й підтвердити її, виконати локальну реакцію, повідомити користувача та зафіксувати подію для подальшого аналізу.

2.2.3 Діаграма діяльності

Діаграма діяльності призначена для опису алгоритмічної логіки роботи системи. Вона дозволяє показати послідовність дій, умови переходів, розгалуження процесів і повернення системи до основного циклу роботи. Для автоматизованої системи управління безпекою будинку така діаграма є важливою, оскільки вона відображає загальний принцип роботи програмного забезпечення мікроконтролера.

Діаграма діяльності автоматизованої системи безпеки наведена на рисунку 2.4.

Робота системи починається з етапу ініціалізації. На цьому етапі мікроконтролер ESP32 виконує початкове налаштування апаратних і програмних компонентів. Зокрема, налаштовуються входи та виходи GPIO, перевіряється доступність датчиків, встановлюється з'єднання з Wi-Fi-мережею, ініціалізується GSM-модуль, задаються початкові стани реле та виконавчих пристроїв. Також

можуть бути встановлені порогові значення для датчиків газу, температури, вологості або інших параметрів.

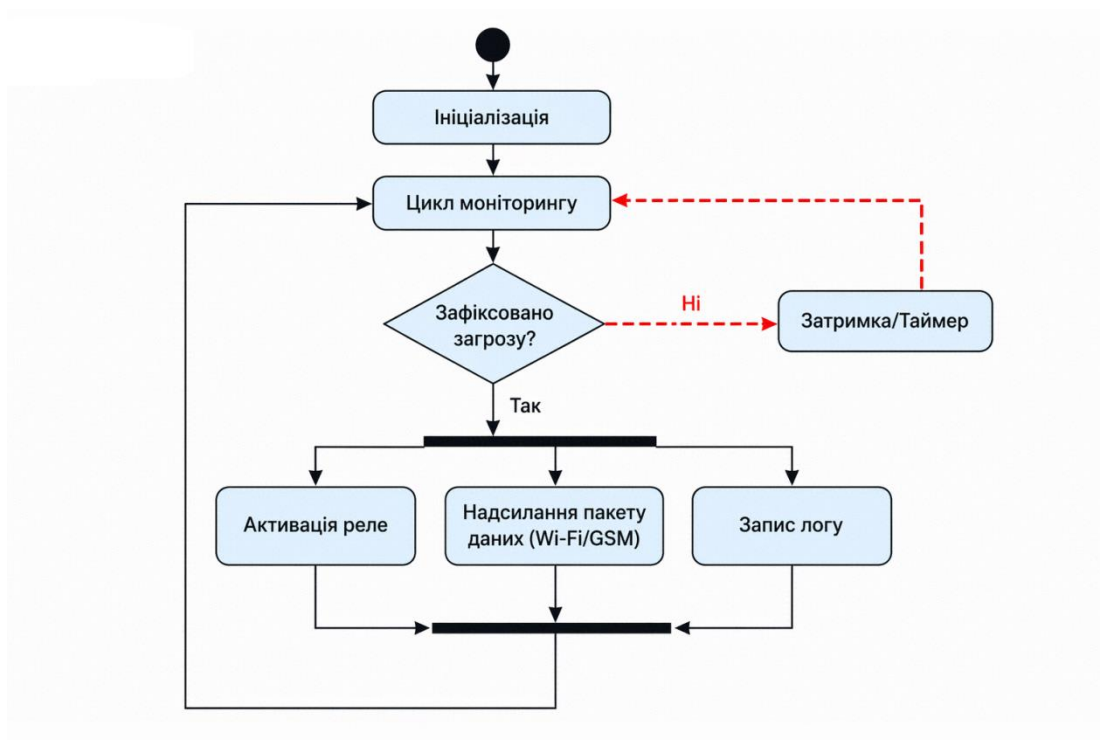


Рисунок 2.4 – UML-діаграма діяльності автоматизованої системи безпеки

Після завершення ініціалізації система переходить у цикл моніторингу. Цикл моніторингу є основним режимом роботи автоматизованої системи безпеки. У цьому режимі мікроконтролер періодично опитує датчики, зчитує їхні значення та аналізує отримані дані. Для цифрових датчиків перевіряється стан логічного сигналу, а для аналогових — виконується перетворення значення АЦП і порівняння з установленим порогом.

У процесі моніторингу система перевіряє, чи зафіксовано загрозу. Під загрозою розуміється така подія або стан, які потребують негайної реакції. Наприклад, загрозою може бути рух у приміщенні при увімкненому режимі охорони, відкриття дверей або вікна, поява диму, перевищення допустимого рівня газу, виявлення води датчиком затоплення або небезпечне підвищення температури.

Якщо загрозу не зафіксовано, система переходить у режим очікування на певний проміжок часу. Цей етап на діаграмі представлено як затримка або таймер. Після завершення затримки система знову повертається до циклу моніторингу.

Такий підхід дозволяє зменшити навантаження на мікроконтролер і уникнути надмірно частого опитування датчиків.

Якщо загрозу зафіксовано, система переходить до блоку реагування. На цьому етапі процес розгалужується на кілька паралельних або послідовних дій. Насамперед активується локальний захист. До локального захисту належить увімкнення сирени, світлового маяка, реле або електроприводного клапана. Конкретний виконавчий пристрій залежить від типу виявленої події [12].

Наприклад, при спрацюванні датчика руху в режимі охорони система активує сирену і надсилає повідомлення власнику. При спрацюванні датчика затоплення система може активувати клапан перекриття води та сформувати аварійне повідомлення. При виявленні газу система повинна повідомити власника, активувати сигналізацію та, за можливості, сформувати команду на перекриття газу. При виявленні диму система активує пожежне попередження та передає повідомлення користувачу.

Другим напрямом реагування є формування та надсилання пакета даних через доступний канал зв'язку. Основним каналом є Wi-Fi, оскільки він забезпечує швидке передавання даних і може використовуватися для роботи з мобільним застосунком або хмарним сервісом. Якщо Wi-Fi-з'єднання недоступне, може використовуватися резервний GSM-канал. Такий підхід підвищує надійність системи, оскільки користувач може отримати повідомлення навіть у разі втрати інтернет-з'єднання.

Третім напрямом є запис події до журналу. Запис журналу дозволяє зберігати інформацію про всі важливі події системи. До журналу можуть вноситися тип події, час спрацювання, стан датчика, активовані виконавчі пристрої та результат надсилання повідомлення. Це є корисним для подальшого аналізу роботи системи, виявлення несправностей і перевірки коректності алгоритмів.

Після виконання дій реагування система об'єднує потоки виконання і повертається до основного циклу моніторингу. Це означає, що після оброблення однієї тривожної події система не завершує роботу, а продовжує контролювати

стан будинку. Такий принцип є важливим для автоматизованих систем безпеки, які повинні працювати безперервно.

Загальна логіка роботи системи за діаграмою діяльності має такий вигляд:

1. подача живлення на систему;
2. ініціалізація мікроконтролера, датчиків, виконавчих пристроїв і каналів зв'язку;
3. перехід до циклу моніторингу;
4. зчитування даних із датчиків;
5. перевірка наявності загрози;
6. якщо загрози немає — перехід у режим очікування і повернення до моніторингу;
7. якщо загроза є — активація локального захисту;
8. надсилання повідомлення користувачу;
9. запис інформації до журналу подій;
10. повернення до циклічного контролю.

Діаграма діяльності дозволяє чітко визначити основні етапи функціонування програмного забезпечення автоматизованої системи безпеки. Вона є основою для подальшого розроблення алгоритмів роботи системи у третьому розділі бакалаврської роботи.

Таким чином, UML-моделювання автоматизованої системи безпеки будинку дозволило визначити функціональні можливості системи, описати взаємодію користувача з програмно-апаратним комплексом, установити послідовність оброблення тривожних подій та сформувану загальну алгоритмічну логіку роботи. Побудовані UML-діаграми забезпечують перехід від загального опису системи до її структурного й алгоритмічного проєктування.

2.3 Розроблення структурної схеми автоматизованої системи безпеки

Після вибору технічних засобів і побудови UML-моделей доцільно перейти до розроблення структурної схеми автоматизованої системи управління безпекою будинку. Структурна схема дозволяє наочно відобразити склад системи,

взаємозв'язки між її основними елементами, напрямки передавання даних та принцип взаємодії між сенсорним, обчислювальним і виконавчим рівнями.

Розроблення структурної схеми є важливим етапом проєктування, оскільки саме вона показує, як окремі технічні засоби об'єднуються в єдину автоматизовану систему. На відміну від UML-діаграм, які описують переважно логіку роботи та взаємодію користувача із системою, структурна схема відображає апаратно-функціональну організацію системи. Вона дає змогу зрозуміти, які датчики використовуються для збору інформації, який пристрій виконує оброблення даних, які виконавчі механізми реалізують реакцію на небезпеку та яким чином користувач отримує інформацію про стан об'єкта [3].

Автоматизована система управління безпекою будинку побудована за трирівневою структурою. Такий підхід дозволяє чітко розподілити функції між окремими частинами системи та забезпечити її модульність. У межах запропонованої структури виділено нижній рівень, який відповідає за отримання інформації від датчиків; середній рівень, на якому відбувається оброблення даних; і верхній рівень, який забезпечує реагування системи та взаємодію з користувачем.

Структурна схема автоматизованої системи управління безпекою будинку наведена на рисунку 2.5.

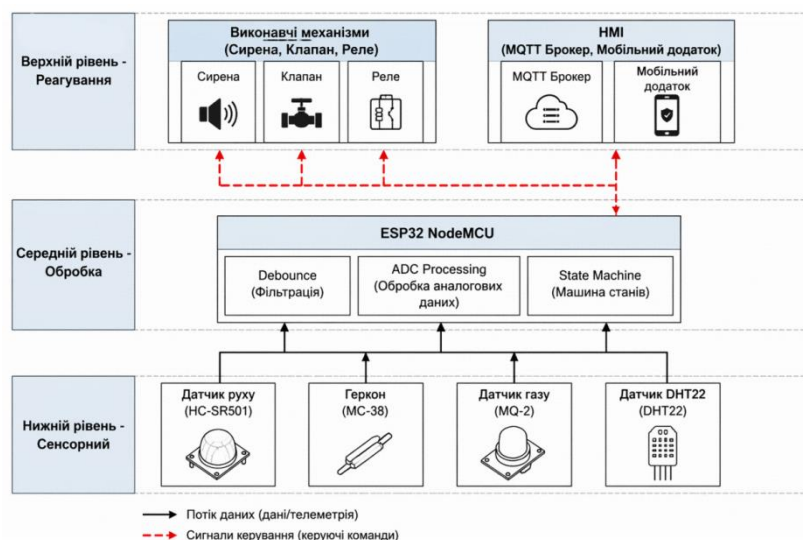


Рисунок 2.5 – Структурна схема автоматизованої системи управління безпекою будинку

Як видно з рисунка 2.5, система має ієрархічну структуру. Дані від датчиків надходять на мікроконтролер ESP32, який є центральним обчислювальним елементом системи. Після оброблення отриманої інформації контролер формує керуючі сигнали для виконавчих пристроїв або передає повідомлення користувачу через канали зв'язку. Така організація дозволяє реалізувати як локальне реагування на небезпеку, так і дистанційне інформування власника будинку.

2.3.1 Нижній рівень системи

Нижній рівень автоматизованої системи безпеки є сенсорним рівнем. Його основне призначення полягає у зборі первинної інформації про стан контрольованого об'єкта. До цього рівня належать датчики, які фіксують події або параметри, важливі для безпеки будинку: рух у приміщенні, відкриття дверей або вікон, появу диму, витік газу, затоплення, температуру та вологість.

Сенсорний рівень є основою всієї системи, оскільки саме від достовірності сигналів датчиків залежить правильність подальших рішень. Якщо датчик некоректно фіксує подію або передає нестабільний сигнал, це може призвести до хибного спрацювання або, навпаки, до пропуску небезпечної ситуації. Тому під час проєктування системи важливо не лише правильно обрати датчики, а й визначити спосіб їхнього підключення до мікроконтролера.

У запропонованій системі використовуються різні типи датчиків залежно від характеру контрольованого параметра. Частина датчиків формує дискретний сигнал, тобто передає до контролера один із двох можливих станів: спрацювання або відсутність спрацювання. До таких датчиків належать, наприклад, датчик руху HC-SR501 і магнітоконтактний датчик MC-38. Вони підключаються до цифрових входів мікроконтролера ESP32.

Датчик руху HC-SR501 призначений для виявлення переміщення людини в зоні контролю. У режимі охорони його спрацювання розглядається як потенційна ознака несанкціонованого проникнення. Магнітоконтактний датчик MC-38 використовується для контролю стану дверей або вікон. Якщо двері чи вікно відчиняються, датчик змінює стан свого виходу, і ця подія передається на мікроконтролер.

До сенсорного рівня також належать датчики, які можуть формувати аналоговий сигнал. Наприклад, датчик газу MQ-2 передає значення, що залежить від концентрації газу в повітрі. Для оброблення такого сигналу використовується аналогово-цифровий перетворювач мікроконтролера ESP32. Значення з датчика порівнюється з попередньо встановленим порогом, після чого система визначає, чи є ситуація небезпечною.

Для контролю параметрів мікроклімату може використовуватися цифровий датчик DHT22. Він передає до контролера значення температури та відносної вологості повітря. Ці дані можуть застосовуватися як для загального моніторингу стану приміщення, так і для додаткового виявлення небезпечних ситуацій. Наприклад, різке підвищення температури може бути непрямою ознакою пожежної небезпеки [16].

Датчик диму та датчик затоплення також входять до нижнього рівня системи. Датчик диму дозволяє виявити ознаки займання на ранній стадії, а датчик затоплення фіксує появу води у контрольованій зоні. Їхні сигнали передаються до мікроконтролера і використовуються для формування аварійного режиму роботи системи.

Таким чином, нижній рівень забезпечує постійне надходження інформації про стан будинку. Усі сигнали від сенсорів передаються до середнього рівня, де вони обробляються відповідно до заданої логіки роботи системи.

2.3.2 Середній рівень системи

Середній рівень є обчислювальним рівнем автоматизованої системи безпеки. Його центральним елементом є мікроконтролер ESP32 NodeMCU або модуль ESP32 WROOM-32, який виконує функції приймання, аналізу та оброблення даних від датчиків. Саме на цьому рівні реалізується основна логіка роботи системи.

Мікроконтролер ESP32 отримує сигнали від сенсорного рівня через цифрові та аналогові входи. Після цього він виконує попередню обробку даних. Для дискретних датчиків визначається стан входу, тобто наявність або відсутність спрацювання. Для аналогових датчиків виконується зчитування значення АЦП і

порівняння з установленим порогом. Для цифрових датчиків мікроклімату зчитуються значення температури та вологості.

Однією з важливих функцій середнього рівня є фільтрація помилкових або нестабільних сигналів. Наприклад, для механічних контактів може виникати явище дрижання контактів, коли під час замикання або розмикання сигнал короткочасно змінюється кілька разів. Для усунення цього ефекту у програмному забезпеченні використовується програмна фільтрація, або *debounce*. Вона дозволяє перевірити стабільність сигналу перед тим, як система визнає подію достовірною.

Для аналогових датчиків, зокрема датчика газу, середній рівень виконує оброблення аналогового сигналу. Значення напруги з виходу датчика перетворюється на цифрове значення за допомогою АЦП. Після цього система порівнює отримане значення з порогом небезпеки. Якщо поріг перевищено, формується аварійна подія [25].

Ще одним важливим елементом середнього рівня є локальний автомат станів. Він визначає поточний режим роботи системи та правила переходу між режимами. Для автоматизованої системи безпеки будинку доцільно передбачити такі основні стани:

- режим очікування;
- режим охорони;
- режим тривоги;
- аварійний режим;
- режим технічного обслуговування.

У режимі очікування система контролює аварійні датчики, але не формує охоронну тривогу за датчиком руху. У режимі охорони система активно контролює проникнення до будинку. У режимі тривоги активуються виконавчі пристрої та надсилається повідомлення користувачу. Аварійний режим використовується при виявленні диму, газу або затоплення. Режим технічного обслуговування може застосовуватися для перевірки працездатності датчиків і виконавчих пристроїв.

Середній рівень також виконує функцію формування керуючих сигналів. Якщо виявлено тривожну або аварійну подію, ESP32 подає сигнал на відповідний

вихід, який через релейний модуль або транзисторний ключ керує сиреною, світловим маяком, електроприводним клапаном або іншим виконавчим пристроєм.

Крім того, мікроконтролер забезпечує передавання повідомлень на верхній рівень системи. Для цього використовується Wi-Fi-з'єднання або резервний GSM-канал. Через Wi-Fi система може передавати інформацію до MQTT-брокера, мобільного застосунку або іншого програмного інтерфейсу. У разі втрати Wi-Fi-зв'язку може використовуватися GSM-модуль, який забезпечує надсилання SMS-повідомлення або тривожного дзвінка.

Отже, середній рівень є центральною частиною структурної схеми. Він поєднує сенсорний рівень із виконавчим рівнем, забезпечує оброблення даних, прийняття рішень і формування команд керування.

2.3.3 Верхній рівень системи

Верхній рівень автоматизованої системи безпеки відповідає за реагування на події та взаємодію з користувачем. До нього належать виконавчі пристрої, засоби локального оповіщення, канали зв'язку та інтерфейс користувача.

Виконавчі пристрої призначені для практичної реалізації рішень, прийнятих мікроконтролером. Якщо система виявляє небезпечну подію, верхній рівень забезпечує відповідну реакцію. У разі несанкціонованого проникнення активується звукова сирена та світловий маяк. У разі затоплення може спрацювати електроприводний клапан перекриття води. У разі витоку газу система може сформувати команду на перекриття газу та надіслати термінове повідомлення користувачу.

Звукова сирена та світловий маяк є засобами локального оповіщення. Їхнє призначення полягає у приверненні уваги мешканців будинку, сусідів або охоронної служби. Локальне оповіщення є важливим, оскільки воно працює незалежно від доступності інтернету. Навіть якщо зв'язок із зовнішніми сервісами втрачено, система все одно може виконати базову захисну функцію.

Електроприводні клапани використовуються для автоматичного перекриття води або газу. Такі пристрої підключаються через релейний модуль або інший комутаційний елемент, оскільки мікроконтролер не може безпосередньо керувати

силовими навантаженнями. Релейний модуль забезпечує гальванічне розділення між низьковольтною логічною частиною системи та виконавчими пристроями, які можуть працювати від іншої напруги.

Крім локальних виконавчих пристроїв, до верхнього рівня належать засоби взаємодії з користувачем. Для цього може використовуватися мобільний застосунок, вебінтерфейс, Telegram-бот або MQTT-клієнт. Через такий інтерфейс користувач може отримувати повідомлення про стан системи, переглядати інформацію з датчиків, отримувати тривожні сповіщення та за потреби змінювати режим роботи системи.

У структурній схемі також передбачено використання MQTT-брокера як проміжного програмного компонента для передавання повідомлень між ESP32 та клієнтським застосунком. MQTT є зручним протоколом для IoT-систем, оскільки дозволяє організувати обмін короткими повідомленнями між пристроями. У межах цієї роботи MQTT-брокер може розглядатися як один із можливих варіантів реалізації верхнього рівня взаємодії з користувачем.

Загальний розподіл функцій між рівнями автоматизованої системи безпеки наведено в таблиці 2.7.

Таблиця 2.7 – Рівні структурної схеми автоматизованої системи безпеки

Рівень системи	Основні компоненти	Основні функції
Нижній рівень — сенсорний	Датчик руху, геркон, датчик диму, датчик газу, датчик затоплення, DHT22	Збір інформації про стан об'єкта та передавання сигналів до контролера
Середній рівень — обчислювальний	ESP32, програмні модулі оброблення сигналів, автомат станів	Оброблення сигналів, перевірка умов тривоги, прийняття рішень
Верхній рівень — рівень реагування та взаємодії	Сирена, маяк, реле, електроприводні клапани, Wi-Fi/GSM, мобільний застосунок	Реагування на події, оповіщення користувача, керування виконавчими пристроями

2.3.4 Потоки даних і керуючих сигналів у системі

У структурній схемі автоматизованої системи безпеки можна виділити два основні типи зв'язків: інформаційні та керуючі. Інформаційні зв'язки забезпечують передавання даних від датчиків до мікроконтролера. Керуючі зв'язки забезпечують передавання команд від мікроконтролера до виконавчих пристроїв.

Інформаційний потік починається на сенсорному рівні. Датчики фіксують зміну параметра або подію та передають сигнал до ESP32. Наприклад, датчик руху передає цифровий сигнал про наявність руху, геркон — сигнал про відкриття дверей або вікна, датчик газу — аналогове значення рівня газу, а DHT22 — цифрові значення температури та вологості.

Після отримання даних мікроконтролер виконує їх оброблення. Він перевіряє, чи відповідають отримані значення умовам тривоги. Якщо умови тривоги не виконуються, система продовжує працювати у штатному режимі. Якщо ж виявлено небезпечну подію, формується керуючий сигнал.

Керуючий сигнал передається на виконавчий пристрій. Це може бути команда на увімкнення сирени, активацію світлового маяка, спрацювання реле або закриття електроприводного клапана. Одночасно система формує інформаційне повідомлення для користувача. Повідомлення може передаватися через Wi-Fi або GSM-зв'язок залежно від доступності каналу.

Такий розподіл потоків даних дозволяє системі працювати послідовно та зрозуміло: спочатку відбувається виявлення події, потім її аналіз, після цього — реагування та інформування користувача.

2.3.5 Надійність та автономність структурної схеми

Важливою вимогою до автоматизованої системи безпеки є здатність працювати автономно у разі втрати зв'язку з інтернетом або зовнішніми сервісами. Запропонована структурна схема передбачає, що основна логіка виявлення загроз і реагування реалізується безпосередньо на мікроконтролері ESP32. Це означає, що система не залежить повністю від хмарних сервісів.

Якщо зникає Wi-Fi-з'єднання, ESP32 продовжує отримувати сигнали від датчиків і виконувати локальні алгоритми безпеки. У разі виявлення диму, газу, затоплення або проникнення система може активувати сирену, світловий маяк або електроприводний клапан навіть без підключення до інтернету. Це підвищує живучість системи та робить її придатною для використання в реальних умовах.

Для підвищення надійності також передбачено резервний канал зв'язку на базі GSM-модуля. Якщо основний Wi-Fi-канал недоступний, система може

надіслати SMS-повідомлення або виконати тривожний дзвінок власнику будинку. Таким чином, поєднання локального реагування та резервного зв'язку дозволяє забезпечити стабільну роботу системи навіть у разі часткової відмови окремих компонентів.

Ще одним чинником надійності є модульність структури. Кожен датчик або виконавчий пристрій є окремим функціональним елементом. У разі потреби систему можна розширити, додавши нові датчики або виконавчі механізми. Наприклад, до неї можна підключити додаткові датчики руху, датчики розбиття скла, камеру відеоспостереження або модуль керування електрозамком.

2.3.6 Узагальнення структурної організації системи

Розроблена структурна схема автоматизованої системи управління безпекою будинку забезпечує логічний розподіл функцій між основними рівнями системи. Нижній рівень відповідає за отримання інформації від датчиків, середній рівень — за оброблення даних і прийняття рішень, а верхній рівень — за реагування на події та взаємодію з користувачем.

Запропонована структура є модульною, що спрощує її подальше розширення та модернізацію. Використання ESP32 як центрального контролера дозволяє поєднати оброблення сигналів, керування виконавчими пристроями та бездротовий зв'язок в одному пристрої. Наявність локальних виконавчих механізмів забезпечує автономне реагування на небезпечні події, а використання Wi-Fi і GSM підвищує надійність інформування користувача.

Висновки до розділу 2

У другому розділі було виконано проектування структури та технічного забезпечення автоматизованої системи управління безпекою будинку. Основну увагу приділено вибору апаратних компонентів, визначенню логіки їхньої взаємодії та побудові структурної моделі системи. Запропоновані рішення спрямовані на забезпечення комплексного контролю стану житлового об'єкта, своєчасне виявлення небезпечних подій та оперативне реагування на них.

У процесі обґрунтування вибору технічних засобів було визначено, що центральним елементом системи доцільно використати мікроконтролер ESP32.

Його вибір пояснюється наявністю достатньої обчислювальної продуктивності, вбудованих засобів бездротового зв'язку Wi-Fi та Bluetooth, підтримкою цифрових і аналогових інтерфейсів, а також можливістю роботи з різними типами датчиків і виконавчих пристроїв. Використання ESP32 дозволяє реалізувати функції збору, оброблення та передавання даних у межах одного керуючого вузла.

Для сенсорного рівня системи було обрано набір датчиків, які забезпечують контроль основних охоронних та аварійних параметрів будинку. До складу системи включено датчик руху, магнітоконттактний датчик відкриття дверей або вікон, датчик диму, датчик витоку газу, датчик затоплення та датчик мікроклімату. Такий набір технічних засобів дозволяє виявляти несанкціоноване проникнення, ознаки пожежної небезпеки, витік газу, появу води та відхилення температурно-вологісних параметрів.

Також було обґрунтовано вибір виконавчих пристроїв системи. Для локального оповіщення передбачено використання звукової сирени та світлового маяка. Для автоматичного реагування на аварійні події можуть застосовуватися електроприводні клапани перекриття води або газу. Керування виконавчими пристроями здійснюється через релейні модулі або інші комутаційні елементи, що забезпечує безпечне розділення низьковольтної керуючої частини та силових кіл.

Окрему увагу приділено каналам зв'язку. Основним каналом передавання повідомлень у системі є Wi-Fi, що забезпечує зручну взаємодію з користувачем через мобільний застосунок, хмарний сервіс або інший інтерфейс. Для підвищення надійності системи передбачено можливість використання резервного GSM-каналу, який дозволяє надсилати SMS-повідомлення або виконувати тривожний дзвінок у разі втрати інтернет-з'єднання.

З метою формалізованого опису логіки роботи системи було застосовано UML-моделювання. Розроблено діаграму варіантів використання, яка відображає основні функції системи та взаємодію користувача із нею. Діаграма послідовності дозволила описати порядок оброблення тривожної події: від спрацювання датчика до активації виконавчого пристрою, надсилання повідомлення користувачу та запису події до журналу. Діаграма діяльності відображає загальний алгоритм

функціонування системи, включаючи ініціалізацію, моніторинг, перевірку наявності загрози, реагування та повернення до циклічного контролю.

У межах розділу також було розроблено структурну схему автоматизованої системи безпеки будинку. Запропонована система має трирівневу організацію. Нижній рівень відповідає за отримання інформації від датчиків, середній рівень виконує оброблення сигналів і прийняття рішень на базі мікроконтролера ESP32, а верхній рівень забезпечує реагування на події, керування виконавчими пристроями та взаємодію з користувачем.

Запропонована структура системи є модульною та придатною до подальшого розширення. За потреби до неї можна додати нові датчики, виконавчі пристрої або додаткові канали зв'язку. Важливою перевагою розробленої структури є можливість автономного функціонування: навіть у разі втрати інтернет-з'єднання система здатна локально обробляти сигнали від датчиків і виконувати базові дії реагування, зокрема активувати сирену або перекрити подачу води чи газу.

Отже, у другому розділі було сформовано технічну та структурну основу автоматизованої системи управління безпекою будинку. Обґрунтовано вибір основних апаратних компонентів, визначено взаємодію між ними, побудовано UML-моделі та розроблено структурну схему системи. Отримані результати є основою для подальшого розроблення алгоритмічного та програмного забезпечення системи безпеки у третьому розділі роботи.

3 РОЗРОБЛЕННЯ АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ БЕЗПЕКИ

3.1 Визначення функціональних вимог до алгоритмічного та програмного забезпечення системи

Після обґрунтування вибору технічних засобів, розроблення UML-моделей і структурної схеми автоматизованої системи безпеки будинку необхідно визначити вимоги до її алгоритмічного та програмного забезпечення. Саме програмна частина забезпечує взаємодію між датчиками, мікроконтролером, виконавчими пристроями та користувачем. Від коректності розроблених алгоритмів залежить своєчасність

виявлення небезпечних подій, правильність прийняття рішень і надійність реагування системи.

Алгоритмічне забезпечення автоматизованої системи управління безпекою будинку має описувати послідовність дій, які виконуються системою під час моніторингу стану об'єкта. До таких дій належать ініціалізація обладнання, зчитування сигналів із датчиків, перевірка отриманих значень, визначення наявності загрози, активація виконавчих пристроїв, формування повідомлення користувачу та запис події до журналу. Програмне забезпечення, у свою чергу, реалізує ці алгоритми на базі мікроконтролера ESP32 [1].

Функціональні вимоги визначають, які саме дії повинна виконувати система. Для автоматизованої системи безпеки будинку ці вимоги формуються з урахуванням призначення системи, складу технічних засобів і можливих сценаріїв її роботи. Основна мета програмного забезпечення полягає у забезпеченні безперервного контролю стану житлового об'єкта та автоматичного реагування на охоронні й аварійні події.

До основних подій, які повинна контролювати система, належать:

- несанкціонований рух у приміщенні;
- відкриття дверей або вікон у режимі охорони;
- поява диму;
- перевищення допустимого рівня газу;
- поява води в контрольованій зоні;
- небезпечна зміна температури або вологості;
- втрата зв'язку з мережею;
- зникнення основного живлення, якщо у системі передбачено

його контроль.

Програмне забезпечення системи повинно забезпечити роботу в декількох основних режимах. Доцільно передбачити режим очікування, режим охорони, режим тривоги, аварійний режим і режим технічного обслуговування. Такий поділ дозволяє чітко визначити поведінку системи залежно від поточного стану об'єкта.

У режимі очікування система не формує охоронну тривогу за датчиком руху, однак продовжує контролювати аварійні параметри, зокрема дим, газ, затоплення, температуру та вологість. Це важливо, оскільки аварійні ситуації можуть виникнути незалежно від того, чи ввімкнений режим охорони.

У режимі охорони система здійснює активний контроль периметра та внутрішнього простору будинку. У цьому режимі спрацювання датчика руху або відкриття дверей чи вікна розглядається як потенційна тривожна подія. Після підтвердження такої події система повинна перейти в режим тривоги.

Режим тривоги передбачає активацію локальних засобів оповіщення, зокрема сирени та світлового маяка, а також формування повідомлення для користувача. У цьому режимі система повинна зафіксувати тип події, час її виникнення та виконані дії.

Аварійний режим використовується у випадку виявлення диму, витоку газу або затоплення. На відміну від охоронної тривоги, аварійний режим має спрацьовувати незалежно від того, чи перебуває система в режимі охорони. Наприклад, якщо датчик затоплення зафіксував появу води, система повинна надіслати повідомлення користувачу та, за наявності відповідного виконавчого пристрою, подати команду на перекриття води [24].

Режим технічного обслуговування призначений для перевірки працездатності датчиків і виконавчих пристроїв. У цьому режимі користувач або розробник може перевірити правильність підключення датчиків, роботу сирени, реле, клапанів і каналів зв'язку без формування повноцінної тривоги.

Основні функціональні вимоги до алгоритмічного та програмного забезпечення наведено в таблиці 3.1.

Таблиця 3.1 – Функціональні вимоги до програмного забезпечення автоматизованої системи безпеки будинку

Позначення вимоги	Функціональна вимога	Опис реалізації
F1	Ініціалізація системи	Після ввімкнення живлення система повинна налаштувати входи та виходи мікроконтролера, датчики, виконавчі пристрої та канали зв'язку

Позначення вимоги	Функціональна вимога	Опис реалізації
F2	Зчитування сигналів із датчиків	Система повинна періодично отримувати дані від датчиків руху, відкриття, диму, газу, затоплення, температури та вологості
F3	Контроль режиму охорони	Система повинна мати можливість переходу між режимом очікування та режимом охорони
F4	Виявлення несанкціонованого проникнення	У режимі охорони система повинна реагувати на спрацювання датчика руху або відкриття дверей/вікна
F5	Виявлення аварійних подій	Система повинна контролювати появу диму, витік газу, затоплення та небезпечні значення температури
F6	Перевірка достовірності сигналу	Перед формуванням тривоги система повинна виконувати коротку перевірку або фільтрацію сигналу для зменшення хибних спрацювань
F7	Активація виконавчих пристроїв	У разі підтвердження загрози система повинна вмикати сирену, світловий маяк, реле або електроприводний клапан
F8	Надсилання повідомлень користувачу	Система повинна формувати повідомлення про тривожні та аварійні події і передавати їх через доступний канал зв'язку
F9	Використання резервного каналу зв'язку	У разі недоступності основного Wi-Fi-з'єднання система повинна мати можливість використання GSM-каналу
F10	Запис подій до журналу	Система повинна зберігати інформацію про важливі події, тип спрацювання та виконані дії
F11	Повернення до циклу моніторингу	Після оброблення події система повинна продовжувати контроль стану об'єкта
F12	Тестування системи	Програмне забезпечення повинно передбачати можливість перевірки працездатності датчиків і виконавчих пристроїв

Важливою функціональною вимогою є правильна ініціалізація системи. Після подачі живлення мікроконтролер повинен налаштувати всі необхідні входи та виходи, встановити початкові стани виконавчих пристроїв, перевірити доступність датчиків і спробувати встановити з'єднання з мережею. На цьому етапі також можуть задаватися початкові параметри системи, зокрема режим роботи, порогові значення для датчиків газу або температури, номер телефону для сповіщення та параметри підключення до Wi-Fi.

Наступною вимогою є безперервне або періодичне зчитування даних із датчиків. Для цифрових датчиків система перевіряє логічний стан входу. Наприклад, датчик руху або герконовий датчик можуть передавати сигнал у вигляді логічного нуля або одиниці. Для аналогових датчиків, зокрема датчика газу, система повинна зчитувати значення з аналогового входу та порівнювати його з установленим порогом. Для цифрового датчика мікроклімату необхідно отримувати значення температури та вологості.

Окремо необхідно передбачити функцію контролю режиму охорони. Система не повинна однаково реагувати на датчик руху в усіх режимах. Якщо користувач перебуває вдома і система знаходиться в режимі очікування, рух у приміщенні не повинен спричиняти тривогу. Якщо ж увімкнено режим охорони, такий самий сигнал від датчика руху повинен розглядатися як ознака можливого проникнення.

Для зменшення кількості хибних спрацювань до функціональних вимог належить перевірка достовірності сигналу. Така перевірка може виконуватися різними способами: повторним зчитуванням сигналу через короткий проміжок часу, програмною фільтрацією дрижання контактів, порівнянням аналогового значення з порогом або перевіркою поєднання декількох умов. Наприклад, якщо геркон короткочасно змінив стан через нестабільний контакт, система не повинна одразу формувати тривогу без додаткової перевірки.

Особливу увагу потрібно приділити аварійним подіям. Якщо датчик диму, газу або затоплення фіксує небезпечний стан, система повинна реагувати незалежно від режиму охорони. Це пояснюється тим, що пожежа, витік газу або

затоплення становлять загрозу для майна та безпеки людей навіть тоді, коли будинок не перебуває під охороною.

Після підтвердження тривожної або аварійної події система повинна активувати відповідні виконавчі пристрої. У разі охоронної тривоги доцільно ввімкнути сирену та світловий маяк. У разі затоплення доцільно подати команду на перекриття води. У разі виявлення газу система може активувати сигналізацію, надіслати повідомлення власнику та, за наявності відповідного обладнання, подати команду на перекриття газу.

Формування повідомлень користувачу є однією з ключових функцій системи. Повідомлення повинно містити зрозумілу інформацію про характер події. Наприклад, замість загального тексту «Тривога» доцільно передавати конкретне повідомлення: «Виявлено рух у приміщенні», «Відкрито двері», «Виявлено дим», «Перевищено рівень газу», «Виявлено затоплення». Це дозволяє власнику швидше оцінити ситуацію та прийняти рішення.

Для підвищення надійності інформування користувача система повинна підтримувати використання основного та резервного каналів зв'язку. Основним каналом може бути Wi-Fi-з'єднання, через яке повідомлення передаються до мобільного застосунку, Telegram-бота або MQTT-брокера. Якщо Wi-Fi-з'єднання недоступне, система може використовувати GSM-модуль для надсилання SMS-повідомлення або виконання тривожного дзвінка [17].

Ще однією важливою вимогою є запис подій до журналу. Журнал подій дозволяє зберігати інформацію про роботу системи та подальше аналізування її стану. До журналу доцільно записувати дату й час події, тип датчика, характер спрацювання, режим системи, виконані дії та результат надсилання повідомлення. Навіть у простій реалізації журнал може бути представлений у вигляді послідовності текстових записів або структури даних у пам'яті пристрою.

Для подальшої реалізації програмного забезпечення доцільно визначити основні програмні модулі системи. Такий поділ спрощує розроблення, тестування і подальшу модернізацію програми.

Таблиця 3.2 – Основні програмні модулі автоматизованої системи безпеки

Програмний модуль	Призначення
Модуль ініціалізації	Налаштування мікроконтролера, датчиків, виконавчих пристроїв і каналів зв'язку
Модуль зчитування датчиків	Отримання цифрових та аналогових сигналів від сенсорів
Модуль аналізу подій	Перевірка умов тривоги та аварійних ситуацій
Модуль керування режимами	Перехід між режимами очікування, охорони, тривоги, аварії та обслуговування
Модуль керування виконавчими пристроями	Увімкнення сирени, маяка, реле та клапанів
Модуль повідомлень	Формування і передавання повідомлень користувачу
Модуль журналювання	Запис інформації про події системи
Модуль тестування	Перевірка працездатності основних компонентів системи

Алгоритмічне забезпечення системи повинно бути побудоване за принципом циклічного контролю. Це означає, що після запуску програма не завершує роботу, а постійно повторює основні дії: зчитує дані з датчиків, аналізує їх, перевіряє наявність загрози, виконує реакцію у разі потреби та повертається до моніторингу. Такий підхід відповідає принципу роботи більшості мікроконтролерних систем.

Загальну логіку роботи програмного забезпечення можна подати таким чином:

1. запуск системи;
2. ініціалізація мікроконтролера і підключених пристроїв;
3. встановлення зв'язку через Wi-Fi або GSM;
4. визначення поточного режиму роботи;
5. зчитування даних із датчиків;
6. перевірка умов тривоги або аварії;
7. підтвердження достовірності події;
8. активація виконавчих пристроїв;
9. надсилання повідомлення користувачу;
10. запис події до журналу;
11. повернення до циклу моніторингу.

Крім функціональних вимог, під час розроблення програмного забезпечення необхідно враховувати загальні експлуатаційні вимоги. Програма повинна бути достатньо простою, стабільною та зрозумілою для подальшого супроводу. Вона повинна коректно працювати після перезапуску живлення, не допускати

одночасного неконтрольованого ввімкнення виконавчих пристроїв і забезпечувати безпечний початковий стан системи після старту.

Також важливо, щоб програмне забезпечення мало можливість подальшого розширення. Наприклад, у майбутньому до системи можна додати датчик розбиття скла, камеру відеоспостереження, електронний замок, додатковий модуль живлення або новий спосіб повідомлення користувача. Тому структура програми повинна бути модульною, щоб нові функції можна було додавати без повного переписування всієї системи.

Отже, у межах цього підрозділу визначено основні функціональні вимоги до алгоритмічного та програмного забезпечення автоматизованої системи управління безпекою будинку. Визначені вимоги охоплюють ініціалізацію системи, зчитування сигналів із датчиків, контроль режимів роботи, виявлення тривожних та аварійних подій, активацію виконавчих пристроїв, надсилання повідомлень користувачу та запис подій до журналу. Сформульовані вимоги є основою для подальшого розроблення функціональної структури, алгоритмів функціонування та програмної реалізації системи.

3.2 Розроблення функціональної структури автоматизованої системи безпеки

Після визначення функціональних вимог до алгоритмічного та програмного забезпечення розроблено функціональну структуру автоматизованої системи управління безпекою будинку. Функціональна структура відображає склад основних програмно-логічних блоків системи, їхнє призначення, порядок взаємодії та напрямки передавання інформаційних і керуючих сигналів.

На відміну від структурної схеми, яка характеризує апаратний склад системи та взаємозв'язки між технічними засобами, функціональна структура описує логіку роботи системи з погляду виконуваних функцій. Вона визначає, які програмні модулі забезпечують зчитування даних із датчиків, оброблення сигналів, визначення режиму роботи, прийняття рішень, керування виконавчими пристроями, передавання повідомлень користувачу та збереження інформації про події.

Автоматизована система безпеки будинку функціонує як єдиний програмно-апаратний комплекс, у якому кожен функціональний блок має визначене призначення. Сигнали від датчиків надходять до модуля зчитування даних, після чого передаються до модуля попереднього оброблення. Далі система визначає поточний режим роботи, аналізує отримані значення та встановлює, чи є вони ознакою тривожної або аварійної події. У разі підтвердження небезпеки активуються виконавчі пристрої, формується повідомлення для користувача та виконується запис інформації до журналу подій.

Функціональна структура автоматизованої системи управління безпекою будинку наведена на рисунку 3.1.

Функціональна структура автоматизованої системи управління безпекою будинку

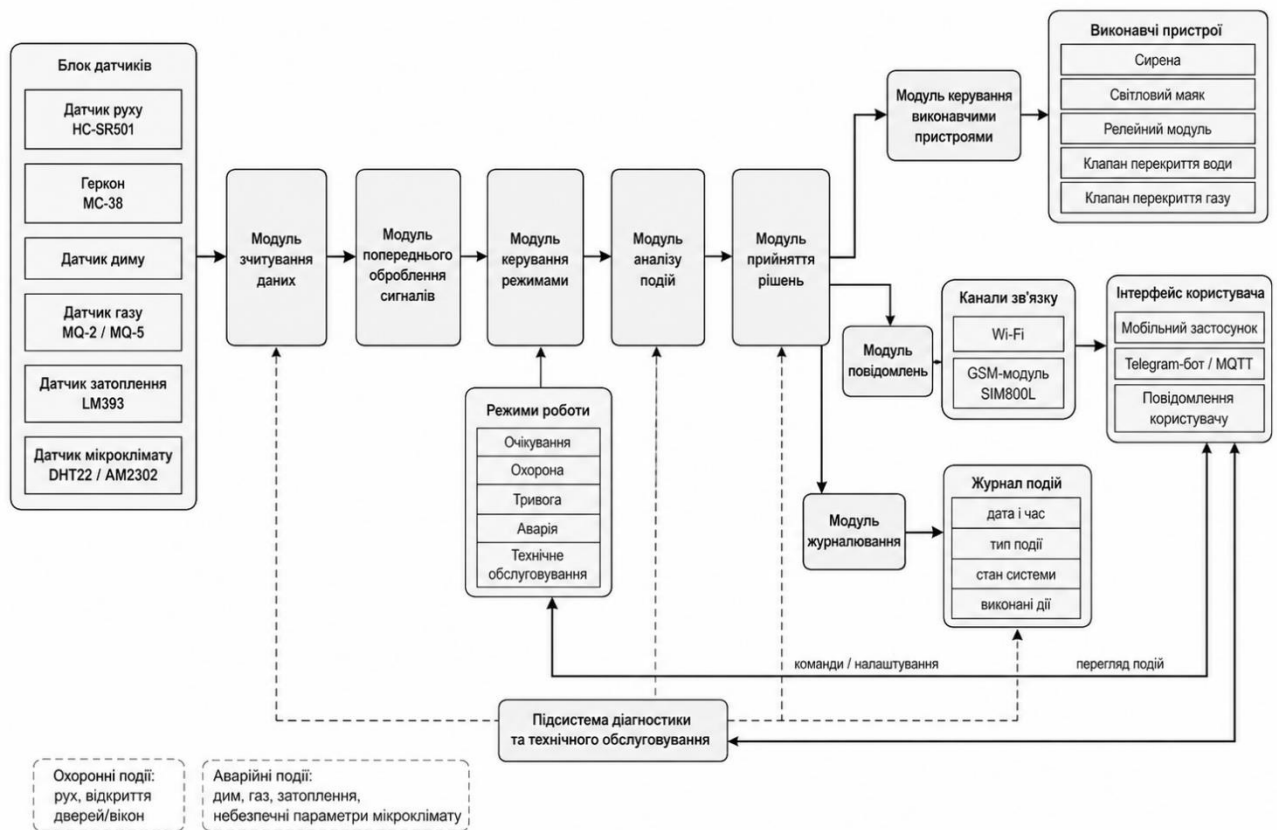


Рисунок 3.1 – Функціональна структура автоматизованої системи управління безпекою будинку

На рисунку 3.1 відображено основні функціональні блоки системи: блок датчиків, модуль зчитування даних, модуль попереднього оброблення сигналів, модуль керування режимами, модуль аналізу подій, модуль прийняття рішень,

модуль керування виконавчими пристроями, модуль повідомлень, модуль журналювання та інтерфейс користувача.

У запропонованій функціональній структурі виділено такі основні підсистеми:

- підсистема збору даних;
- підсистема попереднього оброблення сигналів;
- підсистема керування режимами роботи;
- підсистема аналізу подій і прийняття рішень;
- підсистема керування виконавчими пристроями;
- підсистема повідомлень користувача;
- підсистема журналювання подій;
- підсистема діагностики та технічного обслуговування.

Підсистема збору даних забезпечує отримання інформації від усіх підключених датчиків. До її складу входять програмні процедури зчитування цифрових, аналогових і цифрових шинних сигналів. Цифрові сигнали надходять від датчика руху, магнітоконтального датчика, датчика диму та датчика затоплення. Аналогові сигнали надходять від датчика газу. Дані температури та вологості отримуються від цифрового датчика мікроклімату.

Основним завданням підсистеми збору даних є своєчасне отримання інформації про стан контрольованого об'єкта. Система забезпечує періодичне опитування датчиків і передавання отриманих значень до наступного функціонального блоку. Для датчиків, які формують сигнал тривоги, забезпечується швидке реагування, оскільки затримка в обробленні може знизити ефективність системи безпеки [24].

Підсистема попереднього оброблення сигналів призначена для підготовки отриманих даних до подальшого аналізу. На цьому етапі виконується перевірка коректності сигналів, усунення короточасних нестабільних змін, фільтрація дрижання контактів і порівняння аналогових значень із пороговими величинами. Така обробка зменшує кількість хибних спрацювань і підвищує достовірність роботи системи.

Для герконового датчика відкриття дверей або вікна застосовується програмна затримка, яка дозволяє переконатися, що зміна стану контакту є стабільною. Для датчика газу значення з аналогового входу порівнюється з установленим порогом небезпеки. Для датчика затоплення виконується повторна перевірка сигналу через короткий проміжок часу, що дозволяє виключити випадкове короткочасне замикання контактів.

Підсистема керування режимами роботи визначає, у якому стані перебуває система в конкретний момент часу. В автоматизованій системі безпеки будинку передбачено кілька основних режимів: режим очікування, режим охорони, режим тривоги, аварійний режим і режим технічного обслуговування.

У режимі очікування система контролює аварійні параметри, але не реагує на рух у приміщенні як на охоронну тривогу. У режимі охорони активується контроль проникнення, тому спрацювання датчика руху або відкриття дверей розглядається як потенційна загроза. У режимі тривоги система активує сирену, світловий маяк і надсилає повідомлення користувачу. Аварійний режим вмикається у випадку виявлення диму, газу або затоплення. Режим технічного обслуговування використовується для перевірки датчиків, виконавчих пристроїв і каналів зв'язку.

Керування режимами є важливою частиною функціональної структури, оскільки одна й та сама подія має різне значення залежно від стану системи. Рух у приміщенні в режимі очікування не є тривогою, а рух у режимі охорони є ознакою можливого проникнення. Водночас аварійні події, такі як дим, газ або затоплення, обробляються незалежно від режиму охорони [12].

Підсистема аналізу подій і прийняття рішень є центральним логічним блоком системи. Вона отримує оброблені дані від датчиків і визначає, чи відповідають вони умовам тривоги або аварії. У цьому блоці реалізуються основні правила роботи автоматизованої системи безпеки.

Для охоронних подій умови формуються таким чином: якщо система перебуває в режимі охорони і датчик руху або датчик відкриття дверей спрацював, формується охоронна тривога. Для аварійних подій умови не залежать від режиму

охорони. Якщо датчик диму, газу або затоплення фіксує небезпечний стан, система переходить в аварійний режим.

Рішення системи залежить від типу події. Якщо небезпеку не виявлено, система продовжує моніторинг. Якщо зафіксовано охоронну тривогу, активуються сирена, світловий маяк і повідомлення користувачу. Якщо зафіксовано затоплення, система формує команду на перекриття води. Якщо виявлено витік газу, система формує термінове повідомлення, активує сигналізацію і подає команду на перекриття газу за наявності відповідного виконавчого пристрою.

Підсистема керування виконавчими пристроями реалізує фізичну реакцію системи на прийняте рішення. До виконавчих пристроїв належать сирена, світловий маяк, релейний модуль і електроприводні клапани. Керування ними здійснюється за допомогою виходів мікроконтролера ESP32 через проміжні комутаційні елементи.

У разі охоронної тривоги підсистема керування виконавчими пристроями вмикає сирену та світлову індикацію. У разі затоплення активується клапан перекриття води. У разі витіку газу активується клапан перекриття газу. Такий підхід забезпечує не лише інформування користувача, а й виконання практичних дій для зменшення наслідків небезпечної події.

Підсистема повідомлень користувача призначена для інформування власника будинку про стан системи та виникнення тривожних подій. Повідомлення є короткими, зрозумілими та містять інформацію про тип події. Наприклад: «Виявлено рух у приміщенні», «Відкрито двері», «Виявлено дим», «Перевищено рівень газу», «Виявлено затоплення».

Основним каналом передавання повідомлень є Wi-Fi. Через цей канал система взаємодіє з мобільним застосунком, Telegram-ботом, MQTT-брокером або іншим сервісом повідомлень. Для підвищення надійності використовується резервний GSM-канал, який забезпечує надсилання SMS-повідомлення або виконання тривожного дзвінка у разі втрати Wi-Fi-з'єднання.

Підсистема журналювання подій призначена для збереження інформації про важливі події, які відбуваються в системі. До журналу записуються дата і час події,

тип датчика, характер спрацювання, поточний режим системи, виконані дії та результат надсилання повідомлення. Наявність журналу дозволяє аналізувати роботу системи, перевіряти правильність її функціонування та виявляти повторювані спрацювання.

У простій реалізації журнал подій зберігається у вигляді масиву структур у пам'яті мікроконтролера або передається до зовнішнього сервісу. Такий підхід відповідає загальній логіці роботи системи і не потребує складної бази даних на етапі базової реалізації.

Підсистема діагностики та технічного обслуговування використовується для перевірки працездатності основних компонентів системи. Вона забезпечує тестове зчитування датчиків, перевірку стану виконавчих пристроїв, контроль наявності зв'язку та перевірку правильності налаштувань. Такий режим використовується під час налагодження системи та її подальшого обслуговування.

Основні функціональні блоки системи та їхнє призначення наведено в таблиці 3.3.

Таблиця 3.3 – Функціональні блоки автоматизованої системи безпеки будинку

Функціональний блок	Вхідні дані	Вихідні дані або дії	Призначення
Блок зчитування датчиків	Сигнали від датчиків руху, відкриття, диму, газу, затоплення, мікроклімату	Первинні значення параметрів	Отримання інформації про стан об'єкта
Блок попереднього оброблення	Первинні значення датчиків	Відфільтровані та перевірені дані	Усунення нестабільних сигналів і підготовка даних до аналізу
Блок керування режимами	Команди користувача, поточний стан системи	Активний режим роботи	Визначення поведінки системи залежно від режиму
Блок аналізу подій	Оброблені дані датчиків, активний режим	Рішення про наявність або відсутність загрози	Виявлення тривожних та аварійних подій
Блок прийняття рішень	Тип події, рівень небезпеки	Команди для реагування	Вибір дій системи відповідно до характеру події
Блок керування виконавчими пристроями	Команди від блоку прийняття рішень	Увімкнення сирени, маяка, реле, клапанів	Реалізація локальної реакції системи

Функціональний блок	Вхідні дані	Вихідні дані або дії	Призначення
Блок повідомлень	Інформація про подію	Повідомлення користувачу	Передавання інформації через Wi-Fi або GSM
Блок журналювання	Дані про подію та виконані дії	Запис у журналі подій	Збереження інформації для подальшого аналізу
Блок діагностики	Дані про стан компонентів	Повідомлення про справність або помилку	Перевірка працездатності системи

Функціональна структура системи передбачає послідовне проходження інформації через кілька етапів. Спочатку датчики формують сигнали про стан об'єкта. Потім ці сигнали зчитуються мікроконтролером і проходять попереднє оброблення. Після цього система визначає поточний режим роботи та аналізує, чи є отримані дані ознакою небезпеки. Якщо загроза підтверджена, формується рішення щодо подальших дій. Після прийняття рішення активуються виконавчі пристрої, користувачу надсилається повідомлення, а інформація про подію зберігається в журналі.

Типовий сценарій роботи системи в режимі охорони має таку послідовність. Датчик руху фіксує переміщення в приміщенні. Сигнал надходить до модуля зчитування даних, після чого проходить перевірку на достовірність. Модуль керування режимами підтверджує, що система перебуває в режимі охорони. Блок аналізу подій визначає, що така подія є тривожною. Блок прийняття рішень формує команду на активацію сирени та надсилання повідомлення. Після цього блок керування виконавчими пристроями вмикає сирену, блок повідомлень передає інформацію користувачу, а блок журналювання зберігає запис про подію.

Аварійний сценарій роботи системи має іншу логіку. Якщо датчик затоплення фіксує появу води, система не перевіряє режим охорони, оскільки затоплення є аварійною ситуацією незалежно від поточного режиму. Після підтвердження сигналу система формує команду на перекриття води, надсилає повідомлення власнику та записує подію до журналу. Така логіка підвищує ефективність реагування на аварії та зменшує можливі збитки.

Функціональна структура також враховує можливі відмови окремих елементів. Якщо Wi-Fi-з'єднання недоступне, система переходить до використання

резервного GSM-каналу. Якщо повідомлення не вдалося надіслати через основний канал, виконується спроба передавання через альтернативний канал. Якщо один із датчиків не відповідає або передає некоректні дані, система формує службове повідомлення про несправність.

Важливим принципом побудови функціональної структури є модульність. Кожен функціональний блок виконує окрему задачу. Це спрощує розроблення програмного забезпечення, його тестування та подальше вдосконалення. У разі необхідності спосіб передавання повідомлень змінюється без зміни алгоритму зчитування датчиків, а новий датчик додається без повної перебудови логіки роботи системи.

Модульна організація також забезпечує можливість поступового розширення функціональних можливостей системи. У майбутньому до неї додається модуль відеоспостереження, модуль керування електрозамком, датчик розбиття скла або інтеграція з іншими елементами системи “розумного будинку”. При цьому базова логіка роботи залишається незмінною: збір даних, оброблення, аналіз, прийняття рішення, реагування та інформування користувача.

Отже, розроблена функціональна структури автоматизованої системи управління безпекою будинку визначає склад основних програмно-логічних блоків і порядок їхньої взаємодії. Запропонована структура забезпечує послідовне оброблення інформації від датчиків, прийняття рішень щодо тривожних і аварійних подій, керування виконавчими пристроями, повідомлення користувача та збереження інформації про події. Вона є основою для подальшого розроблення алгоритмів функціонування системи та її програмної реалізації.

3.3 Розроблення алгоритмів функціонування автоматизованої системи безпеки

Після визначення функціональних вимог і розроблення функціональної структури автоматизованої системи безпеки будинку необхідно розробити алгоритми її функціонування. Алгоритми визначають послідовність дій, які виконує система під час запуску, зчитування даних із датчиків, аналізу отриманої

інформації, виявлення тривожних або аварійних подій, активації виконавчих пристроїв, надсилання повідомлень користувачу та запису подій до журналу.

Алгоритмічне забезпечення є основою програмної реалізації системи, оскільки саме воно визначає логіку роботи мікроконтролера ESP32. У межах розроблюваної системи всі основні дії виконуються циклічно. Після подачі живлення система ініціалізує апаратні компоненти, перевіряє стан датчиків і переходить до режиму постійного моніторингу. У процесі моніторингу контролер отримує сигнали від сенсорів, аналізує їх відповідно до поточного режиму роботи та визначає, чи потрібно виконувати дії реагування [12].

Алгоритми роботи системи побудовані з урахуванням поділу подій на охоронні та аварійні. До охоронних подій належать рух у приміщенні та відкриття дверей або вікон у режимі охорони. До аварійних подій належать поява диму, витік газу, затоплення, а також небезпечне відхилення параметрів мікроклімату. Такий поділ є важливим, оскільки охоронні події повинні оброблятися лише тоді, коли система перебуває у режимі охорони, а аварійні події повинні оброблятися незалежно від режиму охорони.

Основними алгоритмами функціонування автоматизованої системи безпеки є:

- загальний алгоритм роботи системи;
- алгоритм оброблення охоронної події;
- алгоритм реагування на аварійну подію;
- алгоритм надсилання повідомлень користувачу;
- алгоритм журналювання подій.

Загальний алгоритм функціонування системи визначає послідовність дій від моменту ввімкнення живлення до переходу системи в режим постійного моніторингу. Він охоплює ініціалізацію мікроконтролера, налаштування входів і виходів, перевірку датчиків, підключення до каналів зв'язку та запуск основного циклу роботи.

Загальний алгоритм функціонування автоматизованої системи безпеки наведено на рисунку 3.2.

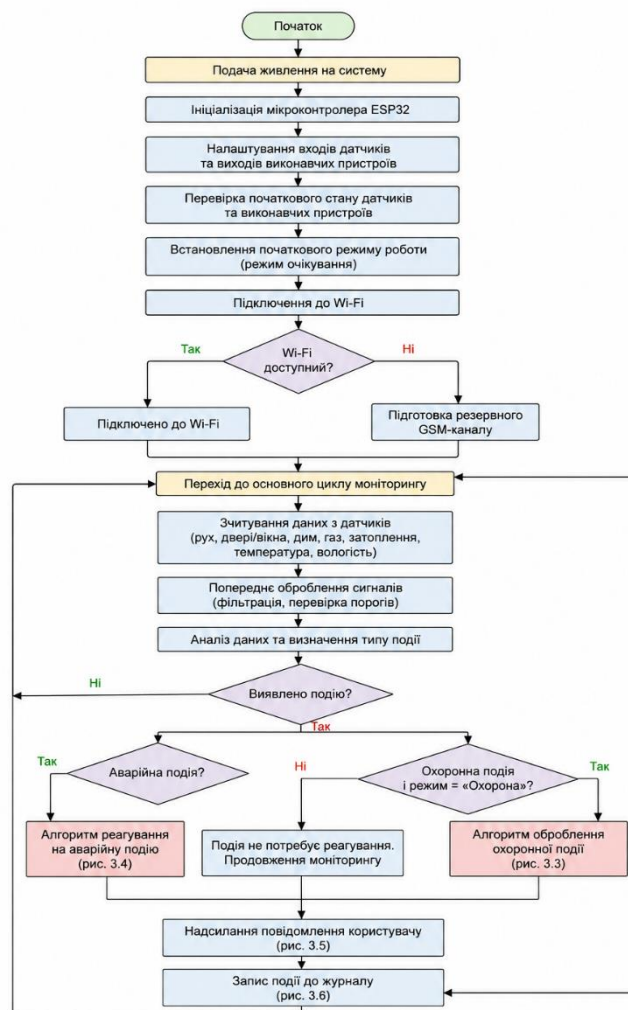


Рисунок 3.2 – Загальний алгоритм функціонування автоматизованої системи безпеки

На початковому етапі після подачі живлення мікроконтролер ESP32 виконує ініціалізацію системи. Під час ініціалізації задаються режими роботи портів GPIO, налаштовуються входи для датчиків і виходи для виконавчих пристроїв, встановлюються початкові стани релейних модулів, сирени, світлового маяка та електроприводних клапанів. На цьому ж етапі система встановлює початковий режим роботи, наприклад режим очікування.

Після апаратної ініціалізації виконується спроба підключення до каналу зв'язку. Основним каналом є Wi-Fi, який використовується для передавання повідомлень користувачу через мобільний застосунок, Telegram-бот або MQTT-брокер. Якщо Wi-Fi-з'єднання недоступне, система може використовувати резервний GSM-канал на базі модуля SIM800L. При цьому відсутність інтернет-

з'єднання не повинна зупиняти роботу системи, оскільки локальна логіка оброблення тривоги реалізується безпосередньо на мікроконтролері.

Після завершення ініціалізації система переходить до основного циклу моніторингу. У цьому циклі ESP32 періодично зчитує дані з датчиків руху, відкриття дверей або вікон, диму, газу, затоплення, температури та вологості. Для цифрових датчиків визначається логічний стан входу. Для аналогових датчиків виконується зчитування значення з АЦП і порівняння з установленим порогом. Для цифрового датчика мікроклімату отримуються значення температури та вологості.

Після зчитування даних система виконує попереднє оброблення сигналів. На цьому етапі усуваються короточасні нестабільні зміни сигналу, виконується фільтрація дрижання контактів, повторна перевірка окремих датчиків і порівняння отриманих значень із граничними параметрами. Це дозволяє зменшити ймовірність хибних спрацювань [17].

Далі система аналізує отримані дані та визначає тип події. Якщо подія не є небезпечною, система повертається до циклу моніторингу. Якщо зафіксовано охоронну або аварійну подію, система переходить до відповідного алгоритму реагування. Після виконання всіх дій реагування система знову повертається до основного циклу контролю.

Загальний порядок роботи системи можна подати у вигляді такої послідовності:

1. подача живлення на систему;
2. ініціалізація мікроконтролера ESP32;
3. налаштування входів датчиків і виходів виконавчих пристроїв;
4. перевірка початкового стану системи;
5. підключення до Wi-Fi або підготовка резервного GSM-каналу;
6. перехід до основного циклу моніторингу;
7. зчитування даних із датчиків;
8. попереднє оброблення сигналів;
9. перевірка наявності охоронної або аварійної події;
10. виконання відповідного алгоритму реагування;

11. надсилання повідомлення користувачу;
12. запис події до журналу;
13. повернення до циклу моніторингу.

Окремим важливим алгоритмом є алгоритм оброблення охоронної події. Він використовується тоді, коли система перебуває в режимі охорони і один із охоронних датчиків фіксує потенційне проникнення. До таких подій належать спрацювання датчика руху HC-SR501 або зміна стану магнітоконтактного датчика МС-38, який контролює відкриття дверей або вікна.

Алгоритм оброблення охоронної події наведено на рисунку 3.3.

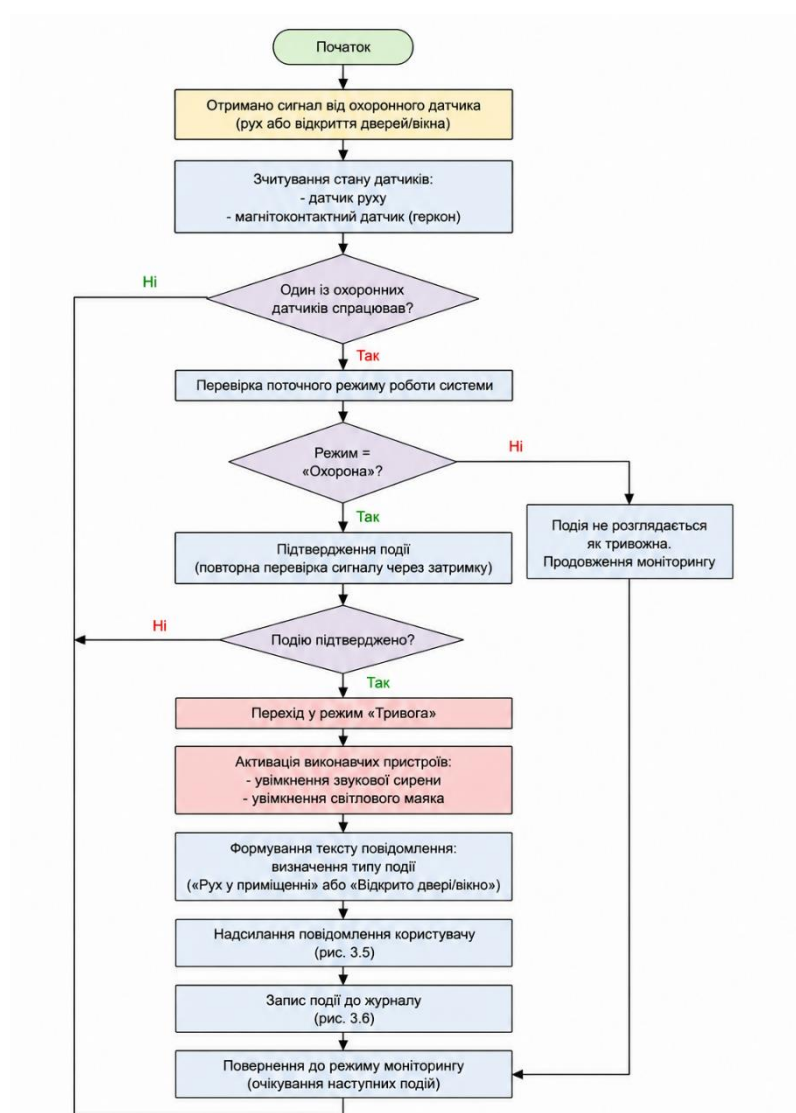


Рисунок 3.3 – Алгоритм оброблення охоронної події

Алгоритм починається зі зчитування стану охоронних датчиків. Якщо датчик руху не зафіксував переміщення, а герконовий датчик не зафіксував відкриття

дверей або вікна, система продовжує моніторинг. Якщо один із датчиків спрацював, система перевіряє поточний режим роботи.

Якщо система перебуває в режимі очікування або технічного обслуговування, спрацювання охоронного датчика не розглядається як тривога. Наприклад, рух у приміщенні в режимі очікування є нормальним станом, оскільки власник може перебувати вдома. Якщо ж система перебуває в режимі охорони, спрацювання датчика руху або відкриття дверей розглядається як потенційна тривожна подія.

Після цього виконується підтвердження події. Підтвердження потрібне для зменшення кількості хибних спрацювань. Для цього система може повторно зчитати сигнал через короткий проміжок часу або перевірити стабільність сигналу. Якщо подія не підтверджується, система повертається до режиму моніторингу. Якщо подія підтверджується, система переходить у режим тривоги [25].

У режимі тривоги активуються виконавчі пристрої. Насамперед вмикається звукова сирена та світловий маяк. Після цього формується повідомлення користувачу із зазначенням типу події. Наприклад, якщо спрацював датчик руху, формується повідомлення «Виявлено рух у приміщенні». Якщо спрацював геркон, формується повідомлення «Відкрито двері або вікно». Після надсилання повідомлення інформація про подію записується до журналу.

Алгоритм оброблення охоронної події можна подати у вигляді такої послідовності:

1. зчитати стан датчика руху та геркона;
2. перевірити, чи спрацював один із охоронних датчиків;
3. якщо спрацювання немає — повернутися до моніторингу;
4. якщо спрацювання є — перевірити режим роботи системи;
5. якщо режим не є режимом охорони — не формувати тривогу;
6. якщо система перебуває в режимі охорони — виконати підтвердження події;
7. якщо подія не підтвердилася — повернутися до моніторингу;
8. якщо подія підтвердилася — перейти в режим тривоги;

9. увімкнути сирену та світловий маяк;
10. сформувати повідомлення користувачу;
11. надіслати повідомлення через доступний канал зв'язку;
12. записати подію до журналу;
13. продовжити моніторинг.

Алгоритм реагування на аварійні події відрізняється від алгоритму охоронної тривоги тим, що він не залежить від режиму охорони. Поява диму, витік газу, затоплення або небезпечне підвищення температури становлять загрозу незалежно від того, перебуває будинок під охороною чи ні. Тому аварійні події повинні оброблятися у будь-якому режимі роботи системи.

Алгоритм реагування на аварійні події наведено на рисунку 3.4.

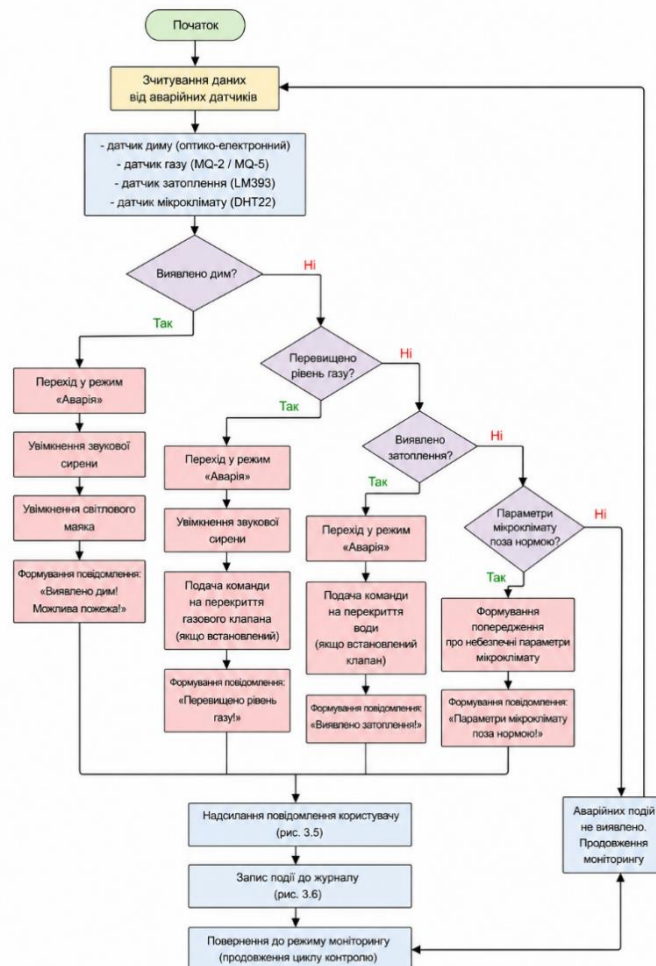


Рисунок 3.4 – Алгоритм реагування на аварійні події

На першому етапі система зчитує дані з аварійних датчиків. До них належать датчик диму, датчик газу MQ-2 або MQ-5, датчик затоплення LM393 і датчик

мікроклімату DHT22/AM2302. Для датчика диму та датчика затоплення перевіряється факт спрацювання. Для датчика газу виконується порівняння значення з установленим порогом. Для датчика мікроклімату аналізуються температура та вологість.

Якщо аварійних подій не виявлено, система продовжує моніторинг. Якщо один із аварійних параметрів перевищує допустиме значення або датчик фіксує небезпечний стан, система переходить в аварійний режим. У цьому режимі реакція системи залежить від типу події.

У разі виявлення диму система вмикає сирену, активує світловий маяк і формує повідомлення користувачу про пожежну небезпеку. У разі перевищення допустимого рівня газу система формує аварійне повідомлення, активує сигналізацію і подає команду на перекриття газового клапана, якщо такий виконавчий пристрій передбачений у системі. У разі затоплення система формує повідомлення про витік води та активує клапан перекриття води. Якщо зафіксовано небезпечні параметри мікроклімату, система формує попередження користувачу.

Після виконання дій реагування система записує інформацію про подію до журналу. У журналі зазначається тип аварійної події, час її виникнення, стан системи та виконані дії. Після цього система продовжує моніторинг, оскільки можливе виникнення нових подій або повторне спрацювання датчиків.

Алгоритм реагування на аварійні події можна подати у вигляді такої послідовності:

1. зчитати дані з датчиків диму, газу, затоплення та мікроклімату;
2. перевірити наявність аварійної події;
3. якщо аварії немає — повернутися до моніторингу;
4. якщо виявлено дим — увімкнути сирену, маяк і сформувати пожежне повідомлення;
5. якщо виявлено витік газу — увімкнути сигналізацію, сформувати повідомлення і подати команду на перекриття газу;
6. якщо виявлено затоплення — сформувати повідомлення і подати команду на перекриття води;

7. якщо виявлено небезпечні параметри мікроклімату — сформулювати попередження;
8. надіслати повідомлення користувачу;
9. записати подію до журналу;
10. повернутися до циклу моніторингу.

Для коректної роботи алгоритмів необхідно визначити умови переходу системи між режимами. Основні умови переходів наведено в таблиці 3.4.

Таблиця 3.4 – Умови переходу між режимами роботи системи

Поточний режим	Умова переходу	Новий режим	Дії системи
Очікування	Команда користувача на постановку під охорону	Охорона	Перевірка датчиків і активація режиму охорони
Охорона	Спрацювання датчика руху або геркона	Тривога	Увімкнення сирени, маяка, повідомлення користувача
Будь-який режим	Виявлено дим	Аварія	Сирена, маяк, повідомлення про пожежну небезпеку
Будь-який режим	Перевищено рівень газу	Аварія	Сирена, повідомлення, перекриття газу
Будь-який режим	Виявлено затоплення	Аварія	Повідомлення, перекриття води
Будь-який режим	Команда користувача на тестування	Технічне обслуговування	Перевірка датчиків, виконавчих пристроїв і зв'язку
Тривога або аварія	Подія оброблена, користувач отримав повідомлення	Моніторинг очікування	Запис події до журналу та повернення до контролю

Окремим елементом алгоритмічного забезпечення є алгоритм надсилання повідомлень користувачу. Його завдання полягає у виборі доступного каналу зв'язку та передачі інформації про подію. Система спочатку перевіряє доступність Wi-Fi-з'єднання. Якщо основний канал доступний, повідомлення передається через нього. Якщо Wi-Fi недоступний, використовується GSM-модуль SIM800L.

Алгоритм надсилання повідомлення має таку послідовність:

1. сформулювати текст повідомлення;
2. перевірити доступність Wi-Fi;

3. якщо Wi-Fi доступний — передати повідомлення через основний канал;
4. якщо Wi-Fi недоступний — активувати GSM-канал;
5. надіслати SMS-повідомлення або виконати тривожний дзвінок;
6. зафіксувати результат передавання в журналі подій.

Алгоритм журналювання подій забезпечує збереження інформації про роботу системи. Журнал подій використовується для подальшого аналізу, перевірки коректності роботи алгоритмів і виявлення повторюваних спрацювань. Для кожної події зберігаються дата і час, тип події, джерело сигналу, режим роботи системи, виконані дії та результат надсилання повідомлення.

Структура запису журналу подій наведена в таблиці 3.5.

Таблиця 3.5 – Структура запису журналу подій

Поле запису	Зміст
Дата і час	Момент виникнення події
Тип події	Охоронна, аварійна або службова подія
Джерело події	Назва датчика або програмного модуля
Режим системи	Очікування, охорона, тривога, аварія, технічне обслуговування
Виконані дії	Сирена, маяк, клапан, повідомлення користувачу
Канал повідомлення	Wi-Fi або GSM
Результат	Подію оброблено, повідомлення надіслано або виявлено помилку

Журналювання виконується після кожної важливої події. Якщо система виявила рух у режимі охорони, до журналу записується інформація про охоронну тривогу. Якщо зафіксовано затоплення, до журналу вноситься аварійна подія та дія з перекриття води. Якщо система не змогла надіслати повідомлення через Wi-Fi і використала GSM, цей факт також фіксується в журналі.

Узагальнено алгоритм журналювання можна подати таким чином:

Отримати дані про подію

Визначити тип події

Визначити джерело події

Записати поточний режим системи

Записати виконані дії

Записати канал повідомлення

Зберегти результат оброблення

Розроблені алгоритми забезпечують послідовну та зрозумілу логіку функціонування автоматизованої системи безпеки будинку. Система спочатку виконує зчитування та оброблення даних, потім аналізує події, приймає рішення, активує виконавчі пристрої, інформує користувача та зберігає інформацію про подію. Такий підхід забезпечує безперервний контроль стану будинку та дозволяє реалізувати як охоронні, так і аварійні функції.

Отже, у межах цього підрозділу розроблено основні алгоритми функціонування автоматизованої системи управління безпекою будинку. Сформовано загальний алгоритм роботи системи, алгоритм оброблення охоронної події, алгоритм реагування на аварійні події, алгоритм надсилення повідомлень і алгоритм журналювання. Розроблені алгоритми є основою для подальшої організації структури даних і програмної реалізації системи.

3.4 Організація та структура даних в автоматизованій системі безпеки

Після визначення функціональних вимог, розроблення функціональної структури та алгоритмів роботи автоматизованої системи безпеки необхідно сформувати організацію даних, які використовуються під час функціонування системи. Структура даних визначає, у якому вигляді програмне забезпечення зберігає інформацію про датчики, режими роботи, події, виконавчі пристрої, повідомлення користувачу та результати реагування системи.

Для автоматизованої системи управління безпекою будинку правильна організація даних має важливе значення, оскільки від неї залежить коректність аналізу подій, швидкість прийняття рішень і можливість подальшого розширення функціональних можливостей системи. Якщо дані від різних датчиків подаються у невпорядкованому вигляді, програмі складніше визначити тип події та виконати відповідну дію. Тому всі дані доцільно згрупувати за їхнім призначенням і використовувати єдині логічні структури [28].

У межах розроблюваної системи дані можна поділити на кілька основних груп: вхідні дані від датчиків, дані про поточний режим роботи, дані про події, дані

про виконавчі пристрої, дані для повідомлення користувача, дані журналу подій та конфігураційні параметри системи.

Основні групи даних автоматизованої системи безпеки наведено в таблиці 3.6.

Таблиця 3.6 – Основні групи даних автоматизованої системи безпеки будинку

Група даних	Приклади даних	Призначення
Дані від датчиків	Стан датчика руху, геркона, датчика диму, газу, затоплення, температура, вологість	Отримання інформації про стан контролюваного об'єкта
Дані про режим роботи	Очікування, охорона, тривога, аварія, технічне обслуговування	Визначення поведінки системи залежно від поточного стану
Дані про події	Тип події, джерело події, час виникнення, рівень небезпеки	Аналіз тривожних, аварійних і службових ситуацій
Дані про виконавчі пристрої	Стан сирени, світлового маяка, реле, клапана перекриття води або газу	Формування керуючих впливів на обладнання
Дані повідомлень	Текст повідомлення, канал передавання, результат надсилання	Інформування користувача про стан системи
Дані журналу	Запис про подію, виконані дії, результат оброблення	Збереження історії роботи системи
Конфігураційні дані	Порогові значення, налаштування каналів зв'язку, параметри режимів	Налаштування роботи системи відповідно до умов експлуатації

Вхідні дані формуються датчиками, які підключені до мікроконтролера ESP32. Частина датчиків передає цифровий сигнал, який може мати два логічні стани: активний або неактивний. До таких датчиків належать датчик руху, магнітоконтактний датчик відкриття дверей або вікна, датчик диму та датчик затоплення. Датчик газу формує аналоговий сигнал, який зчитується за допомогою аналогового входу мікроконтролера. Датчик мікроклімату передає значення температури та вологості у цифровому вигляді.

Для уніфікації роботи з різними типами датчиків у програмному забезпеченні доцільно використовувати єдину структуру опису датчика. Така структура повинна містити ідентифікатор датчика, його тип, поточне значення, порогове значення, стан спрацювання та час останнього зчитування. Це дозволяє обробляти дані від різних сенсорів за єдиною логікою.

Структуру даних датчика наведено в таблиці 3.7.

Таблиця 3.7 – Структура даних датчика автоматизованої системи безпеки

Поле структури	Зміст поля	Приклад значення
sensorId	Унікальний номер або назва датчика	PIR_01, DOOR_01, GAS_01
sensorType	Тип датчика	рух, відкриття, дим, газ, затоплення, температура
pin	Номер входу мікроконтролера	GPIO 14, GPIO 27, ADC 34
currentValue	Поточне зчитане значення	0, 1, 2150, 23.5
thresholdValue	Порогове значення для аналізу	1, 1800, 60
isActive	Ознака спрацювання датчика	true / false
lastReadTime	Час останнього зчитування	12:15:04
sensorStatus	Стан датчика	справний, несправний, немає сигналу

Для цифрових датчиків поле currentValue може містити логічне значення 0 або 1. Наприклад, якщо герконовий датчик перебуває у замкненому стані, система фіксує нормальний стан. Якщо контакт розімкнувся, це означає відкриття дверей або вікна. Для аналогового датчика газу поле currentValue містить числове значення, отримане з аналогового входу. Це значення порівнюється з установленим порогом небезпеки. Для датчика температури та вологості зберігаються числові значення відповідних параметрів.

Окрему групу становлять дані про поточний режим роботи системи. Режим визначає, як саме система повинна реагувати на отримані сигнали. Наприклад, спрацювання датчика руху в режимі очікування не є тривоною, а таке саме спрацювання в режимі охорони розглядається як можлива спроба проникнення. Водночас аварійні події, зокрема дим, витік газу або затоплення, повинні оброблятися незалежно від режиму охорони.

Для опису режимів роботи системи доцільно використовувати перелік фіксованих станів. У програмній реалізації такі стани можуть бути подані у вигляді числових кодів або текстових позначень.

Таблиця 3.8 – Структура даних для опису режимів роботи системи

Назва режиму	Умовне позначення	Призначення
Режим очікування	WAITING	Система контролює аварійні параметри, але не формує охоронну тривогу за датчиком руху

Назва режиму	Умовне позначення	Призначення
Режим охорони	ARMED	Активний контроль проникнення за датчиками руху та відкриття
Режим тривоги	ALARM	Активовано реакцію на охоронну подію
Аварійний режим	EMERGENCY	Виявлено дим, газ, затоплення або небезпечне відхилення параметрів
Режим технічного обслуговування	SERVICE	Виконується перевірка датчиків, виконавчих пристроїв і каналів зв'язку

Поточний режим роботи зберігається у змінній стану системи. Зміна цієї змінної відбувається після команди користувача або після виявлення відповідної події. Наприклад, після команди користувача на постановку будинку під охорону система переходить із режиму очікування в режим охорони. Якщо в цьому режимі спрацює датчик руху або відкриття дверей, система переходить у режим тривоги. Якщо в будь-якому режимі виявлено дим або витік газу, система переходить в аварійний режим.

Для збереження загального стану системи доцільно передбачити структуру, яка містить не лише поточний режим, а й інформацію про доступність каналів зв'язку, наявність активної події та стан основних виконавчих пристроїв.

Таблиця 3.9 – Структура даних поточного стану системи

Поле структури	Зміст поля	Приклад значення
currentMode	Поточний режим роботи системи	WAITING, ARMED, ALARM
activeEvent	Поточна активна подія	рух, дим, газ, затоплення
alarmState	Ознака активної тривоги	true / false
emergencyState	Ознака аварійної ситуації	true / false
wifiAvailable	Доступність Wi-Fi-з'єднання	true / false
gsmAvailable	Доступність GSM-каналу	true / false
sirenState	Стан сирени	увімкнено / вимкнено
lightState	Стан світлового маяка	увімкнено / вимкнено
valveWaterState	Стан клапана перекриття води	відкритий / закритий
valveGasState	Стан клапана перекриття газу	відкритий / закритий

Наявність такої структури спрощує роботу програмного забезпечення, оскільки всі основні параметри стану системи зберігаються в одному логічному блоці. Це дає змогу швидко перевірити, у якому режимі перебуває система, чи активна тривога, який канал зв'язку доступний і які виконавчі пристрої вже задіяні.

Дані про події є однією з найважливіших складових інформаційної моделі системи. Подією вважається будь-яка зміна стану, яка має значення для роботи системи. До подій належать спрацювання датчика руху, відкриття дверей або вікна, виявлення диму, перевищення рівня газу, затоплення, відхилення температури або вологості від установлених меж, втрата Wi-Fi-з'єднання, перехід на GSM-канал, помилка датчика або команда користувача.

Усі події доцільно поділяти за типом і рівнем пріоритету. Це дозволяє системі правильно визначати порядок реагування, особливо у випадках, коли одночасно виникає кілька подій. Наприклад, виявлення диму або витоку газу має вищий пріоритет, ніж службове повідомлення про втрату Wi-Fi-з'єднання.

Таблиця 3.10 – Класифікація подій автоматизованої системи безпеки

Тип події	Приклад події	Рівень пріоритету	Реакція системи
Аварійна подія	Дим, газ, затоплення	Високий	Перехід в аварійний режим, активація виконавчих пристроїв, повідомлення користувачу
Охоронна подія	Рух, відкриття дверей або вікна в режимі охорони	Високий	Перехід у режим тривоги, сирена, маяк, повідомлення
Попереджувальна подія	Відхилення температури або вологості від установлених меж	Середній	Формування попередження користувачу
Службова подія	Втрата Wi-Fi, перехід на GSM, несправність датчика	Середній	Запис до журналу, повідомлення користувачу за потреби
Інформаційна подія	Постановка під охорону, зняття з охорони, тестування	Низький	Запис до журналу, зміна режиму роботи

Для програмного представлення події доцільно використовувати структуру, яка містить тип події, джерело, час виникнення, поточний режим системи, рівень небезпеки, сформоване повідомлення та перелік дій, які потрібно виконати.

Таблиця 3.11 – Структура даних події

Поле структури	Зміст поля	Приклад значення
eventId	Унікальний номер події	00125
eventType	Тип події	охоронна, аварійна, службова
eventSource	Джерело події	PIR_01, GAS_01, FLOOD_01
eventTime	Час виникнення події	14:35:18
systemMode	Режим системи під час події	ARMED, WAITING, SERVICE

Поле структури	Зміст поля	Приклад значення
priority	Пріоритет події	високий, середній, низький
messageText	Текст повідомлення користувачу	«Виявлено затоплення»
actionList	Перелік дій реагування	сирена, повідомлення, перекриття води
processingResult	Результат оброблення	виконано, помилка, очікує

У разі виникнення події система створює відповідний запис, аналізує його та передає до блоку прийняття рішень. Якщо подія має високий пріоритет, вона обробляється першочергово. Якщо подія не потребує негайного реагування, вона може бути лише записана до журналу або передана користувачу як службове повідомлення.

Дані про виконавчі пристрої використовуються для формування фізичної реакції системи. До виконавчих пристроїв належать сирена, світловий маяк, релейний модуль, електроприводний клапан перекриття води, електроприводний клапан перекриття газу та інші пристрої, які можуть бути додані до системи. Для кожного виконавчого пристрою необхідно зберігати його назву, тип, номер виходу мікроконтролера, поточний стан і команду керування.

Таблиця 3.12 – Структура даних виконавчого пристрою

Поле структури	Зміст поля	Приклад значення
actuatorId	Унікальна назва виконавчого пристрою	SIREN_01, VALVE_WATER
actuatorType	Тип пристрою	сирена, маяк, реле, клапан
pin	Номер виходу мікроконтролера	GPIO 25
currentState	Поточний стан пристрою	увімкнено / вимкнено
controlCommand	Команда керування	ON, OFF, CLOSE, OPEN
relatedEvent	Подія, з якою пов'язана дія	охоронна тривога, затоплення, газ
lastActionTime	Час останньої зміни стану	14:36:02

Наприклад, у разі виявлення затоплення система формує подію типу EMERGENCY, визначає джерело FLOOD_01 і створює команду для виконавчого пристрою VALVE_WATER. Після виконання команди стан клапана змінюється на «закритий», а результат дії записується до журналу подій.

Окреме місце в організації даних займають повідомлення користувачу. Повідомлення повинні бути короткими, зрозумілими та містити інформацію про

тип події. Для кожного повідомлення зберігається текст, канал передавання, адресат, час формування та результат надсилання.

Таблиця 3.13 – Структура даних повідомлення користувачу

Поле структури	Зміст поля	Приклад значення
messageId	Номер повідомлення	MSG_001
messageText	Текст повідомлення	«Перевищено рівень газу»
eventType	Тип події, що спричинила повідомлення	аварійна
channel	Канал передавання	Wi-Fi, GSM
recipient	Отримувач повідомлення	мобільний застосунок, Telegram, номер телефону
sendTime	Час надсилання	14:36:10
sendResult	Результат надсилання	надіслано, помилка, повторити

Основним каналом передавання повідомлень є Wi-Fi. Якщо Wi-Fi-з'єднання недоступне, система використовує резервний GSM-канал для надсилання SMS-повідомлення або виконання тривожного дзвінка. Такий підхід забезпечує підвищення надійності інформування користувача у випадку втрати основного каналу зв'язку.

Журнал подій призначений для збереження інформації про важливі дії та стани системи. У журналі повинні фіксуватися охоронні, аварійні та службові події. До журналу записується не лише факт спрацювання датчика, а й режим системи, виконані дії та результат надсилання повідомлення. Це дозволяє аналізувати роботу системи, виявляти повторювані спрацювання та перевіряти правильність алгоритмів.

У простій реалізації журнал може бути організований як послідовність записів у пам'яті мікроконтролера або у вигляді текстових повідомлень, що передаються до зовнішнього сервісу. Для зменшення використання пам'яті доцільно застосовувати циклічний журнал, у якому після заповнення доступного обсягу пам'яті найстаріші записи замінюються новими.

Таблиця 3.14 – Приклад запису журналу подій

Поле запису	Приклад значення
Дата і час	15.05.2026 14:36:10
Тип події	Аварійна
Джерело події	Датчик затоплення FLOOD 01
Поточний режим	Очікування
Виконані дії	Перекриття води, повідомлення користувачу
Канал повідомлення	Wi-Fi

Поле запису	Приклад значення
Результат	Подію оброблено, повідомлення надіслано

Конфігураційні дані використовуються для налаштування роботи системи. До них належать порогові значення для датчика газу, допустимі межі температури та вологості, параметри підключення до мережі, номер телефону для резервного повідомлення, налаштування режимів роботи та інші параметри. Такі дані не повинні змінюватися під час кожного циклу роботи системи, тому їх доцільно зберігати окремо від оперативних даних.

Таблиця 3.15 – Конфігураційні параметри автоматизованої системи безпеки

Параметр	Призначення	Приклад значення
gasThreshold	Порогове значення для датчика газу	1800
tempMin	Мінімально допустима температура	5 °C
tempMax	Максимально допустима температура	45 °C
humidityMax	Максимально допустима вологість	80 %
alarmDelay	Затримка підтвердження охоронної події	3 с
wifiStatus	Ознака використання Wi-Fi-каналу	увімкнено
gsmStatus	Ознака використання GSM-каналу	увімкнено
userPhone	Номер телефону для резервного повідомлення	заданий користувачем
systemName	Назва або ідентифікатор системи	ASUBB_01

Оперативні дані, які часто змінюються під час роботи програми, зберігаються в оперативній пам'яті мікроконтролера. До них належать поточні значення датчиків, режим роботи, активні події та поточні стани виконавчих пристроїв. Конфігураційні параметри можуть зберігатися в енергонезалежній пам'яті або задаватися під час початкового налаштування системи. Журнал подій може зберігатися локально або передаватися до зовнішнього сервісу, залежно від обраного способу реалізації.

Для передавання інформації через мережу доцільно формувати повідомлення у структурованому вигляді. Це спрощує подальше оброблення даних мобільним застосунком, Telegram-ботом, MQTT-брокером або іншим сервісом. Один із

можливих форматів передавання повідомлення може бути поданий у вигляді JSON-структури.

Приклад структури повідомлення про аварійну подію:

```
{  
  "device_id": "ASUBB_01",  
  "event_type": "emergency",  
  "source": "FLOOD_01",  
  "mode": "WAITING",  
  "message": "Виявлено затоплення",  
  "actions": ["close_water_valve", "send_notification"],  
  "channel": "Wi-Fi",  
  "result": "processed"  
}
```

У наведеному прикладі повідомлення містить ідентифікатор системи, тип події, джерело події, поточний режим роботи, текст повідомлення, виконані дії, канал передавання та результат оброблення. Такий формат є зрозумілим і може бути використаний для передавання даних між мікроконтролером та зовнішнім програмним сервісом.

Узагальнену інформаційну модель автоматизованої системи безпеки можна подати як взаємодію кількох основних структур: *SensorData*, *SystemState*, *EventData*, *ActuatorData*, *MessageData*, *LogRecord* і *ConfigData*. Дані від датчиків надходять до структури *SensorData*, після чого на їхній основі формується подія *EventData*. Поточний режим і загальний стан системи зберігаються у структурі *SystemState*. Якщо подія потребує реагування, формується команда для структури *ActuatorData* та повідомлення *MessageData*. Після завершення оброблення створюється запис *LogRecord*.

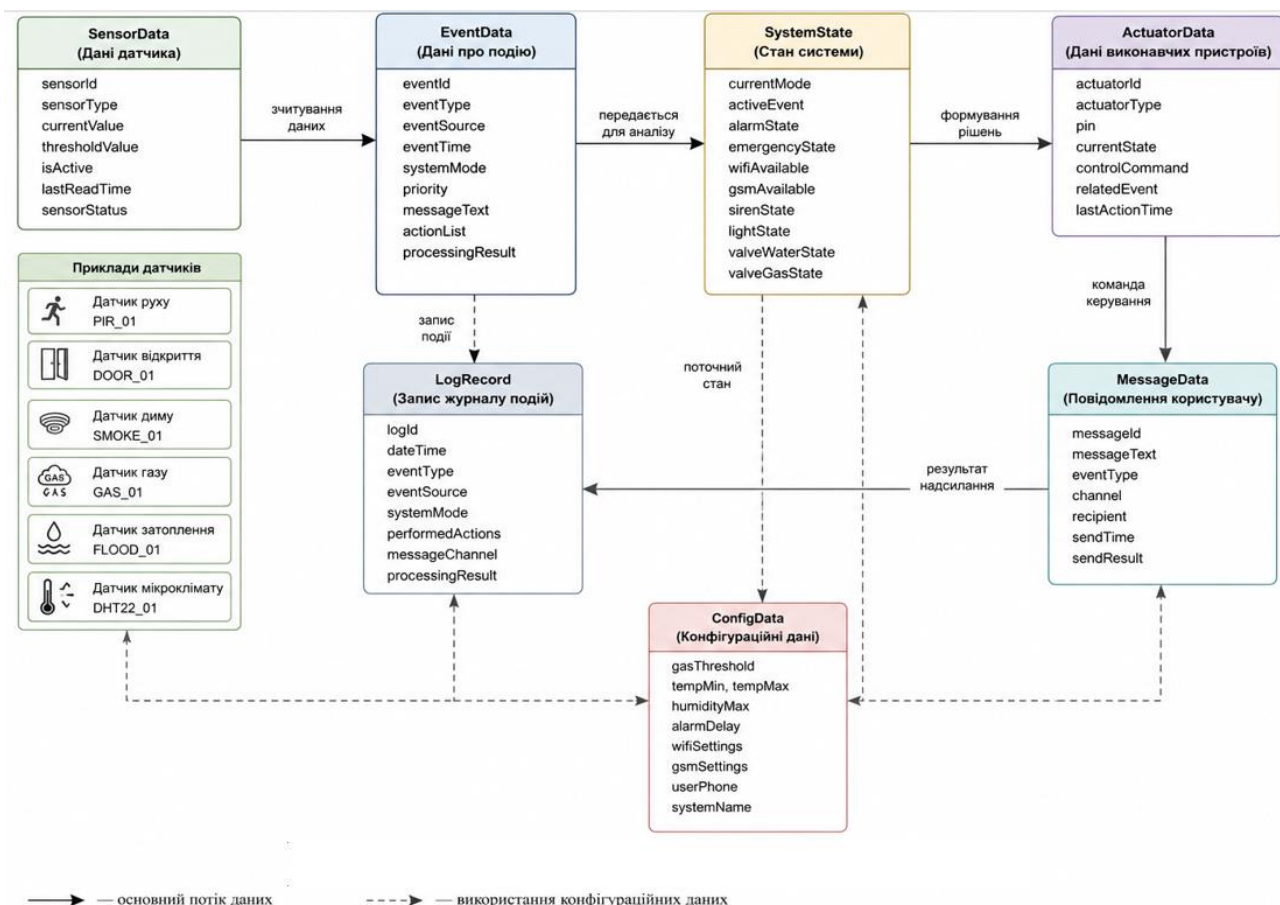


Рисунок 3.5 – Інформаційна модель автоматизованої системи безпеки будинку

Логічний зв'язок між основними структурами даних має такий вигляд:

1. датчик формує первинне значення;
2. дані зберігаються у структурі датчика;
3. система порівнює значення з умовами тривоги або аварії;
4. у разі виявлення події створюється запис події;
5. подія передається до блоку прийняття рішень;
6. формується команда для виконавчого пристрою;
7. створюється повідомлення користувачу;
8. результат оброблення записується до журналу.

Такий підхід дозволяє впорядкувати всі інформаційні потоки автоматизованої системи безпеки. Кожна структура даних має чітке призначення, а взаємодія між ними відповідає алгоритмам роботи системи, розробленим у попередньому підрозділі.

Важливою перевагою запропонованої організації даних є можливість подальшого розширення системи. У разі додавання нового датчика достатньо

створити новий елемент структури `SensorData` з відповідним типом і параметрами. Якщо додається новий виконавчий пристрій, він описується у структурі `ActuatorData`. Якщо змінюється спосіб повідомлення користувача, достатньо змінити параметри структури `MessageData`, не змінюючи загальну логіку зчитування датчиків та аналізу подій.

Отже, у межах цього підрозділу визначено організацію та структуру даних автоматизованої системи управління безпекою будинку. Дані системи поділено на групи відповідно до їхнього призначення: дані від датчиків, дані про режими роботи, події, виконавчі пристрої, повідомлення користувачу, журнал подій і конфігураційні параметри. Запропонована структура даних забезпечує впорядковане зберігання інформації, спрощує реалізацію алгоритмів роботи системи та створює основу для подальшого розроблення програмного забезпечення автоматизованої системи безпеки.

3.5 Розроблення програмного забезпечення автоматизованої системи управління безпекою будинку

Після визначення функціональних вимог, розроблення функціональної структури, алгоритмів функціонування та структури даних автоматизованої системи безпеки будинку виконано розроблення програмного забезпечення системи. Програмне забезпечення призначене для реалізації основних функцій системи: зчитування сигналів із датчиків, аналізу отриманих даних, визначення типу події, керування режимами роботи, формування команд для виконавчих пристроїв, надсилання повідомлень користувачу та запису інформації про події.

Програмна реалізація системи виконується на базі мікроконтролера ESP32, який має достатню кількість цифрових та аналогових входів/виходів, підтримує бездротове з'єднання Wi-Fi та може взаємодіяти із зовнішніми модулями, зокрема GSM-модулем для резервного каналу зв'язку. Використання ESP32 дозволяє поєднати локальну логіку реагування на події з можливістю дистанційного інформування користувача [25].

Програмне забезпечення розробляється мовою C/C++ у середовищі Arduino IDE або PlatformIO. Такий підхід є доцільним для мікроконтролерних систем,

оскільки забезпечує доступ до готових бібліотек для роботи з датчиками, Wi-Fi-з'єднанням, Telegram-ботом, MQTT-сервісом або GSM-модулем. При цьому основна логіка роботи системи реалізується безпосередньо в коді мікроконтролера, тому система здатна виконувати базові функції безпеки навіть у разі тимчасової втрати інтернет-з'єднання.

Програмне забезпечення автоматизованої системи управління безпекою будинку побудоване за модульним принципом. Кожен програмний модуль виконує окрему функцію, що спрощує розроблення, налагодження та подальше розширення системи. Структура програмного забезпечення показана рисунку 3.6

Основні програмні модулі наведено в таблиці 3.16.

Таблиця 3.16 – Основні програмні модулі автоматизованої системи безпеки будинку

Програмний модуль	Призначення	Основні функції
Модуль ініціалізації	Початкове налаштування системи	Налаштування входів і виходів, встановлення початкового режиму, перевірка датчиків
Модуль зчитування датчиків	Отримання даних від сенсорів	Зчитування цифрових, аналогових і цифрових сенсорних сигналів
Модуль попереднього оброблення	Підготовка даних до аналізу	Фільтрація нестабільних сигналів, повторна перевірка, порівняння з порогами
Модуль керування режимами	Визначення поточного стану системи	Перехід між режимами очікування, охорони, тривоги, аварії та обслуговування
Модуль аналізу подій	Виявлення охоронних та аварійних подій	Перевірка умов тривоги, визначення типу події та її пріоритету
Модуль керування виконавчими пристроями	Реалізація фізичної реакції системи	Увімкнення сирени, маяка, реле, перекриття води або газу
Модуль повідомлень	Інформування користувача	Передавання повідомлень через Wi-Fi або GSM
Модуль журналювання	Збереження інформації про події	Запис типу події, часу, дій системи та результату оброблення
Модуль діагностики	Перевірка працездатності компонентів	Тестування датчиків, виконавчих пристроїв і каналів зв'язку

На початку роботи програма виконує ініціалізацію системи. У цьому блоці задаються режими роботи входів і виходів мікроконтролера, встановлюються

початкові стани виконавчих пристроїв, виконується запуск послідовного порту для налагодження, перевіряється доступність датчиків і виконується спроба підключення до Wi-Fi. Якщо основний канал зв'язку недоступний, система готує до використання резервний GSM-канал.

Загальна структура програмного забезпечення може бути подана у вигляді трьох основних частин: блоку опису змінних і структур даних, функції `setup()` та основного циклу `loop()`. У блоці опису змінних задаються піни підключення датчиків і виконавчих пристроїв, порогові значення, режими роботи та структури для збереження подій. У функції `setup()` виконується початкове налаштування. В основному циклі `loop()` реалізується постійний моніторинг стану об'єкта.

Основні етапи роботи програми наведено на рисунку 3.6.

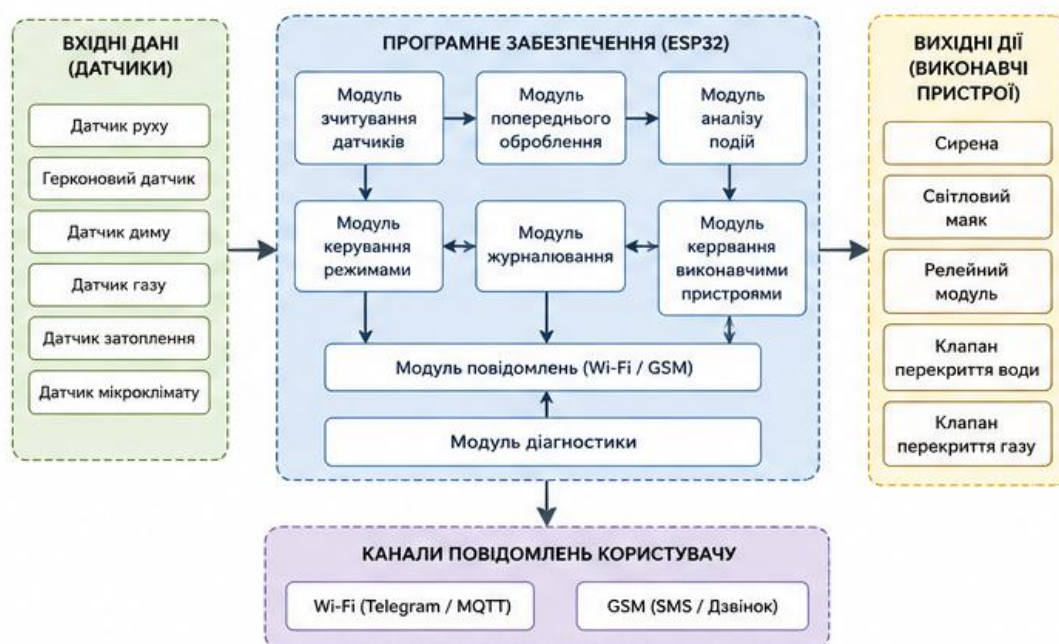


Рисунок 3.6 Структура програмного забезпечення автоматизованої системи безпеки будинку

У програмі передбачено такі основні режими роботи: очікування, охорона, тривога, аварія та технічне обслуговування. Для їхнього зручного використання доцільно подати режими у вигляді переліку.

```

enum SystemMode {
    MODE_WAITING,
    MODE_ARMED,

```

```

MODE_ALARM,
MODE_EMERGENCY,
MODE_SERVICE
};

SystemMode currentMode = MODE_WAITING;

```

Такий запис дозволяє уникнути використання випадкових числових значень у коді та робить програму більш зрозумілою. Наприклад, умова `currentMode == MODE_ARMED` одразу показує, що система перебуває в режимі охорони.

Для опису подій у програмі також використовується перелік типів подій. Це дозволяє розділити охоронні, аварійні, службові та інформаційні події.

```

enum EventType {
    EVENT_NONE,
    EVENT_MOTION,
    EVENT_DOOR_OPEN,
    EVENT_SMOKE,
    EVENT_GAS,
    EVENT_FLOOD,
    EVENT_MICROCLIMATE,
    EVENT_WIFI_LOST,
    EVENT_SENSOR_ERROR
};

```

Основні датчики та виконавчі пристрої підключаються до визначених входів і виходів мікроконтролера. Призначення входів і виходів у базовій реалізації наведено в таблиці 3.17.

Таблиця 3.17 – Призначення входів і виходів мікроконтролера ESP32

Елемент системи	Тип сигналу	Призначення в програмі
Датчик руху HC-SR501	Цифровий вхід	Виявлення руху в приміщенні
Герконовий датчик MC-38	Цифровий вхід	Контроль відкриття дверей або вікна
Датчик диму	Цифровий вхід	Виявлення ознак задимлення
Датчик газу MQ-2 / MQ-5	Аналоговий вхід	Контроль рівня газу
Датчик затоплення LM393	Цифровий вхід	Виявлення появи води

Елемент системи	Тип сигналу	Призначення в програмі
Датчик DHT22 / AM2302	Цифровий вхід	Вимірювання температури та вологості
Сирена	Цифровий вихід	Звукове оповіщення
Світловий маяк	Цифровий вихід	Світлова індикація тривоги
Релейний модуль	Цифровий вихід	Керування виконавчими пристроями
Клапан перекриття води	Цифровий вихід через реле	Перекриття води у разі затоплення
Клапан перекриття газу	Цифровий вихід через реле	Перекриття газу у разі небезпечного рівня газу

Під час ініціалізації програми для входів датчиків задається режим INPUT або INPUT_PULLUP, залежно від особливостей підключення. Для виконавчих пристроїв задається режим OUTPUT, а їхній початковий стан встановлюється як вимкнений. Це важливо для безпечного запуску системи, оскільки після подачі живлення сирена, реле або клапани не повинні вмикатися неконтрольовано.

Приклад фрагмента ініціалізації входів і виходів:

```
void setupPins() {
    pinMode(PIR_PIN, INPUT);
    pinMode(DOOR_PIN, INPUT_PULLUP);
    pinMode(SMOKE_PIN, INPUT);
    pinMode(FLOOD_PIN, INPUT);
    pinMode(GAS_PIN, INPUT);
    pinMode(SIREN_PIN, OUTPUT);
    pinMode(LIGHT_PIN, OUTPUT);
    pinMode(WATER_VALVE_PIN, OUTPUT);
    pinMode(GAS_VALVE_PIN, OUTPUT);
    digitalWrite(SIREN_PIN, LOW);
    digitalWrite(LIGHT_PIN, LOW);
    digitalWrite(WATER_VALVE_PIN, LOW);
    digitalWrite(GAS_VALVE_PIN, LOW);
}
```

Основна функція setup() виконує запуск усіх базових модулів системи. У ній налаштовуються піни, запускається послідовний порт, ініціалізується датчик

мікроклімату, виконується спроба підключення до Wi-Fi та встановлюється початковий режим роботи.

```
void setup() {
    Serial.begin(115200);
    setupPins();
    initSensors();
    initWiFi();
    if (!wifiAvailable) {
        initGSM();
    }
    currentMode = MODE_WAITING;
    Serial.println("System started in waiting mode");
}
```

Після виконання функції `setup()` програма переходить до основного циклу `loop()`. У цьому циклі система постійно зчитує дані з датчиків, перевіряє їх, визначає тип події та виконує необхідні дії реагування.

```
void loop() {
    readSensors();
    processSensorData();
    EventType event = detectEvent();
    if (event != EVENT_NONE) {
        handleEvent(event);
    }
    delay(500);
}
```

Функція `readSensors()` відповідає за отримання поточних значень від усіх підключених датчиків. Для цифрових датчиків використовується функція `digitalRead()`, для аналогового датчика газу — `analogRead()`, для датчика мікроклімату — відповідні функції бібліотеки DHT.

```
void readSensors() {
```

```

motionValue = digitalRead(PIR_PIN);
doorValue = digitalRead(DOOR_PIN);
smokeValue = digitalRead(SMOKE_PIN);
floodValue = digitalRead(FLOOD_PIN);
gasValue = analogRead(GAS_PIN);
temperature = dht.readTemperature();
humidity = dht.readHumidity();
}

```

Після зчитування значень виконується попереднє оброблення даних. Для цифрових датчиків це може бути повторна перевірка сигналу через короткий проміжок часу, а для аналогового датчика газу — порівняння з пороговим значенням. Такий підхід дозволяє зменшити кількість хибних спрацювань.

```

bool confirmDigitalSignal(int pin) {
    int firstRead = digitalRead(pin);
    delay(100);
    int secondRead = digitalRead(pin);
    return firstRead == secondRead;
}

```

Функція `detectEvent()` визначає, чи є поточні значення датчиків ознакою охоронної або аварійної події. Аварійні події мають вищий пріоритет і перевіряються першими, оскільки дим, газ або затоплення становлять небезпеку незалежно від режиму охорони.

```

EventType detectEvent() {
    if (smokeValue == HIGH) {
        return EVENT_SMOKE;
    }
    if (gasValue > gasThreshold) {
        return EVENT_GAS;
    }
    if (floodValue == HIGH) {

```

```

    return EVENT_FLOOD;
}
if (temperature > tempMax || humidity > humidityMax) {
    return EVENT_MICROCLIMATE;
}
if (currentMode == MODE_ARMED) {
    if (motionValue == HIGH && confirmDigitalSignal(PIR_PIN)) {
        return EVENT_MOTION;
    }
    if (doorValue == LOW && confirmDigitalSignal(DOOR_PIN)) {
        return EVENT_DOOR_OPEN;
    }
}
return EVENT_NONE;
}

```

У наведеному фрагменті спочатку перевіряються аварійні події: дим, перевищення рівня газу, затоплення та відхилення параметрів мікроклімату. Лише після цього перевіряються охоронні події, причому тільки в тому випадку, якщо система перебуває в режимі охорони. Така послідовність відповідає алгоритмам, розробленим у попередньому підрозділі.

Функція `handleEvent()` виконує реагування на виявлену подію. Залежно від типу події система змінює режим роботи, активує виконавчі пристрої, формує повідомлення користувачу та записує подію до журналу.

```

void handleEvent(EventType event) {
    String message = "";
    switch (event) {
        case EVENT_MOTION:
            currentMode = MODE_ALARM;
            activateAlarmDevices();
            message = "Виявлено рух у приміщенні";

```

```
    break;
case EVENT_DOOR_OPEN:
    currentMode = MODE_ALARM;
    activateAlarmDevices();
    message = "Відкрито двері або вікно";
    break;
case EVENT_SMOKE:
    currentMode = MODE_EMERGENCY;
    activateAlarmDevices();
    message = "Виявлено дим. Можлива пожежа";
    break;
case EVENT_GAS:
    currentMode = MODE_EMERGENCY;
    activateAlarmDevices();
    closeGasValve();
    message = "Перевищено рівень газу";
    break;
case EVENT_FLOOD:
    currentMode = MODE_EMERGENCY;
    closeWaterValve();
    message = "Виявлено затоплення";
    break;
case EVENT_MICROCLIMATE:
    message = "Параметри мікроклімату поза нормою";
    break;
default:
    return;
}

sendNotification(message);
writeLog(event, message);
```

```
}
```

Для охоронної тривоги система активує сирену та світловий маяк. Для аварійних подій реакція залежить від характеру небезпеки. У разі витоку газу подається команда на перекриття газу, а у разі затоплення — команда на перекриття води. Якщо зафіксовано відхилення температури або вологості від установлених меж, система формує попереджувальне повідомлення без обов’язкового вмикання сирени.

Функції керування виконавчими пристроями мають просту структуру. Вони змінюють стан відповідних виходів мікроконтролера.

```
void activateAlarmDevices() {
    digitalWrite(SIREN_PIN, HIGH);
    digitalWrite(LIGHT_PIN, HIGH);
}

void deactivateAlarmDevices() {
    digitalWrite(SIREN_PIN, LOW);
    digitalWrite(LIGHT_PIN, LOW);
}

void closeWaterValve() {
    digitalWrite(WATER_VALVE_PIN, HIGH);
}

void closeGasValve() {
    digitalWrite(GAS_VALVE_PIN, HIGH);
}
```

Окремим програмним блоком є модуль повідомлень користувача. Він відповідає за вибір доступного каналу зв’язку. Основним каналом є Wi-Fi, через який повідомлення може бути передане до Telegram-бота, мобільного застосунку або MQTT-сервісу. Якщо Wi-Fi недоступний, використовується резервний GSM-канал.

```
void sendNotification(String message) {
    if (wifiAvailable) {
```

```

    sendWiFiMessage(message);
    lastMessageChannel = "Wi-Fi";
} else if (gsmAvailable) {
    sendGSMMessage(message);
    lastMessageChannel = "GSM";
} else {
    lastMessageChannel = "No connection";
}
}

```

Функція передавання повідомлення через Wi-Fi може бути реалізована різними способами. У базовій реалізації доцільно використати Telegram-бот або MQTT-брокер, оскільки такі рішення дозволяють швидко передати коротке текстове повідомлення користувачу.

```

void sendWiFiMessage(String message) {
    Serial.println("Sending message via Wi-Fi:");
    Serial.println(message);
    // У практичній реалізації тут може бути виклик
    // Telegram-бота, MQTT-публікація або HTTP-запит
}

```

Резервний GSM-канал може бути реалізований за допомогою модуля SIM800L. У цьому випадку система надсилає SMS-повідомлення або виконує тривожний дзвінок на номер користувача.

```

void sendGSMMessage(String message) {
    Serial.println("Sending message via GSM:");
    Serial.println(message);
    // У практичній реалізації тут формуються AT-команди
    // для GSM-модуля SIM800L
}

```

Журналювання подій необхідне для збереження інформації про роботу системи. У базовій реалізації журнал може виводитися в послідовний порт для

налагодження, а в розширеній реалізації — зберігатися у пам'яті пристрою або передаватися до зовнішнього сервісу.

```
void writeLog(EventType event, String message) {
    Serial.println("----- LOG RECORD -----");
    Serial.print("Mode: ");
    Serial.println(currentMode);
    Serial.print("Event: ");
    Serial.println(event);
    Serial.print("Message: ");
    Serial.println(message);
    Serial.print("Channel: ");
    Serial.println(lastMessageChannel);
    Serial.println("-----");
}
```

Для практичного використання журнал подій повинен містити час виникнення події, її тип, джерело, поточний режим системи, виконані дії та результат передавання повідомлення. У разі використання зовнішнього сервісу запис журналу може передаватися у форматі JSON.

Приклад формування JSON-повідомлення для передавання інформації про подію:

```
String createJsonMessage(EventType event, String message) {
    String json = "{";
    json += "\"device_id\": \"ASUBB_01\",";
    json += "\"event\": \"" + String(event) + "\",";
    json += "\"mode\": \"" + String(currentMode) + "\",";
    json += "\"message\": \"" + message + "\",";
    json += "\"channel\": \"" + lastMessageChannel + "\"";
    json += "}";
    шшшшх
    return json;
}
```

```
}
```

У процесі роботи системи важливо забезпечити безпечний початковий і робочий стан виконавчих пристроїв. Після ввімкнення живлення всі реле, сирена, світловий маяк і клапани повинні перебувати у вимкненому або штатному стані. Команди на їхнє ввімкнення мають формуватися лише після підтвердження відповідної події. Це зменшує ризик випадкового спрацювання обладнання під час перезапуску мікроконтролера.

Для перевірки працездатності системи передбачено режим технічного обслуговування. У цьому режимі можна виконати тестове зчитування станів датчиків, перевірити роботу сирени, світлового маяка, релейного модуля, клапанів і каналів зв'язку. При цьому повноцінна тривога не формується, а результати тестування можуть бути виведені в послідовний порт або записані до журналу.

```
void serviceTest() {
    Serial.println("Service mode started");
    readSensors();
    Serial.print("Motion sensor: ");
    Serial.println(motionValue);
    Serial.print("Door sensor: ");
    Serial.println(doorValue);
    Serial.print("Gas value: ");
    Serial.println(gasValue);
    digitalWrite(LIGHT_PIN, HIGH);
    delay(1000);
    digitalWrite(LIGHT_PIN, LOW);
    Serial.println("Service mode completed");
}
```

Під час розроблення програмного забезпечення важливо враховувати можливі помилки роботи системи. До них належать відсутність Wi-Fi-з'єднання, недоступність GSM-модуля, некоректні дані датчика мікроклімату, перевищення порогового значення газу, помилкове спрацювання цифрового датчика або

відсутність відповіді від окремого компонента. Для таких ситуацій програма повинна передбачати службові повідомлення або записи до журналу.

Основні програмні ситуації та реакції системи наведено в таблиці 3.18.

Таблиця 3.18 – Оброблення типових програмних ситуацій

Ситуація	Ознака в програмі	Реакція системи
Wi-Fi доступний	wifiAvailable = true	Повідомлення передається через Wi-Fi
Wi-Fi недоступний	wifiAvailable = false	Використовується GSM-канал
GSM недоступний	gsmAvailable = false	Подія записується до журналу, повторна спроба передавання виконується пізніше
Спрацював датчик руху в режимі охорони	currentMode == MODE_ARMED і motionValue == HIGH	Перехід у режим тривоги
Спрацював датчик руху в режимі очікування	currentMode == MODE_WAITING	Тривога не формується
Перевищено рівень газу	gasValue > gasThreshold	Перехід в аварійний режим, сирена, повідомлення, перекриття газу
Виявлено затоплення	floodValue == HIGH	Перекриття води, повідомлення, запис до журналу
Некоректні дані датчика	Значення не отримано або воно виходить за фізично можливі межі	Формування службового повідомлення

Налагодження програмного забезпечення виконується поетапно. Спочатку перевіряється окрема робота кожного датчика. Потім тестується керування виконавчими пристроями. Після цього перевіряється логіка переходів між режимами роботи. Наступним етапом є тестування алгоритмів реагування на охоронні та аварійні події. Окремо перевіряється надсилання повідомлень через основний і резервний канали зв'язку.

Послідовність тестування програмного забезпечення наведено в таблиці 3.19.

Таблиця 3.19 – Етапи тестування програмного забезпечення системи

Етап тестування	Зміст перевірки	Очікуваний результат
Перевірка запуску системи	Подача живлення та виконання setup()	Система переходить у режим очікування
Перевірка датчиків	Зчитування сигналів від кожного датчика	Дані коректно відображаються у програмі
Перевірка режиму охорони	Увімкнення режиму охорони	Система реагує на рух і відкриття дверей
Перевірка аварійних подій	Імітація диму, газу або затоплення	Система переходить в аварійний режим

Етап тестування	Зміст перевірки	Очікуваний результат
Перевірка виконавчих пристроїв	Активація сирени, маяка, клапанів	Пристрої вмикаються відповідно до команди
Перевірка повідомлень	Передавання повідомлення через Wi-Fi або GSM	Користувач отримує повідомлення
Перевірка журналу	Формування запису після події	Подія зберігається з основними параметрами
Перевірка відмови Wi-Fi	Вимкнення Wi-Fi-з'єднання	Система використовує резервний GSM-канал

Результатом розроблення програмного забезпечення є програма для мікроконтролера ESP32, яка реалізує логіку автоматизованої системи управління безпекою будинку. Програма забезпечує початкове налаштування системи, зчитування сигналів із датчиків, перевірку достовірності даних, визначення типу події, керування режимами роботи, активацію виконавчих пристроїв, надсилання повідомлень користувачу та журналювання подій.

Запропонована програмна структура є модульною, що дозволяє в подальшому розширювати систему без повного переписування програмного коду. До системи можуть бути додані нові датчики, додаткові виконавчі пристрої, інший канал повідомлень або розширений вебінтерфейс користувача. При цьому базова логіка роботи залишається незмінною: зчитування даних, попереднє оброблення, аналіз події, прийняття рішення, реагування, повідомлення користувача та запис до журналу.

Повний програмний код автоматизованої системи управління безпекою будинку наведено в додатку А.

Висновки до розділу 3

У третьому розділі виконано розроблення алгоритмічного та програмного забезпечення автоматизованої системи управління безпекою будинку. На основі технічних рішень, обґрунтованих у попередньому розділі, визначено функціональні вимоги до програмної частини системи, розроблено її функціональну структуру, алгоритми роботи, структуру даних і програмну реалізацію.

У процесі визначення функціональних вимог встановлено, що програмне забезпечення системи має забезпечувати ініціалізацію обладнання, зчитування

сигналів із датчиків, контроль режимів роботи, виявлення охоронних та аварійних подій, активацію виконавчих пристроїв, формування повідомлень користувачу та запис інформації про події до журналу. Окрему увагу приділено поділу подій на охоронні й аварійні. Охоронні події обробляються лише в режимі охорони, тоді як аварійні ситуації, зокрема дим, витік газу або затоплення, мають оброблятися незалежно від поточного режиму роботи системи.

Розроблена функціональна структура автоматизованої системи безпеки відображає взаємодію основних програмно-логічних блоків: підсистеми збору даних, попереднього оброблення сигналів, керування режимами, аналізу подій, прийняття рішень, керування виконавчими пристроями, повідомлень користувача, журналювання та діагностики. Така структура забезпечує послідовне проходження інформації від датчиків до блоку прийняття рішень і формування відповідної реакції системи.

У межах розроблення алгоритмів функціонування сформовано загальний алгоритм роботи системи, алгоритм оброблення охоронної події, алгоритм реагування на аварійні події, алгоритм надсилання повідомлень і алгоритм журналювання. Запропоновані алгоритми передбачають циклічний моніторинг стану об'єкта, перевірку отриманих сигналів, визначення типу події, активацію відповідних виконавчих пристроїв, інформування користувача та збереження інформації про подію.

Для забезпечення впорядкованої роботи програмного забезпечення визначено організацію та структуру даних автоматизованої системи. Дані системи поділено на групи відповідно до їхнього призначення: дані від датчиків, дані про режими роботи, події, виконавчі пристрої, повідомлення користувачу, журнал подій і конфігураційні параметри. Запропонована інформаційна модель дозволяє узгодити зчитування даних, їх оброблення, прийняття рішень і подальше журналювання результатів роботи системи.

На завершальному етапі розроблено програмне забезпечення автоматизованої системи управління безпекою будинку на базі мікроконтролера ESP32. Програмна реалізація передбачає роботу з датчиками руху, відкриття, диму,

газу, затоплення, температури та вологості, а також керування сиреною, світловим маяком, релейними модулями та клапанами перекриття води й газу. Для інформування користувача передбачено використання основного каналу зв'язку Wi-Fi і резервного GSM-каналу.

Таким чином, у розділі сформовано повну логіку функціонування автоматизованої системи безпеки будинку — від отримання сигналів із датчиків до прийняття рішення, виконання керуючих дій, надсилання повідомлень користувачу та запису подій до журналу. Розроблені алгоритми, структура даних і програмне забезпечення створюють основу для практичної реалізації системи та її подальшого вдосконалення.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Під час розроблення та експлуатації автоматизованої системи управління безпекою будинку необхідно враховувати вимоги охорони праці, електробезпеки, пожежної безпеки та безпеки в надзвичайних ситуаціях. Незважаючи на те, що система побудована на базі малопотужного мікроконтролера ESP32 і низьковольтних датчиків, окремі її елементи можуть бути пов'язані з підключенням до електромережі, релейними модулями, виконавчими пристроями, джерелами живлення та монтажними роботами [19].

Дотримання вимог безпеки є важливою умовою як на етапі розроблення програмно-апаратного комплексу, так і під час його встановлення та подальшої експлуатації. Правильна організація робочого місця, безпечне виконання монтажних операцій, використання справного обладнання та своєчасна перевірка системи дозволяють знизити ризик травмування людини, пошкодження обладнання та виникнення аварійних ситуацій [20].

4.1 Аналіз небезпечних і шкідливих виробничих факторів під час розроблення та експлуатації автоматизованої системи безпеки будинку

Під час розроблення автоматизованої системи безпеки будинку основна частина робіт виконується за комп'ютером, а також передбачає підключення електронних компонентів, налагодження мікроконтролера, перевірку датчиків і

виконавчих пристроїв. У процесі виконання таких робіт на людину можуть впливати небезпечні та шкідливі фактори.

До основних факторів, які можуть виникати під час розроблення системи, належать: тривала робота за комп'ютером, зорове напруження, статичне навантаження на опорно-руховий апарат, підвищене психоемоційне навантаження під час налагодження програмного забезпечення, ризик ураження електричним струмом при неправильному підключенні обладнання, а також можливість термічних ушкоджень під час паяння або монтажу електронних компонентів [22].

Під час експлуатації автоматизованої системи безпеки додатковими факторами ризику можуть бути несправність джерела живлення, коротке замикання в електричному колі, перегрівання елементів системи, неправильне підключення релейних модулів або виконавчих пристроїв, а також помилкове спрацювання системи через некоректне встановлення датчиків.

Основні небезпечні та шкідливі фактори наведено в таблиці 4.1.

Таблиця 4.1 – Небезпечні та шкідливі фактори під час розроблення й експлуатації системи

Фактор	Можливі наслідки	Заходи запобігання
Тривала робота за комп'ютером	Зорове напруження, втома, біль у спині	Раціональна організація робочого місця, перерви в роботі, правильне освітлення
Робота з електронними компонентами	Пошкодження обладнання, коротке замикання	Перевірка схеми підключення, використання справних проводів і джерел живлення
Паяння та монтаж	Опіки, вдихання шкідливих випарів	Використання підставки для паяльника, провітрювання приміщення
Робота з електроживленням	Ураження електричним струмом	Відключення живлення перед монтажем, використання низьковольтних джерел живлення
Несправність релейних модулів	Неконтрольоване вмикання виконавчих пристроїв	Перевірка реле, запобіжників і правильності комутації
Помилкове спрацювання датчиків	Хибна тривога або неправильна реакція системи	Налаштування порогів, тестування датчиків, правильне місце встановлення

Таким чином, найбільш важливими факторами ризику під час розроблення та експлуатації системи є електробезпека, правильність монтажу, безпечна організація робочого місця та надійність роботи електронних компонентів.

4.2 Заходи з охорони праці під час монтажу та експлуатації системи безпеки

Під час монтажу автоматизованої системи безпеки будинку необхідно дотримуватися правил безпечної роботи з електронними пристроями та електричними колами. Усі монтажні роботи слід виконувати лише при вимкненому живленні. Перед підключенням датчиків, релейних модулів, сирени, світлового маяка та електроприводних клапанів необхідно перевірити правильність схеми з'єднань і відповідність напруги живлення технічним характеристикам компонентів.

Особливу увагу слід приділяти живленню системи. Мікроконтролер ESP32, датчики та модулі повинні працювати від стабілізованого низьковольтного джерела живлення. Не допускається безпосереднє підключення низьковольтних електронних компонентів до мережі 220 В. Якщо в системі використовуються релейні модулі для керування пристроями, що працюють від вищої напруги, їх підключення повинно виконуватися з дотриманням електробезпеки та із забезпеченням надійної ізоляції силових кіл [5].

Під час монтажу датчиків необхідно враховувати їхнє функціональне призначення. Датчики руху доцільно встановлювати в місцях можливого проходу людини, але без прямого впливу джерел тепла. Герконові датчики встановлюються на дверях або вікнах так, щоб забезпечити стабільне спрацювання при відкритті. Датчики диму потрібно розміщувати у зонах, де можливе накопичення диму, а датчики затоплення — у місцях імовірного витоку води. Датчик газу необхідно встановлювати з урахуванням особливостей газу, який контролюється, та забезпечити достатній доступ повітря до чутливого елемента.

Під час експлуатації системи користувач повинен періодично перевіряти її працездатність. Доцільно виконувати тестування датчиків, перевірку сирени, світлового маяка, каналів зв'язку та виконавчих пристроїв. Якщо система використовує резервне живлення або GSM-модуль, необхідно контролювати їхній стан і справність [19].

Для безпечної експлуатації автоматизованої системи безпеки будинку необхідно дотримуватися таких заходів:

1. виконувати монтаж і підключення системи лише при вимкненому живленні;
2. використовувати справні джерела живлення, дроти, клеми та з'єднувальні елементи;
3. не допускати перевантаження релейних модулів і виконавчих пристроїв;
4. ізолювати відкриті контакти та місця з'єднань;
5. не розміщувати електронні модулі в місцях із підвищеною вологістю без захисного корпусу;
6. періодично перевіряти працездатність датчиків і каналів повідомлення;
7. не виконувати ремонт системи без попереднього відключення живлення;
8. забезпечити доступ користувача до інструкції з експлуатації та порядку дій у разі спрацювання системи.

Дотримання зазначених заходів дозволяє зменшити ризик виникнення несправностей, підвищити надійність роботи системи та забезпечити безпечну експлуатацію обладнання.

4.3 Пожежна безпека об'єкта та заходи щодо запобігання надзвичайним ситуаціям

Пожежна безпека є одним із ключових аспектів експлуатації автоматизованої системи безпеки будинку. Розроблювана система передбачає контроль пожежонебезпечних та аварійних факторів, зокрема появи диму, перевищення рівня газу, затоплення та небезпечних відхилень параметрів мікроклімату. Завдяки використанню відповідних датчиків система може своєчасно виявити небезпечну ситуацію та сформулювати повідомлення користувачу [4].

Основними причинами виникнення пожежі в житловому будинку можуть бути коротке замикання електропроводки, перегрів електроприладів, неправильна експлуатація газового обладнання, порушення правил користування побутовими приладами, залишення ввімкнених електричних пристроїв без нагляду, а також несправність джерел живлення або комутаційних елементів.

Для запобігання пожежі під час експлуатації системи безпеки необхідно використовувати справні джерела живлення, не допускати перегрівання електронних компонентів, забезпечити правильне підключення релейних модулів і виконавчих пристроїв, а також розміщувати електронні блоки в захисних корпусах. Проводи повинні бути надійно закріплені, а місця з'єднань — ізольовані [8].

У разі виявлення диму система повинна активувати звукове та світлове оповіщення, сформувані повідомлення користувачу та записати подію до журналу. У разі перевищення допустимого рівня газу система повинна сформувані аварійне повідомлення, увімкнути сигналізацію та, за наявності відповідного виконавчого пристрою, подати команду на перекриття газу. У разі затоплення система може подати команду на перекриття води, що дозволяє зменшити можливі матеріальні збитки.

Під час виникнення надзвичайної ситуації користувач повинен діяти відповідно до характеру події. У разі пожежі необхідно негайно залишити небезпечну зону, повідомити відповідні служби та не намагатися самотійно ліквідувати пожежу, якщо це створює загрозу життю. У разі виявлення запаху газу необхідно припинити користування електроприладами, забезпечити провітрювання приміщення, перекрити подачу газу та звернутися до аварійної служби. У разі затоплення необхідно перекрити воду, вимкнути електроживлення у небезпечній зоні та усунути джерело витоку [21].

Таким чином, автоматизована система управління безпекою будинку не лише виконує функції охоронної сигналізації, а й сприяє підвищенню рівня пожежної та аварійної безпеки об'єкта. Своєчасне виявлення небезпечних подій, автоматизоване формування повідомлень і можливість керування виконавчими пристроями дозволяють зменшити час реагування та мінімізувати наслідки надзвичайних ситуацій.

Висновки до розділу 4

У четвертому розділі розглянуто питання охорони праці та безпеки в надзвичайних ситуаціях під час розроблення, монтажу й експлуатації автоматизованої системи управління безпекою будинку. Визначено основні

небезпечні та шкідливі фактори, які можуть виникати під час роботи з комп'ютерною технікою, електронними компонентами, джерелами живлення та виконавчими пристроями.

Запропоновано заходи з охорони праці під час монтажу та експлуатації системи, зокрема виконання робіт при вимкненому живленні, використання справних компонентів, правильну ізоляцію електричних з'єднань, перевірку датчиків і періодичне тестування працездатності системи. Окремо розглянуто питання пожежної безпеки об'єкта та порядок реагування на аварійні ситуації, пов'язані з димом, витоком газу або затопленням.

Дотримання визначених заходів дозволяє забезпечити безпечні умови роботи під час розроблення системи, знизити ризик аварій під час її експлуатації та підвищити загальний рівень захисту житлового будинку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Базилевич В. М., Мальцева М. В., Петренко Т. А., Черниш Л. Г. Захищена система розумного будинку з використанням Internet of Things. Технічні науки та технології. 2020. № 2(20). С. 218–228. DOI: [https://doi.org/10.25140/2411-5363-2020-2\(20\)-218-228](https://doi.org/10.25140/2411-5363-2020-2(20)-218-228).
2. Галичак Н. Система «розумний будинок» як спосіб енергозбереження. Формування механізму зміцнення конкурентних позицій національних економічних систем у глобальному, регіональному та локальному вимірах : матеріали конф. Тернопіль : ТНТУ, 2020. С. 113–114. URL: https://elartu.tntu.edu.ua/bitstream/lib/31473/2/FMZKP_2020_Halychak_N-The_smart_house_system_113-114.pdf (дата звернення: 18.06.2026).
3. Датчики розумного будинку від Ajax Systems : офіц. сайт Ajax Systems. URL: <https://ajax.systems/ua/catalogue/comfort-and-automation/> (дата звернення: 18.06.2026).
4. ДБН В.2.5-20:2018. Газопостачання. URL: https://e-construction.gov.ua/laws_detail/3200377925350196674?doc_type=2 (дата звернення: 18.06.2026).
5. ДБН В.2.5-23:2025. Проектування електроустановок житлових будинків та громадських будівель і споруд. URL: https://e-construction.gov.ua/laws_detail/3728029580368283042?doc_type=2 (дата звернення: 18.06.2026).
6. ДБН В.2.5-56:2014. Системи протипожежного захисту. URL: https://e-construction.gov.ua/laws_detail/3200383488549193714?doc_type=2 (дата звернення: 18.06.2026).
7. ДБН В.2.5-67:2013. Опалення, вентиляція та кондиціонування. URL: https://e-construction.gov.ua/laws_detail/3074971619479783152 (дата звернення: 18.06.2026).
8. ДБН В.2.5-76:2014. Автоматизовані системи раннього виявлення загрози виникнення надзвичайних ситуацій та оповіщення населення. URL: https://e-construction.gov.ua/laws_detail/3074971619479783152

- construction.gov.ua/laws_detail/3200403474810405979 (дата звернення: 18.06.2026).
9. Додон Д. Ю., Риндюк С. В. Розумний будинок — ефективна технологія безпечного житла. Матеріали науково-технічної конференції підрозділів Вінницького національного технічного університету. Вінниця : ВНТУ, 2024. URL: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/42273/20990.pdf> (дата звернення: 18.06.2026).
 10. ДСТУ ISO/IEC 27001:2023 (ISO/IEC 27001:2022, IDT). Інформаційна безпека, кібербезпека та захист конфіденційності. Системи керування інформаційною безпекою. Вимоги : прийнято наказом ДП «УкрНДНЦ» від 17.08.2023 № 210 // База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/v0210774-23> (дата звернення: 18.06.2026).
 11. Дужак І. О. Розумний будинок. Автоматизація технологічних і бізнес-процесів. 2013. № 13–14. С. 31–33. DOI: <https://doi.org/10.15673/2312-3125.13-14/2010.32920>.
 12. Жураковський Б. Ю., Зенів І. О. Технології інтернету речей : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2021. 271 с. URL: <https://ela.kpi.ua/items/03f7d3c6-aa4a-4615-b864-0b4a6ab409e4> (дата звернення: 18.06.2026).
 13. Інструкція з використання Hub (2G) / (4G) Jeweller : офіц. документація Ajax Systems. URL: <https://ajax.systems/ua/support/manuals/hub/> (дата звернення: 18.06.2026).
 14. Кодекс цивільного захисту України : Кодекс України від 02.10.2012 № 5403-VI // База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/5403-17> (дата звернення: 18.06.2026).
 15. Кравчук С. О., Міночкін Д. А., Кайденко М. М. Застосування телекомунікаційних технологій в структурі «розумного будинку». Системи обробки інформації. 2017. Вип. 1(147). С. 22–26. DOI: <https://doi.org/10.30748/soi.2017.147.05>.
 16. Макєєв С. О., Вировець С. В. Сучасна апаратура та обладнання систем «розумний будинок». Вісник НТУ «ХПІ». Серія: Проблеми удосконалювання

- електричних машин і апаратів. Теорія і практика. 2020. № 2. С. 16–20. DOI: <https://doi.org/10.20998/2079-3944.2020.2.03>.
17. Невлюдов І. Ш., Новоселов С. П., Андрусович В. А., Резніченко О. Г. Технології Інтернету речей в управлінні пристроями на мікроконтролерах : навч. посіб. Харків : ХНУРЕ, 2023. DOI: <https://doi.org/10.30837/978-966-659-364-4>. ISBN 978-966-659-364-4. URL: <https://publish.nure.ua/catalog/book/339> (дата звернення: 18.06.2026).
 18. Охоронна система та автоматизація для будинку : офіц. сайт Ajax Systems. URL: <https://ajax.systems/ua/solutions-by-facility-type/private-houses/> (дата звернення: 18.06.2026).
 19. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями : наказ Міністерства соціальної політики України від 14.02.2018 № 207 // База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/z0508-18> (дата звернення: 18.06.2026).
 20. Про затвердження Правил безпечної експлуатації електроустановок споживачів : наказ Міністерства праці та соціальної політики України від 09.01.1998 № 4 // База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/z0093-98> (дата звернення: 18.06.2026).
 21. Про затвердження Правил пожежної безпеки в Україні : наказ Міністерства внутрішніх справ України від 30.12.2014 № 1417 // База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/z0252-15> (дата звернення: 18.06.2026).
 22. Про охорону праці : Закон України від 14.10.1992 № 2694-XII // База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/2694-12> (дата звернення: 18.06.2026).
 23. Рябчун Ю., Середа Д., Кохан В., Доля О. Можливості та переваги українського ринку технологій «розумний будинок». Управління розвитком складних систем. 2023. № 56. С. 181–187. DOI: <https://doi.org/10.32347/2412-9933.2023.56.181-187>.

24. Сагун А. В., Хайдуров В. В., Поліщук І. А. Розробка програмних модулів для обміну даними у промислових мережах : навч. посіб. для студ. спец. 151 «Автоматизація та комп'ютерно-інтегровані технології». Київ : КПІ ім. Ігоря Сікорського, 2021. 103 с. URL: https://ela.kpi.ua/bitstream/123456789/46647/1/RPMODPM_posibnyk.pdf (дата звернення: 18.06.2026).
25. Сторчак К. П., Тушич А. М., Срібна І. М., Яковенко Н. Д., Кравець Д. В. Технології Інтернет речей : навч. посіб. Київ : Державний університет телекомунікацій, 2021. 68 с. URL: <https://files.znu.edu.ua/files/Bibliobooks/Inshi83/0063485.pdf> (дата звернення: 18.06.2026).
26. Строкань О. В., Назаров Є. М. Система управління «Розумний будинок» на основі технології Internet of Things. Вісник Національного технічного університету «ХПІ». Серія: Нові рішення в сучасних технологіях. 2022. № 2(12). С. 42–47. DOI: <https://doi.org/10.20998/2413-4295.2022.02.06>.
27. Уланська М. Правовий погляд на функціонування технологій «інтернет речей» та «розумний будинок». Цивілістична платформа. 2024. № 1. С. 233–254. DOI: <https://doi.org/10.69724/2786-8834-2024-1-1-233-254>.
28. Федько А. О., Зеленцова І. Я., Тягунова М. Ю. Розробка та моделювання системи «Розумний будинок» з підвищеною надійністю на базі Arduino. Наукові праці Донецького національного технічного університету. Серія: Проблеми моделювання та автоматизації проектування. 2019. № 1(14). С. 16–25. DOI: <https://doi.org/10.31474/2074-7888-2019-1-23-32>.

ДОДАТКИ

Додаток А. Програмний код автоматизованої системи управління безпекою будинку

/*

*Автоматизована система управління безпекою будинку
Мікроконтролер: ESP32*

Основні функції програми:

- зчитування даних із датчиків руху, відкриття, диму, газу, затоплення;
- контроль температури та вологості за допомогою DHT22;
- визначення охоронних та аварійних подій;
- керування сиреною, світловим маяком, клапаном перекриття води та газу;
- надсилання повідомлень користувачу через Wi-Fi;
- використання GSM-каналу як резервного способу повідомлення;
- запис інформації про події до журналу через Serial Monitor.

*/

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <DHT.h>
```

```
/* -----
   1. НАЛАШТУВАННЯ МЕРЕЖІ ТА КАНАЛІВ ПОВІДОМЛЕНЬ
   ----- */
```

```
// Дані Wi-Fi-мережі
const char* WIFI_SSID = "YOUR_WIFI_NAME";
const char* WIFI_PASSWORD = "YOUR_WIFI_PASSWORD";
```

```
// Дані Telegram-бота
const char* BOT_TOKEN = "YOUR_TELEGRAM_BOT_TOKEN";
const char* CHAT_ID = "YOUR_CHAT_ID";
```

```
// Номер телефону користувача для резервного GSM-повідомлення
const char* USER_PHONE = "+380XXXXXXXXXX";
```

```
// Об'єкти для роботи з Telegram
WiFiClientSecure securedClient;
UniversalTelegramBot bot(BOT_TOKEN, securedClient);
```

```
// GSM-модуль SIM800L підключається до апаратного порту UART
HardwareSerial sim800(2);
```

```
// Піни UART для GSM-модуля
const int GSM_RX_PIN = 16;
const int GSM_TX_PIN = 17;
```

```

/* -----
2. ПРИЗНАЧЕННЯ ВХОДІВ І ВИХОДІВ ESP32
----- */

```

```

// Входи датчиків
const int PIR_PIN = 14;      // датчик руху HC-SR501
const int DOOR_PIN = 27;     // герконовий датчик MC-38
const int SMOKE_PIN = 26;    // датчик диму
const int FLOOD_PIN = 25;    // датчик затоплення LM393
const int GAS_PIN = 34;      // аналоговий вхід для датчика газу MQ-2 / MQ-5
const int DHT_PIN = 4;       // датчик температури та вологості DHT22

```

```

// Виходи виконавчих пристроїв
const int SIREN_PIN = 18;    // сирена
const int LIGHT_PIN = 19;    // світловий маяк
const int WATER_VALVE_PIN = 21; // клапан перекриття води
const int GAS_VALVE_PIN = 22; // клапан перекриття газу

```

```

// Тип датчика мікроклімату
#define DHT_TYPE DHT22
DHT dht(DHT_PIN, DHT_TYPE);

```

```

/* -----
3. ЛОГІЧНІ РІВНІ СПРАЦЮВАННЯ ДАТЧИКІВ
----- */

```

У різних модулях логіка спрацювання може відрізнятися.
Тому активні рівні винесені в окремі константи.
За потреби їх можна змінити відповідно до реального підключення.

```

const int PIR_ACTIVE_LEVEL = HIGH;    // рух виявлено
const int DOOR_OPEN_LEVEL = HIGH;    // двері/вікно відкрито
const int SMOKE_ACTIVE_LEVEL = HIGH;  // дим виявлено
const int FLOOD_ACTIVE_LEVEL = HIGH;  // затоплення виявлено

```

```

/* -----
4. ПОРОГОВІ ЗНАЧЕННЯ ТА ПАРАМЕТРИ СИСТЕМИ
----- */

```

```

int gasThreshold = 1800;    // поріг для датчика газу
float tempMin = 5.0;        // мінімально допустима температура
float tempMax = 45.0;       // максимально допустима температура
float humidityMax = 80.0;    // максимально допустима вологість

```

```

unsigned long alarmDelay = 3000; // затримка підтвердження охоронної події, мс
unsigned long sensorReadDelay = 500; // пауза між циклами опитування датчиків, мс

```

```

/* -----

```

5. РЕЖИМИ РОБОТИ СИСТЕМИ

```

----- */

enum SystemMode {
    MODE_WAITING,    // режим очікування
    MODE_ARMED,      // режим охорони
    MODE_ALARM,      // режим тривоги
    MODE_EMERGENCY,  // аварійний режим
    MODE_SERVICE     // режим технічного обслуговування
};

```

```

SystemMode currentMode = MODE_WAITING;

```

```

/* -----
6. ТИПИ ПОДІЙ
----- */

```

```

enum EventType {
    EVENT_NONE,
    EVENT_MOTION,
    EVENT_DOOR_OPEN,
    EVENT_SMOKE,
    EVENT_GAS,
    EVENT_FLOOD,
    EVENT_MICROCLIMATE,
    EVENT_WIFI_LOST,
    EVENT_SENSOR_ERROR,
    EVENT_SYSTEM_ARMED,
    EVENT_SYSTEM_DISARMED
};

```

```

/* -----
7. СТРУКТУРИ ДАНИХ
----- */

```

```

struct SensorData {
    String sensorId;
    String sensorType;
    int currentValue;
    int thresholdValue;
    bool isActive;
    unsigned long lastReadTime;
    String sensorStatus;
};

```

```

struct SystemState {
    SystemMode currentMode;
    EventType activeEvent;
    bool alarmState;
    bool emergencyState;
    bool wifiAvailable;
};

```

```

    bool gsmAvailable;
    bool sirenState;
    bool lightState;
    bool valveWaterState;
    bool valveGasState;
};

```

```

struct EventData {
    int eventId;
    EventType eventType;
    String eventSource;
    unsigned long eventTime;
    SystemMode systemMode;
    String priority;
    String messageText;
    String actionList;
    String processingResult;
};

```

```

struct LogRecord {
    int logId;
    unsigned long dateTime;
    EventType eventType;
    String eventSource;
    SystemMode systemMode;
    String performedActions;
    String messageChannel;
    String processingResult;
};

```

```

/* -----
   8. ГЛОБАЛЬНІ ЗМІННІ ДЛЯ ЗБЕРЕЖЕННЯ ПОТОЧНИХ ДАНИХ
   ----- */

```

```

int motionValue = 0;
int doorValue = 0;
int smokeValue = 0;
int floodValue = 0;
int gasValue = 0;

```

```

float temperature = 0.0;
float humidity = 0.0;

```

```

bool wifiAvailable = false;
bool gsmAvailable = false;

```

```

String lastMessageChannel = "none";

```

```

int eventCounter = 0;
int logCounter = 0;

```

```
SystemState systemState;
```

```
/* -----  
9. ДОПОМІЖНІ ФУНКЦІЇ ДЛЯ ПЕРЕТВОРЕННЯ ДАНИХ У ТЕКСТ  
----- */
```

```
String modeToString(SystemMode mode) {  
    switch (mode) {  
        case MODE_WAITING:  
            return "Очікування";  
        case MODE_ARMED:  
            return "Охорона";  
        case MODE_ALARM:  
            return "Тривога";  
        case MODE_EMERGENCY:  
            return "Аварія";  
        case MODE_SERVICE:  
            return "Технічне обслуговування";  
        default:  
            return "Невідомий режим";  
    }  
}
```

```
String eventToString(EventType event) {  
    switch (event) {  
        case EVENT_NONE:  
            return "Подія відсутня";  
        case EVENT_MOTION:  
            return "Виявлено рух";  
        case EVENT_DOOR_OPEN:  
            return "Відкрито двері або вікно";  
        case EVENT_SMOKE:  
            return "Виявлено дим";  
        case EVENT_GAS:  
            return "Перевищено рівень газу";  
        case EVENT_FLOOD:  
            return "Виявлено затоплення";  
        case EVENT_MICROCLIMATE:  
            return "Параметри мікроклімату поза нормою";  
        case EVENT_WIFI_LOST:  
            return "Втрачено Wi-Fi-з'єднання";  
        case EVENT_SENSOR_ERROR:  
            return "Помилка датчика";  
        case EVENT_SYSTEM_ARMED:  
            return "Систему поставлено під охорону";  
        case EVENT_SYSTEM_DISARMED:  
            return "Систему знято з охорони";  
        default:  
            return "Невідома подія";  
    }  
}
```

```
/* -----
10. ІНІЦІАЛІЗАЦІЯ ВХОДІВ ТА ВИХОДІВ
----- */
```

```
void setupPins() {
  pinMode(PIR_PIN, INPUT);
  pinMode(DOOR_PIN, INPUT_PULLUP);
  pinMode(SMOKE_PIN, INPUT);
  pinMode(FLOOD_PIN, INPUT);
  pinMode(GAS_PIN, INPUT);

  pinMode(SIREN_PIN, OUTPUT);
  pinMode(LIGHT_PIN, OUTPUT);
  pinMode(WATER_VALVE_PIN, OUTPUT);
  pinMode(GAS_VALVE_PIN, OUTPUT);

  digitalWrite(SIREN_PIN, LOW);
  digitalWrite(LIGHT_PIN, LOW);
  digitalWrite(WATER_VALVE_PIN, LOW);
  digitalWrite(GAS_VALVE_PIN, LOW);
}
```

```
/* -----
11. ІНІЦІАЛІЗАЦІЯ ДАТЧИКІВ
----- */
```

```
void initSensors() {
  dht.begin();
  Serial.println("Датчики ініціалізовано");
}
```

```
/* -----
12. ПІДКЛЮЧЕННЯ ДО WI-FI
----- */
```

```
void initWiFi() {
  Serial.println("Підключення до Wi-Fi...");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  int attempts = 0;

  while (WiFi.status() != WL_CONNECTED && attempts < 20) {
    delay(500);
    Serial.print(".");
    attempts++;
  }

  if (WiFi.status() == WL_CONNECTED) {
    wifiAvailable = true;
    securedClient.setInsecure();
    Serial.println();
  }
}
```

```

    Serial.println("Wi-Fi підключено");
    Serial.print("IP-адреса: ");
    Serial.println(WiFi.localIP());
} else {
    wifiAvailable = false;
    Serial.println();
    Serial.println("Wi-Fi недоступний");
}
}

/* -----
   13. ІНІЦІАЛІЗАЦІЯ GSM-МОДУЛЯ
   ----- */

void initGSM() {
    Serial.println("Ініціалізація GSM-модуля...");

    sim800.begin(9600, SERIAL_8N1, GSM_RX_PIN, GSM_TX_PIN);
    delay(1000);

    sim800.println("AT");
    delay(1000);

    if (sim800.available()) {
        String response = sim800.readString();
        Serial.println(response);

        if (response.indexOf("OK") >= 0) {
            gsmAvailable = true;
            Serial.println("GSM-модуль доступний");
        } else {
            gsmAvailable = false;
            Serial.println("GSM-модуль не відповідає коректно");
        }
    } else {
        gsmAvailable = false;
        Serial.println("GSM-модуль недоступний");
    }
}

/* -----
   14. ІНІЦІАЛІЗАЦІЯ ЗАГАЛЬНОГО СТАНУ СИСТЕМИ
   ----- */

void initState() {
    systemState.currentMode = MODE_WAITING;
    systemState.activeEvent = EVENT_NONE;
    systemState.alarmState = false;
    systemState.emergencyState = false;
    systemState.wifiAvailable = wifiAvailable;
    systemState.gsmAvailable = gsmAvailable;
}

```

```

systemState.sirenState = false;
systemState.lightState = false;
systemState.valveWaterState = false;
systemState.valveGasState = false;
}

/* -----
15. ФУНКЦІЯ SETUP()
----- */

void setup() {
  Serial.begin(115200);
  delay(1000);

  Serial.println("-----");
  Serial.println("Автоматизована система безпеки");
  Serial.println("Запуск системи...");
  Serial.println("-----");

  setupPins();
  initSensors();
  initWiFi();

  if (!wifiAvailable) {
    initGSM();
  }

  initSystemState();

  currentMode = MODE_WAITING;

  Serial.println("Система запущена в режимі очікування");
  writeSystemLog("Система запущена");
}

/* -----
16. ОСНОВНИЙ ЦИКЛ LOOP()
----- */

void loop() {
  checkWiFiStatus();

  readSensors();
  EventType event = detectEvent();

  if (event != EVENT_NONE) {
    handleEvent(event);
  }

  delay(sensorReadDelay);
}

```

```

}

/* -----
  17. ПЕРЕВІРКА СТАНУ WI-FI
  ----- */

void checkWiFiStatus() {
  bool currentWifiStatus = (WiFi.status() == WL_CONNECTED);

  if (wifiAvailable && !currentWifiStatus) {
    wifiAvailable = false;
    systemState.wifiAvailable = false;

    writeSystemLog("Втрачено Wi-Fi-з'єднання");

    if (!gsmAvailable) {
      initGSM();
    }
  }

  if (!wifiAvailable && currentWifiStatus) {
    wifiAvailable = true;
    systemState.wifiAvailable = true;
    writeSystemLog("Wi-Fi-з'єднання відновлено");
  }
}

/* -----
  18. ЗЧИТУВАННЯ ДАНИХ ІЗ ДАТЧИКІВ
  ----- */

void readSensors() {
  motionValue = digitalRead(PIR_PIN);
  doorValue = digitalRead(DOOR_PIN);
  smokeValue = digitalRead(SMOKE_PIN);
  floodValue = digitalRead(FLOOD_PIN);
  gasValue = analogRead(GAS_PIN);

  temperature = dht.readTemperature();
  humidity = dht.readHumidity();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Помилка зчитування DHT22");
  }
}

/* -----
  19. ПІДТВЕРДЖЕННЯ ЦИФРОВОГО СИГНАЛУ
  ----- */

```

```

bool confirmDigitalSignal(int pin, int activeLevel) {
    int firstRead = digitalRead(pin);
    delay(100);
    int secondRead = digitalRead(pin);

    if (firstRead == activeLevel && secondRead == activeLevel) {
        return true;
    }

    return false;
}

/* -----
   20. ВИЯВЛЕННЯ ПОДІЙ
   ----- */

EventType detectEvent() {
    // Аварійні події перевіряються першими,
    // оскільки вони не залежать від режиму охорони.

    if (smokeValue == SMOKE_ACTIVE_LEVEL) {
        if (confirmDigitalSignal(SMOKE_PIN, SMOKE_ACTIVE_LEVEL)) {
            return EVENT_SMOKE;
        }
    }

    if (gasValue > gasThreshold) {
        return EVENT_GAS;
    }

    if (floodValue == FLOOD_ACTIVE_LEVEL) {
        if (confirmDigitalSignal(FLOOD_PIN, FLOOD_ACTIVE_LEVEL)) {
            return EVENT_FLOOD;
        }
    }

    if (!isnan(temperature) && !isnan(humidity)) {
        if (temperature < tempMin || temperature > tempMax || humidity > humidityMax) {
            return EVENT_MICROCLIMATE;
        }
    }

    // Охоронні події перевіряються лише в режимі охорони.

    if (currentMode == MODE_ARMED) {
        if (motionValue == PIR_ACTIVE_LEVEL) {
            if (confirmDigitalSignal(PIR_PIN, PIR_ACTIVE_LEVEL)) {
                return EVENT_MOTION;
            }
        }
    }
}

```

```

    if (doorValue == DOOR_OPEN_LEVEL) {
        if (confirmDigitalSignal(DOOR_PIN, DOOR_OPEN_LEVEL)) {
            return EVENT_DOOR_OPEN;
        }
    }
}

return EVENT_NONE;
}

```

```

/* -----
21. ОБРОБЛЕННЯ ПОДІЇ
----- */

```

```

void handleEvent(EventType event) {
    String message = "";
    String actions = "";

    switch (event) {
        case EVENT_MOTION:
            currentMode = MODE_ALARM;
            systemState.currentMode = MODE_ALARM;
            systemState.alarmState = true;
            systemState.activeEvent = event;

            activateAlarmDevices();

            message = "Тривога! Виявлено рух у приміщенні.";
            actions = "Увімкнено сирену та світловий маяк";
            break;

        case EVENT_DOOR_OPEN:
            currentMode = MODE_ALARM;
            systemState.currentMode = MODE_ALARM;
            systemState.alarmState = true;
            systemState.activeEvent = event;

            activateAlarmDevices();

            message = "Тривога! Відкрито двері або вікно.";
            actions = "Увімкнено сирену та світловий маяк";
            break;

        case EVENT_SMOKE:
            currentMode = MODE_EMERGENCY;
            systemState.currentMode = MODE_EMERGENCY;
            systemState.emergencyState = true;
            systemState.activeEvent = event;

```

```

activateAlarmDevices();

message = "Аварія! Виявлено дим. Можлива пожежа.";
actions = "Увімкнено сирену та світловий маяк";
break;

case EVENT_GAS:
    currentMode = MODE_EMERGENCY;
    systemState.currentMode = MODE_EMERGENCY;
    systemState.emergencyState = true;
    systemState.activeEvent = event;

    activateAlarmDevices();
    closeGasValve();

    message = "Аварія! Перевищено рівень газу.";
    actions = "Увімкнено сирену, світловий маяк, подано команду на перекриття газу";
    break;

case EVENT_FLOOD:
    currentMode = MODE_EMERGENCY;
    systemState.currentMode = MODE_EMERGENCY;
    systemState.emergencyState = true;
    systemState.activeEvent = event;

    closeWaterValve();

    message = "Аварія! Виявлено затоплення.";
    actions = "Подано команду на перекриття води";
    break;

case EVENT_MICROCLIMATE:
    systemState.activeEvent = event;

    message = "Попередження! Параметри мікроклімату поза нормою.";
    actions = "Сформовано попередження користувачу";
    break;

default:
    return;
}

sendNotification(message);
writeEventLog(event, message, actions);
}

/* -----
22. КЕРУВАННЯ ВИКОНАВЧИМИ ПРИСТРОЯМИ
----- */

```

```

void activateAlarmDevices() {
    digitalWrite(SIREN_PIN, HIGH);
    digitalWrite(LIGHT_PIN, HIGH);

    systemState.sirenState = true;
    systemState.lightState = true;
}

void deactivateAlarmDevices() {
    digitalWrite(SIREN_PIN, LOW);
    digitalWrite(LIGHT_PIN, LOW);

    systemState.sirenState = false;
    systemState.lightState = false;
}

void closeWaterValve() {
    digitalWrite(WATER_VALVE_PIN, HIGH);
    systemState.valveWaterState = true;
}

void openWaterValve() {
    digitalWrite(WATER_VALVE_PIN, LOW);
    systemState.valveWaterState = false;
}

void closeGasValve() {
    digitalWrite(GAS_VALVE_PIN, HIGH);
    systemState.valveGasState = true;
}

void openGasValve() {
    digitalWrite(GAS_VALVE_PIN, LOW);
    systemState.valveGasState = false;
}

/* -----
   23. НАДСИЛАННЯ ПОВІДОМЛЕНЬ КОРИСТУВАЧУ
   ----- */

void sendNotification(String message) {
    if (wifiAvailable) {
        sendTelegramMessage(message);
        lastMessageChannel = "Wi-Fi / Telegram";
    } else if (gsmAvailable) {
        sendSMS(message);
        lastMessageChannel = "GSM / SMS";
    } else {
        lastMessageChannel = "Канал зв'язку недоступний";
        Serial.println("Не вдалося надіслати повідомлення: канали зв'язку недоступні");
    }
}

```

```

}

void sendTelegramMessage(String message) {
    Serial.println("Надсилення повідомлення через Telegram:");
    Serial.println(message);

    bool result = bot.sendMessage(CHAT_ID, message, "");

    if (result) {
        Serial.println("Telegram-повідомлення надіслано");
    } else {
        Serial.println("Помилка надсилення Telegram-повідомлення");
    }
}

```

```

void sendSMS(String message) {
    Serial.println("Надсилення SMS через GSM:");
    Serial.println(message);

    sim800.println("AT+CMGF=1");
    delay(1000);

    sim800.print("AT+CMGS=\");
    sim800.print(USER_PHONE);
    sim800.println("\");
    delay(1000);

    sim800.print(message);
    delay(500);

    sim800.write(26);
    delay(3000);

    Serial.println("Команду на надсилення SMS виконано");
}

```

```

/* -----
24. ФОРМУВАННЯ JSON-ПОВІДОМЛЕННЯ
----- */

```

```

String createJsonMessage(EventType event, String message) {
    String json = "{";
    json += "\"device_id\":\"ASUBB_01\"";
    json += "\"event_type\":\"" + eventToString(event) + "\"";
    json += "\"mode\":\"" + modeToString(currentMode) + "\"";
    json += "\"message\":\"" + message + "\"";
    json += "\"channel\":\"" + lastMessageChannel + "\"";
    json += "\"gas_value\":\"" + String(gasValue) + "\"";
    json += "\"temperature\":\"" + String(temperature) + "\"";
    json += "\"humidity\":\"" + String(humidity) + "\"";
}

```

```

    json += "}";

    return json;
}

/* -----
   25. ЖУРНАЛЮВАННЯ ПОДІЙ
   ----- */

void writeEventLog(EventType event, String message, String actions) {
    logCounter++;

    Serial.println();
    Serial.println("----- ЗАПИС ЖУРНАЛУ -----");
    Serial.print("Номер запису: ");
    Serial.println(logCounter);

    Serial.print("Час від запуску, мс: ");
    Serial.println(millis());

    Serial.print("Режим системи: ");
    Serial.println(modeToString(currentMode));

    Serial.print("Тип події: ");
    Serial.println(eventToString(event));

    Serial.print("Повідомлення: ");
    Serial.println(message);

    Serial.print("Виконані дії: ");
    Serial.println(actions);

    Serial.print("Канал повідомлення: ");
    Serial.println(lastMessageChannel);

    Serial.print("Значення газу: ");
    Serial.println(gasValue);

    Serial.print("Температура: ");
    Serial.println(temperature);

    Serial.print("Вологість: ");
    Serial.println(humidity);

    Serial.println("JSON-представлення події:");
    Serial.println(createJsonMessage(event, message));

    Serial.println("-----");
    Serial.println();
}

```

```

void writeSystemLog(String text) {
    Serial.println();
    Serial.println("----- СЛУЖБОВИЙ ЗАПИС -----");
    Serial.print("Час від запуску, мс: ");
    Serial.println(millis());
    Serial.println(text);
    Serial.println("-----");
    Serial.println();
}

/* -----
   26. КЕРУВАННЯ РЕЖИМАМИ РОБОТИ
   ----- */

void armSystem() {
    currentMode = MODE_ARMED;
    systemState.currentMode = MODE_ARMED;

    String message = "Систему поставлено під охорону.";
    sendNotification(message);
    writeEventLog(EVENT_SYSTEM_ARMED, message, "Активовано режим охорони");
}

void disarmSystem() {
    currentMode = MODE_WAITING;
    systemState.currentMode = MODE_WAITING;
    systemState.alarmState = false;
    systemState.emergencyState = false;
    systemState.activeEvent = EVENT_NONE;

    deactivateAlarmDevices();

    String message = "Систему знято з охорони.";
    sendNotification(message);
    writeEventLog(EVENT_SYSTEM_DISARMED, message, "Систему переведено в режим очікування");
}

void resetEmergencyMode() {
    currentMode = MODE_WAITING;
    systemState.currentMode = MODE_WAITING;
    systemState.emergencyState = false;
    systemState.activeEvent = EVENT_NONE;

    deactivateAlarmDevices();

    String message = "Аварійний режим скинуто. Система повернулася до режиму очікування.";
    sendNotification(message);
    writeSystemLog(message);
}

```

```

/* -----
  27. РЕЖИМ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ
  ----- */

void serviceTest() {
  currentMode = MODE_SERVICE;
  systemState.currentMode = MODE_SERVICE;

  Serial.println();
  Serial.println("===== РЕЖИМ ТЕСТУВАННЯ =====");

  readSensors();

  Serial.print("Датчик руху: ");
  Serial.println(motionValue);

  Serial.print("Герконовий датчик: ");
  Serial.println(doorValue);

  Serial.print("Датчик диму: ");
  Serial.println(smokeValue);

  Serial.print("Датчик затоплення: ");
  Serial.println(floodValue);

  Serial.print("Датчик газу: ");
  Serial.println(gasValue);

  Serial.print("Температура: ");
  Serial.println(temperature);

  Serial.print("Вологість: ");
  Serial.println(humidity);

  Serial.println("Перевірка світлового маяка...");
  digitalWrite(LIGHT_PIN, HIGH);
  delay(1000);
  digitalWrite(LIGHT_PIN, LOW);

  Serial.println("Перевірка сирени...");
  digitalWrite(SIREN_PIN, HIGH);
  delay(1000);
  digitalWrite(SIREN_PIN, LOW);
  Serial.println("Тестування завершено");
  Serial.println("=====");
  Serial.println();
  currentMode = MODE_WAITING;
  systemState.currentMode = MODE_WAITING;
}

```