МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ВОЇНСЬКИЙ Юрій Володимирович

Інтелектуальна інформаційна система керування оповіщенням в освітніх установах / Intelligent Information System for Alert Control in Educational Institutions

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Штучний інтелект

Кваліфікаційна робота

Виконав студент групи КНШІ-41
Ю. В. Воїнський

_____

Науковий керівник:
к.т.н., доцент, В. С. Коваль

_____

Кваліфікаційну роботу допущено до захисту
«___»_____ 2026 р.

Завідувач кафедри
_____ Н.В. Дзюбановська

ТЕРНОПІЛЬ – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Ступінь вищої освіти «бакалавр»
спеціальність 122 – «Комп'ютерні науки»
освітньо-професійна програма –«Штучний інтелект»

«ЗАТВЕРДЖУЮ»
В.о. завідувача кафедри
_____ Н.В. Дзюбановська
«____» _____20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Воїнському Юрію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Інтелектуальна інформаційна система керування оповіщенням в освітніх установах / Intelligent Information System for Alert Control in Educational Institutions

керівник роботи В.С. Коваль, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджений наказом по університету від 01 грудня 2025 року № 894.
2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.
3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
– проаналізувати вимоги цивільного захисту в закладах освіти та існуючі аналоги систем оповіщення;
– спроєктувати дворівневу архітектуру та алгоритми асинхронного збору, обробки й пріоритезації подій безпеки;
– розробити інтелектуальний модуль семантичного аналізу текстових розпоряджень адміністрації на основі мовних моделей (LLM);
– спроєктувати інформаційне забезпечення комплексу: реляційну базу даних SQLite та UART-протокол зв'язку;
– реалізувати програмно-апаратне забезпечення (Python-сервер, Telegram-бот, C++ прошивка) та дослідити характеристики латентності й автономності системи.

5. Перелік графічного матеріалу в роботі:
– блок-схема алгоритму ієрархічної обробки тригерів та пріоритезації подій;
– схема структурно-функціональної архітектури взаємодії програмно-апаратних та хмарних засобів.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Огляд літературних джерел відповідно до предметної області та складання плану роботи	до 01.01.2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02.2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05.2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2026 р.	

Студент_____ Ю.В. Воїнський
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи_____ В.С. Коваль
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи: 58 с., 9 рис., 1 додаток, 41 джерел.

Об’єктом дослідження є процеси автоматизації моніторингу параметрів фізичного середовища, інтелектуального аналізу даних та оперативного сповіщення персоналу й здобувачів освіти в освітньому закладі.

Метою роботи є проєктування, програмно-алгоритмічна реалізація інтелектуальної системи керування оповіщеннями, здатної в асинхронному режимі обробляти різнорідні тригери та адаптивно управляти виконавчими пристроями.

Методи дослідження: аналіз та синтез, методи штучного інтелекту, методи розробки програмного забезпечення.

Результатом роботи є дворівневий програмно-апаратний комплекс, який включає центральний асинхронний Python-сервер на базі мікрокомп’ютера Raspberry Pi, локальну базу даних SQLite, модуль інтелектуальної класифікації інтентів через Telegram Bot API (LLM) та виконавчий уровень на базі мікроконтролера Arduino з низькорівневою C++ прошивкою. Розроблено алгоритм ієрархічної пріоритезації подій з функцією автоматичного блокування побутових сигналів розкладу при виникненні безпекових загроз, а також спроєктовано контур автономного живлення на LiFePO4 акумуляторах.

Практична цінність роботи полягає у створенні автономної та інтелектуальної системи цивільного захисту, яка може бути легко масштабована та впроваджена у закладах освіти будь-якого рівня для мінімізації людського фактора в кризових ситуаціях.

Ключові слова: ІНТЕЛЕКТУАЛЬНА СИСТЕМА, СИСТЕМА ОПОВІЩЕННЯ, ARDUINO, RASPBERRY PI, ВЕЛИКІ МОВНІ МОДЕЛІ, КЛАСИФІКАЦІЯ ОБРАЗІВ, TELEGRAM-БОТ, ЦИВІЛЬНИЙ ЗАХИСТ, АВТОНОМНЕ ЖИВЛЕННЯ, ПОСЛІДОВНИЙ ІНТЕРФЕЙС.

ANNOTATION

Explanatory note to the qualification work: 58 pages, 9 figures, 1 appendice, 41 references.

The object of the study is the processes of automating physical environment parameters monitoring, data mining, and prompt notification of personnel and students in an educational institution.

The purpose of the work is the design, software, and algorithmic implementation of an intelligent notification management system capable of processing heterogeneous triggers in an asynchronous mode and adaptively controlling executive devices.

Research methods: analysis and synthesis, artificial intelligence methods, software engineering methodologies.

The result of the work is a two-level software and hardware complex, which includes a central asynchronous Python server based on the Raspberry Pi single-board computer, a local SQLite database, an intelligent intent classification module via the Telegram Bot API (LLM), and an executive level based on the Arduino microcontroller with low-level C++ firmware. An algorithm for hierarchical prioritization of events with the function of automatic suppression of routine schedule signals in case of safety threats has been developed, and an autonomous power supply circuit based on LiFePO4 batteries has been designed.

The practical value of the work lies in the creation of an autonomous and intelligent civil protection system that can be easily scaled and implemented in educational institutions of any level to minimize the human factor in crisis situations.

Keywords: INTELLIGENT SYSTEM, NOTIFICATION SYSTEM, ARDUINO, RASPBERRY PI, LARGE LANGUAGE MODELS, PATTERN CLASSIFICATION, TELEGRAM BOT, CIVIL PROTECTION, AUTONOMOUS POWER SUPPLY, SERIAL INTERFACE.

ЗМІСТ

ВСТУП

Сучасний етап розвитку освітньої інфраструктури в Україні вимагає докорінної модернізації та інтелектуалізації систем внутрішньої безпеки, цивільного захисту та оперативного інформування. Традиційні засоби оповіщення, що десятиліттями базувалися на аналогових таймерах, ручному керуванні дзвінками або застарілих дротових радіомережах, демонструють повну неефективність перед обличчям сучасних комбінованих загроз. Вони не здатні забезпечити адекватну швидкість реакції, є вразливими до людського фактора та абсолютно ізольованими від зовнішнього безпекового середовища.

В умовах воєнного стану та постійних загроз виникнення надзвичайних ситуацій критично важливим аспектом функціонування будь-якого закладу освіти є миттєва трансляція сигналів повітряної тривоги, оперативне сповіщення про загрози техногенного та природного характеру, а також безперервний локальний моніторинг параметрів середовища. Зокрема, життєво необхідним є виявлення накопичення монооксиду вуглецю в підвальних укриттях або аварійних протікань інженерних мереж. Одночасно з цим постає потреба в автоматизації повсякденної діяльності закладу, включаючи точну подачу дзвінків за навчальним розкладом з можливістю його гнучкого коригування у разі форс-мажорних обставин.

Вирішення цих науково-практичних завдань лежить у площині проєктування комплексних інтелектуальних систем, які поєднують низькорівневі апаратні мікроконтролерні платформи (такі як Arduino) для прямої взаємодії з фізичним світом із високорівневими обчислювальними серверами (на базі Raspberry Pi), інтегрованими з хмарними сервісами та великими мовними моделями (LLM). Використання штучного інтелекту дозволяє реалізувати когнітивний аналіз неструктурованих текстових повідомлень, що надходять від адміністрації через популярні комунікаційні канали (наприклад, Telegram), трансформуючи природну мову в детерміновані керуючі сигналі для виконавчих пристроїв. Це дозволяє усунути затримки, зумовлені людським фактором, забезпечити безперебійність

функціонування системи в умовах відключень електроенергії та сформувати безпечний освітній простір.

Метою роботи є проєктування, теоретичне обґрунтування та алгоритмічне представлення комплексної інтелектуальної системи керування оповіщеннями в освітньому закладі, здатної в автоматичному асинхронному режимі обробляти різнорідні фізичні й цифрові тригери та здійснювати оперативне інформування з урахуванням вимог автономності.

Об’єкт дослідження – процеси автоматизації систем сповіщення в освітніх закладах.

Предмет дослідження – архітектурні рішення, алгоритми ієрархічної пріоритезації подій, методи семантичної класифікації неструктурованих повідомлень великими мовними моделями, структури реляційних даних розкладу та програмно-технічні засоби створення систем оповіщення.

Результати кваліфікаційної роботи апробовані та опубліковані у матеріалах IV Міжнародної науково-практичної конференції "Information Control Systems & Project Management" (ICSPM-2026), копії матеріалів апробації наведено у додатку А.

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області цивільного захисту в навчальних закладах

Цивільний захист в освітньому середовищі є складною багаторівневою системою, що охоплює заходи з запобігання надзвичайним ситуаціям, захисту здобувачів освіти та персоналу, а також ліквідації наслідків у разі виникнення небезпечних подій [5]. В умовах сучасної безпекової ситуації в Україні пріоритетність функціонування таких систем значно зросла, а вимоги до їхньої оперативності, надійності та інтелектуальності стали критичними. Освітній заклад як об’єкт цивільного захисту має специфічні характеристики, що визначають архітектуру систем сповіщення, зокрема високу щільність перебування людей на обмеженій території, що вимагає забезпечення максимальної швидкості доведення сигналу тривоги, а також необхідність інтеграції безпекових функцій із повсякденними адміністративними процесами, такими як навчальний розклад.

Потенційні загрози в освітньому середовищі можна класифікувати за їхньою природою та характером впливу, причому ключове місце займають воєнні загрози, такі як повітряні тривоги, ракетна небезпека та загроза застосування БПЛА. Ці події характеризуються екстремальною динамікою, що вимагає миттєвої реакції системи від отримання зовнішнього сигналу через API державних сервісів до активації звукових та візуальних засобів оповіщення. Окрім військових ризиків, важливе значення мають техногенні та аварійні ситуації, наприклад, пожежі, витоки чадного газу в підвальних приміщеннях, що використовуються як укриття, або аварії на інженерних мережах. Моніторинг цих показників вимагає наявності апаратного рівня сенсорів [29], здатних працювати в автономному режимі у складі автоматизованих систем цивільного захисту [16], а в окремих випадках і кризові ситуації соціального характеру, такі як загрози мінування, що вимагають наявності каналів оперативного керування, доступних для адміністрації закладу.

Аналіз існуючих систем оповіщення та практичних аспектів організації безпеки в закладах освіти [8] показує, що значна частина застарілих рішень базується на аналогових мережах або ручному керуванні, що є головним джерелом

затримок. Традиційні системи часто є ізольованими: датчики пожежі не пов'язані з динаміками сповіщення, а розклад дзвінків функціонує незалежно від безпекового контуру. Така фрагментація призводить до того, що в критичних ситуаціях персонал змушений виконувати багато дій вручну, що збільшує ризик помилок через стрес. Більше того, відсутність інтелектуального аналізу вхідних сигналів робить такі системи вразливими до хибних спрацювань та нездатними адаптуватися до зміни ситуації в режимі реального часу.

Аналіз існуючих систем оповіщення та практичних аспектів організації безпеки в закладах освіти показує, що значна частина експлуатованих рішень базується на застарілих аналогових мережах або локальних пультах ручного керування, які є головним джерелом критичних часових затримок. Традиційні технічні засоби найчастіше функціонують як ізольовані, фрагментовані інфраструктурні одиниці: автоматичні сповіщувачі пожежной сигналізації не мають прямих апаратних зв'язків із загальношкільною трансляційною акустикою, а цифрові чи механічні таймери подачі дзвінків працюють автономно, повністю відокремлено від безпекового контуру будівлі. Практика використання таких систем в освітньому просторі виявляє їхню тотальну нездатність до каскадного реагування: при виникнення загрозі персонал змушений виконувати довгі послідовності ручних комутацій у стресовому стані, що стрімко збільшує ймовірність помилок через людський фактор. Додатковою деструктивною ознакою наявних рішень є відсутність інтелектуальних модулів динамічної фільтрації вхідних сигналів, що робить безпекові контури чутливими до хибних спрацювань і позбавляє їх можливості адаптувати логіку роботи під мінливі параметри середовища в режимі реального часу. Більшість промислових систем, що встановлюються за типовими проєктами, діють за лінійним пасивним принципом «тригер - сирена», не забезпечуючи інформативності сповіщення (наприклад, трансляції чітких мовних інструкцій щодо напрямку евакуації залежно від локалізації осередку задимлення чи витоку токсичних газів).

Нормативно-правова база діяльності у сфері цивільного захисту визначає алгоритми оповіщення населення [11], тому завданням для інтелектуальних систем

є не лише відповідність цим нормам, а й їхнє розширення за рахунок автоматизації. Сучасні вимоги наголошують на необхідності дублюванні каналів передачі сигналів [4], забезпеченні автономності систем у разі знеструмлення та наявності чітких сценаріїв реагування. Інтелектуальна інформаційна система керування оповіщеннями в даному контексті розглядається як інструмент трансформації пасивного сповіщення у проактивне управління, що означає перехід від застарілих систем до таких, де логіка оповіщення є контекстно-залежною.

1.2 Огляд та аналіз існуючих рішень і технологій оповіщення

Аналіз існуючого ринку засобів оповіщення свідчить про глибоку диференціацію технологій - від застарілих аналогових радіовузлів до сучасних мережевих систем на базі IP-протоколів. У контексті навчальних закладів вибір системи оповіщення визначається балансом між надійністю, вартістю впровадження та гнучкістю інтеграції з існуючою IT-інфраструктурою. Більшість комерційних рішень, що представлені на ринку, можна розділити на кілька ключових сегментів залежно від принципу їхнього функціонування та способу передачі сигналу.

Перший сегмент охоплює традиційні провідні системи, що базуються на використанні централізованих підсилювачів та дротових ліній зв’язку. Такі системи характеризуються високою стійкістю до зовнішніх електромагнітних завад та передбачуваною якістю звуку, однак вони мають значні обмеження в частині адаптивності. Модернізація такої системи для виконання нових завдань, наприклад, інтеграції з хмарними сервісами повітряної тривоги, вимагає заміни значної частини апаратної бази або впровадження додаткових перехідних модулів для переходу на цифрові комунікаційні платформи [14], що суттєво підвищує загальну вартість володіння. Крім того, централізовані системи мають єдину точку відмови: вихід з ладу підсилювача призводить до повної зупинки роботи оповіщення в межах усього об’єкта.

Другий сегмент становлять IP-базовані системи оповіщення, які використовують локальну мережу закладу для передачі аудіопотоків та керуючих сигналів. Ці рішення пропонують високий рівень гнучкості, дозволяючи сегментувати зони оповіщення та автоматизувати розклад дзвінків через програмні інтерфейси. Разом з тим, впровадження таких систем в умовах освітніх закладів часто стикається з проблемою залежності від стабільності локальної мережі (LAN). У разі відключення електроенергії та переходу на автономне живлення, підтримка активного мережевого обладнання (комутаторів, маршрутизаторів) стає критичним завданням, що потребує значних витрат на системи безперебійного живлення (UPS). Дослідження архітектури та оцінки живучості таких розподілених IP-систем масового інформування [2] показують, що закриті архітектури багатьох комерційних IP-систем ускладнюють інтеграцію з нестандартними безпековими датчиками, наприклад, сенсорами рівня газу або спеціалізованими тривожними кнопками.

Окрему категорію складають мобільні та гібридні рішення, які використовують бездротові канали зв’язку або поєднують різні середовища передачі даних. Такі системи приваблюють своєю легкістю монтажу, оскільки не потребують прокладання багатокілометрових кабельних трас. Однак для великих навчальних корпусів з товстими стінами та підвальними приміщеннями використання бездротових технологій часто супроводжується проблемами з рівнем сигналу та затримками передачі даних. Більше того, більшість існуючих систем, представлених на ринку, є пасивними - вони лише транслюють заздалегідь записані повідомлення або сигнали, не аналізуючи контекст події та не адаптуючись до нього автоматично.

Аналіз інтелектуальних можливостей сучасних систем управління показує, що головним системним обмеженням комерційних рішень є повна відсутність когнітивної обробки інформаційного контексту вхідних сигналів. Пасивні алгоритми не здатні самостійно ідентифікувати семантичні відтінки розпоряджень, виокремлювати критичні інтенти в умовах зашумлених текстових каналів або динамічно перераховувати пріоритети черги сповіщень. Інтелектуалізація

архітектури, що досягається шляхом інтеграції великих мовних моделей (LLM), забезпечує якісно новий рівень автоматизації безпекових процесів. Можливості інтелектуальних агентів дозволяють здійснювати не просто жорсткий синтаксичний аналіз, а глибоке логічне декодування неструктурованого тексту. Це дає змогу системі гнучко розпізнавати непрямі загрози, валідувати повноваження оператора без залучення складних графічних меню, автоматично підбирати сценарії звукового супроводу під конкретний тип надзвичайної ситуації та в реальному часі координувати роботу виконавчих пристроїв на основі семантичного ядра отриманої команди.

Порівняльний аналіз показує, що більшість комерційних систем оповіщення орієнтовані на виконання однотипних завдань: або трансляцію музики та оголошень (офісні системи), або лише тривожні сигнали (пожежна сигналізація). В освітньому закладі постає потреба в системі, яка б виконувала роль «інтелектуального хаба». Такий хаб повинен поєднувати функції автоматичного управління розкладом з можливістю пріоритезації критичних загроз, таких як повітряна тривога. Сучасні інтелектуальні системи (рисунок 1.1) на базі мікроконтролерів (Arduino) [18] та одноплатних комп'ютерів (Raspberry Pi) [23] пропонують альтернативу, яка є значно дешевшою за промислові аналоги, але при цьому дозволяє реалізувати глибоку програмну кастомізацію [24].

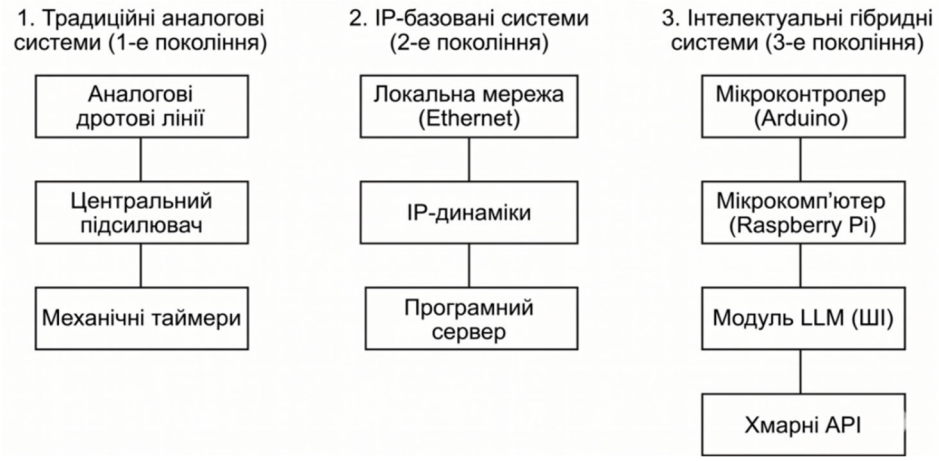

Рисунок 1.1 - Порівняльний аналіз процедур оповіщення

Використання в розробці гібридного підходу, де низькорівнева логіка (Arduino) відповідає за гарантоване спрацювання виконавчих пристроїв, а високорівнева логіка (Raspberry Pi з використанням LLM) забезпечує гнучкий аналіз даних, дозволяє усунути недоліки обох типів існуючих систем. Можливість обробки неструктурованих повідомлень, що надходять через Telegram або інші месенджери, за допомогою великих мовних моделей, відкриває нові перспективи для взаємодії з системою. Користувач може сформулювати запит у довільній формі, а система, провівши семантичний аналіз, самостійно визначить пріоритет дії та виконає необхідні маніпуляції з обладнанням.

Отже, аналіз існуючих рішень виявляє чітку тенденцію до інтелектуалізації безпекових систем. Традиційні засоби поступово витісняються відкритими програмно-апаратними платформами, які дозволяють закладам освіти будувати гнучкі та масштабовані безпекові контури з мінімальними капіталовкладеннями. Ключовим вектором розвитку таких систем стає інтеграція з зовнішніми інформаційними потоками та автоматизація реакції на основі інтелектуальних алгоритмів, що дозволяє мінімізувати вплив людського фактора в умовах кризових ситуацій. Такий підхід робить систему не просто інструментом сповіщення, а повноцінним елементом інфраструктури, що активно взаємодіє з оточенням для забезпечення безпеки учасників освітнього процесу.

1.3 Постановка задачі дослідження

На основі проведеного аналізу предметної області та огляду існуючих рішень, можна констатувати, що актуальним науково-технічним завданням є створення інтелектуального програмно-апаратного комплексу, здатного до автономного функціонування в умовах обмежених ресурсів та забезпечення високого рівня безпеки. Важливим акцентом уваги у цьому напрямі є необхідність у забезпеченні надійної роботи в реальному часі (що вимагає низькорівневого контролю), а також в інтелектуальній інтерпретації команд, що надходять через

високорівневі канали зв’язку на базі вбудованих IoT-рішень та одноплатних обчислювальних платформ [13].

Метою роботи є проєктування, теоретичне обґрунтування та алгоритмічне представлення інтелектуальної системи керування оповіщеннями в освітньому закладі, здатної в асинхронному режимі обробляти різнорідні фізичні й цифрові тригери та здійснювати оперативне інформування з урахуванням вимог автономності. Для реалізації поставленої мети та візуалізації базових принципів побудови комплексу запропоновано загальну концептуальну схему дворівневої архітектури інтелектуальної системи оповіщення (рисунок 1.2).

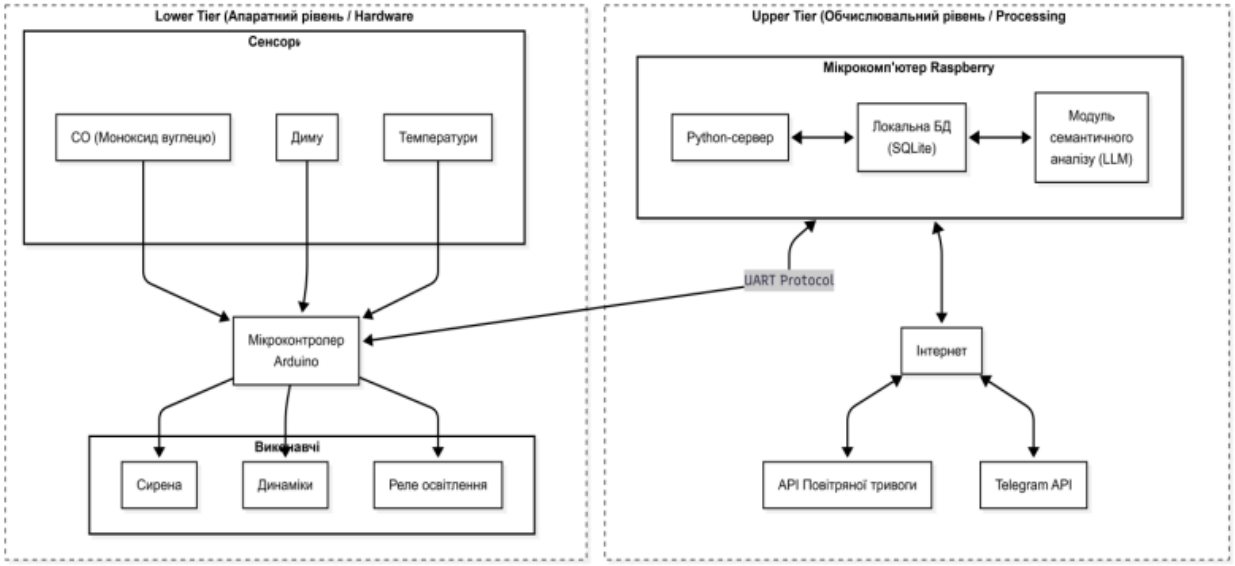

Рисунок 1.2 – Концептуальна схема взаємодії рівнів інтелектуальної системи оповіщення

Як показано на наведеній схемі, передбачається архітектурне рішення, що розділяється на чіткі ієрархічні рівні, де нижній (апаратний) рівень повністю ізолює сенсорно-виконавчі функції реального часу, а верхній (обчислювальний) - відповідає за інтелектуальну обробку контексту інформаційних потоків та семантичний аналіз неструктурованих повідомлень.

Передбачається наступна логіка дослідження, що розпочинається із системного аналізу умов цивільного захисту та безпеки в освітніх установах. Такий аналіз дозволить формалізувати вимоги до автоматизації повсякденних процесів та

виявити критичні точки взаємодії користувачів із подібними системами. На основі цього передбачається виконати критичний огляд існуючих технічних рішень та індустріальних технологій оповіщення, визначено їхні системні обмеження у контексті сучасної безпекової ситуації в Україні та детально обґрунтовано доцільність розробки інтегрованого інтелектуального комплексу. Отримані аналітичні висновки дозволять сформулювати чітку науково-технічну постановку задачі проєктування програмно-апаратних засобів керування сповіщеннями на базі гібридної архітектури.

На наступному етапі передбачається здіснити інженерне проєктуваня структурно-функціональної архітектури системи іх детальним описом принципів поєднання мікроконтролера Arduino та одноплатного комп'ютера Raspberry Pi. а також теоретичним обґрунтованням контуру автономного живлення на базі літій-залізо-фосфатних акумуляторів для забезпечення живучості системи під час тривалих знеструмлень об'єкта.

Математико-алгоритмічний базис системи передбачається реалізувати через створення алгоритмічного забезпечення асинхронної обробки різнорідних вхідних тригерів. Запропоновані алгоритми вирішать задачу ієрархічної пріоритезації подій з функцією примусового блокування рутинного навчального розкладу під час виникнення кризових ситуацій. Паралельно з цим обґрунтованню підлягає концепція інтелектуального модуля семантичного аналізу неструктурованих текстових повідомлень у месенджері Telegram з використанням великих мовних моделей та методів Prompt Engineering, що мінімізує людський фактор при координуванні дій адміністрації.

Для забезпечення інформаційної цілісності комплексу передбачається побудова структури локального інформаційного забезпечення. Воно включатиме нормалізовану реляційну модельбази даних SQLite для ведення розкладу занять та логування подій, а також низькорівневу структуру бінарних пакетів передачі даних через послідовний інтерфейс UART. Теоретично запропоновані архітектурні рішення підкріплюються представленням програмної реалізації логіки

функціонування системи у вигляді асинхронного Python-сервера верхнього рівня та керуючої прошивки мікроконтролера на мові C++.

На завершальній стадії дослідження передбачається проведення тестування та експериментальну оцінку часової латентності обробки локальних апаратних сенсорів, процесів опитування зовнішніх веб-сервісів повітряної тривоги та семантичного аналізу текстів мовною моделлю. Підсумковим кроком є проведення ретельного аналізу потенційних інженерних помилок, системних обмежень та формулювання детальних практичних рекомендацій щодо впровадження, експлуатації й подальшого технологічного розвитку спроєктованих рішень у реальних інфраструктурних проєктах навчальних закладів.

Окреслене коло взаємопов'язаних етапів дослідження повністю охоплює всі аспекти функціонування системи - від фізичного рівня взаємодії з оточенням до абстрактного рівня прийняття рішень за допомогою методів штучного інтелекту. Такий комплексний підхід дозволить сформувати цілісну теоретичну базу для створення нового покоління засобів оповіщення, адаптованих до специфіки сучасного освітнього простору.

Для досягнення поставленої мети в межах кваліфікаційної роботи передбачається вирішення комплексу взаємопов'язаних завдань:

– проаналізувати вимоги цивільного захисту в закладах освіти та існуючі аналоги систем оповіщення;

– спроєктувати дворівневу архітектуру та алгоритми асинхронного збору, обробки й пріоритезації подій безпеки;

– розробити інтелектуальний модуль семантичного аналізу текстових розпоряджень адміністрації на основі мовних моделей (LLM);

– спроєктувати інформаційне забезпечення комплексу: реляційну базу даних SQLite та UART-протокол зв'язку;

– реалізувати програмно-апаратне забезпечення (Python-сервер, Telegram-бот, C++ прошивка) та дослідити характеристики латентності й автономності системи.

2 ПРОЄКТУВАННЯ ТА АЛГОРИТМІЧНЕ ПРЕДСТАВЛЕННЯ СИСТЕМИ

2.1 Системна архітектура взаємодії апаратних та програмних засобів

Проєктування архітектури інтелектуальної системи оповіщення базується на фундаментальних принципах модульності, дворівневої ієрархічної побудови та асинхронності обробки потоків даних. Такий підхід дозволяє ефективно розмежувати вузькоспеціалізовані функції збору даних та безпосереднього фізичного керування виконавчими пристроями, що покладені на низькорівневий апаратний рівень, від складних алгоритмів інтелектуального аналізу, обробки запитів та інтеграції з розподіленими хмарними сервісами, які виконуються на високорівневому обчислювальному модулі [35]. Поділ системи на такі функціональні рівні забезпечує її високу автономність, оскільки критично важливі безпекові функції можуть продовжувати виконуватися навіть за умови тимчасового розриву зв’язку між модулями або виходу з ладу окремих програмних компонентів верхнього рівня.

Для забезпечення максимальної стійкості системи в умовах нестабільного енергоживлення та високої відповідальності за безпекові процеси було обрано гібридну архітектурну структуру. Взаємодія між фізичним рівнем, де розташовані різноманітні сенсори стану середовища та пристрої виводу звукових або візуальних сигналів, та обчислювальним модулем здійснюється за допомогою послідовного інтерфейсу UART, який довів свою ефективність як надійний інструмент детермінованого обміну даними з мінімальною латентністю серед інших послідовних інтерфейсів. Крім того, даний інтерфейс характеризується низькими апаратними вимогами та високою стабільністю роботи в реальних умовах експлуатації.

Візуалізація структурно-функціональної взаємодії ключових компонентів системи представлена на рисунку 2.1, де відображено логічний зв’язок між апаратними компонентами, сервером обробки та зовнішніми інформаційними каналами.

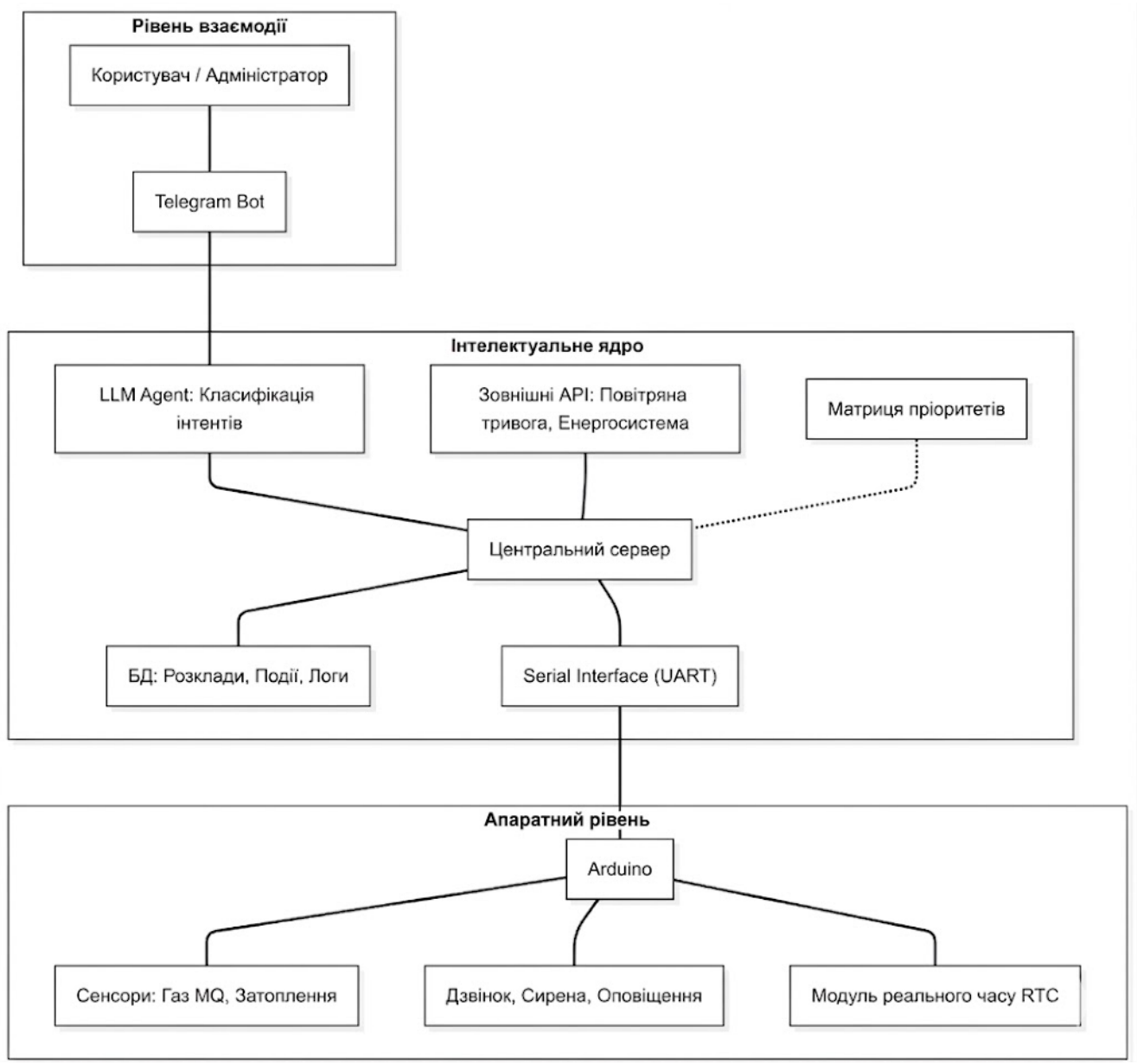

Рисунок 2.1 - Запропонована структурно-функціональна архітектура взаємодії апаратних та програмних засобів

Нижній апаратний рівень побудований на базі мікроконтролера Arduino, програмування якого забезпечує стабільну взаємодію з фізичними компонентами , і який виконує роль надійного інструменту взаємодії з фізичним світом у реальному часі. Використання мікроконтролера дозволяє забезпечити стабільний збір даних з фізичних датчиків концентрації чадного газу, рівня задимленості, температури та тривожних кнопок [7]. Ключовою перевагою платформи Arduino є її здатність до безперебійної роботи з аналоговими та цифровими сигналами, а також можливість створення стійкої до програмних збоїв прошивки, написаної на мові C++ [27]. Цей

рівень є повністю автономним у прийнятті рішень щодо активації виконавчих пристроїв, таких як сирени, динаміки чи реле аварійного вимкнення інженерних систем, що критично важливо у разі виникнення загрози, коли затримка в кілька секунд може мати вирішальне значення.

Верхній обчислювально-керуючий рівень реалізовані на базі мікрокомп’ютера Raspberry Pi, який виступає в ролі інтелектуального центру управління. На цьому рівні функціонує асинхронний Python-сервер, який відповідає за координацію роботи всіх вузлів системи та забезпечує зв’язок із зовнішніми безпековими сервісами через глобальну мережу. Тут також розгорнуто модуль інтелектуального аналізу, який використовує великі мовні моделі (LLM) через інтерфейси API для глибокої інтерпретації довільних текстових повідомлень від адміністрації закладу. Окрім цього, Raspberry Pi підтримує локальне сховище даних на базі реляційної СУБД SQLite [32], де зберігається розклад навчальних занять та системні логи, що дозволяє системі зберігати функціональність навіть за умови повної втрати зв'язку із хмарними ресурсами. Комунікація між мікрокомп’ютером та мікроконтролером базується на розробленому бінарному протоколі передачі даних через UART, що є значно ефективнішим за текстові запити, оскільки дозволяє мінімізувати обсяг переданої інформації та час, необхідний на її обробку.

Окремим аспектом архітектури є забезпечення живучості системи завдяки інтеграції контуру автономного живлення на базі акумуляторів LiFePO4 [39], які мають вищу циклічну стійкість та безпечність [37] порівняно з іншими типами накопичувачів [25]. Стабілізатори напруги, інтегровані безпосередньо у фізичний рівень [15], гарантують, що критичні вузли системи продовжуватимуть роботу навіть при тривалому знеструмленні будівлі. Така архітектурна стратегія дозволяє поєднати надійність класичних апаратних рішень із гнучкістю та аналітичною потужністю сучасних алгоритмів штучного інтелекту, що забезпечує не лише миттєве сповіщення про загрози, а й проактивне керування безпековим простором освітнього закладу. Спроектована система демонструє високий рівень адаптивності до змін зовнішнього середовища, мінімізуючи вплив людського фактора завдяки

здатності автоматично пріоритезувати критичні сигнали тривоги над рутинними завданнями закладу.

2.2 Алгоритмічне забезпечення обробки тригерів та пріоритезації

Ефективність функціонування інтелектуальної системи оповіщення в умовах критичних ситуацій безпосередньо залежить від стійкості та швидкодії алгоритмів, що відповідають за обробку різнорідних вхідних подій та визначення їхнього пріоритету в динамічному середовищі. В умовах освітнього закладу система повинна функціонувати як конвергентна платформа, здатна одночасно обробляти рутинні запити, такі як сигнали розкладу занять, та екстрені події, наприклад, повітряні тривоги, загрози техногенного характеру або спрацювання датчиків загазованості. Для розв'язання цього завдання було спроєктовано багаторівневий ієрархічний алгоритм обробки тригерів, який реалізує принцип витіснення менш пріоритетних завдань більш критичними подіями, гарантуючи при цьому живучість системи та стабільність її реакції навіть у випадках часткового виходу з ладу окремих вузлів.

Фундаментом алгоритмічного забезпечення є кооперативна багатозадачність, що реалізована на рівні програмного сервера під управлінням Python за допомогою засобів асинхронної обробки подій [20]. Вхідні дані від усіх джерел класифікуються за рівнем безпекової загрози, де кожна подія отримує індивідуальний ваговий коефіцієнт та категорію пріоритетності. Система підтримує динамічний каталог станів, де найвищий рівень пріоритету закріплено за сигналами повітряної тривоги та даними з датчиків критичних концентрацій небезпечних газів у підвальних приміщеннях. При отриманні події такого класу система миттєво ініціює переривання, призупиняючи або повністю блокуючи всі поточні фонові процеси, включно з планувальником розкладу дзвінків, за рахунок використання неблокуючих операцій введення-виведення. Такий механізм забезпечує мінімальну латентність між моментом фіксації загрози сенсором або

отриманням зовнішнього сигналу та активацією виконавчого контуру, мінімізуючи часові затримки, зумовлені черговістю обробки пакетів даних.

Для наочного представлення логіки прийняття рішень та механізму примусового переривання рутинних завдань, на рисунку 2.2 наведено схему алгоритму ієрархічної обробки тригерів та пріоритезації подій.

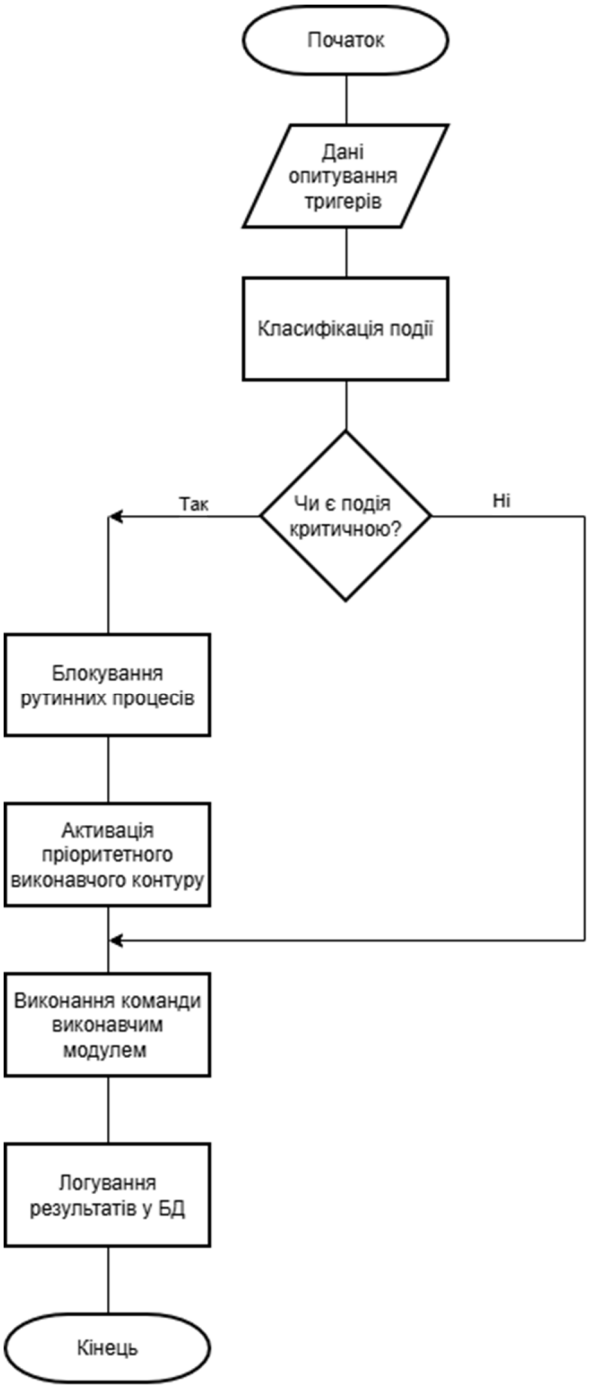

Рисунок 2.2 - Схема алгоритму ієрархічної обробки тригерів та пріоритезації подій

Алгоритмічна логіка обробки даних використовує класичні шаблони організації корпоративних додатків та обробки повідомлень [21] і базується на дворівневому механізмі черги. повідомлень, де кожна подія, незалежно від джерела її виникнення, проходить обов’язковий етап первинної фільтрації та валідації. Для мінімізації впливу хибних спрацювань, які особливо характерні для бюджетних апаратних датчиків, застосовуються математичні методи цифрової фільтрації сигналів, зокрема алгоритм експоненційного згладжування [34], який широко застосовується для цифрової фільтрації сигналів з низьковартісних апаратних сенсорів [17]. Цей метод дозволяє ефективно відсікати короткочасні викиди значень, що не свідчать про реальну загрозу, а є наслідком електричних перешкод або фонових шумів середовища. Після успішного проходження фільтрації подія надходить до центрального модуля пріоритезації, який аналізує поточний стан системи та, залежно від отриманого пріоритету, або додає завдання в чергу виконання, або ініціює процедуру негайного переривання активного процесу через механізм сигналів операційної системи.

Важливим елементом надійності є впровадження логіки «захищеного контуру» між високорівневим сервером (Raspberry Pi) та низькорівневим мікроконтролером (Arduino). Мікроконтролер реалізує алгоритм «сторожового таймера» (Watchdog) для підвищення відмовостійкості обчислювальних засобів нижнього рівня, постійно очікуючи підтвердження від серверної частини про її коректну роботу. У разі відсутності такого підтвердження протягом встановленого часового вікна, мікроконтролер автоматично переходить у режим локальної автономної безпеки. У цьому режимі пріоритезація та обробка виконуються суто на рівні Arduino, де закладена мінімально необхідна логіка реагування на критичні тригери, такі як кнопка «Тривога» або показники датчиків газу. Це архітектурне рішення гарантує, що навіть при виході з ладу обчислювального модуля або розриві зв'язку, система не втратить здатності оповістити персонал про загрозу.

Семантична складова алгоритму пріоритезації суттєво доповнюється інтелектуальним аналізом текстових повідомлень, що надходять від адміністрації закладу через Telegram-бота. Використання великих мовних моделей (LLM) [19]

дозволяє класифікувати намір користувача з високою точністю: система розрізняє запит на планову зміну розкладу від вимоги негайної евакуації. В залежності від результату семантичного аналізу, LLM-модуль формує керуючий сигнал з відповідним рівнем пріоритету, який інтегрується в загальний потік подій системи. Якщо модель класифікує повідомлення як критичне, алгоритм примусово зупиняє виконання всіх планових операцій, незалежно від їхньої тривалості. Така багаторівнева організація обробки даних усуває конфлікти ресурсів та забезпечує детерміновану, передбачувану поведінку системи у всіх сценаріях - від звичайного навчального дня до кризових ситуацій, де швидкість і точність реакції алгоритмів є ключовими чинниками безпеки здобувачів освіти.

2.3 Модуль семантичного аналізу тексту на основі великих мовних моделей

Інтелектуалізація сучасної системи оповіщення освітньої установи вимагає відмови від використання жорстко детермінованих командних інтерфейсів та реалізації адаптивного контуру управління, здатного інтерпретувати запити на природній мові. У межах цієї роботи автором розроблено та алгоритмічно описано спеціалізовану процедуру семантичного аналізу неструктурованих текстових повідомлень, що надходять від адміністрації закладу через інтерфейс месенджера Telegram. Запропонована процедура базується на інтеграції хмарних сервісів великих мовних моделей, які здатні виступати в ролі ефективних когнітивних систем [10], та авторських методів конструювання керуючих запитів, що дозволяє трансформувати довільний текст оператора у структуровані об'єкти керування апаратним рівнем комплексу в умовах підвищеного стресу.

Розроблена автором процедура когнітивного аналізу інформаційного потоку є послідовним багатоетапним процесом обробки даних, що виконується на високорівневому обчислювальному модулі Raspberry Pi в асинхронному режимі. На початковому етапі асинхронного перехоплення та первинної валідації за допомогою розробленого модуля взаємодії з програмним інтерфейсом месенджера сервер перехоплює вхідний пакет на рівні Telegram Bot API [3], що містить

текстову інформацію. На цьому ж кроці виконується обов'язакова перевірка унікального ідентифікатора відправника за локальним білим списком адміністраторів, який зберігається в базі даних SQLite, що дозволяє заблокувати несанкціонований доступ ще на стадії ініціалізації запиту.

На наступному етапі процедури очищений від службових символів текстовий масив передається до інтерфейсу мовної моделі для безпосереднього лінгвістичного аналізу, токенізації та контекстного розбору. Після цього запускається етап класифікації наміру користувача, де модель зіставляє семантичне ядро повідомлення з вектором дозволених системних станів. Авторський вклад на цьому кроці полягає у формалізації та обмеженні простору рішень системи чотирма базовими класами, до яких віднесено екстрену активацію тривоги, модифікацію навчального розкладу, моніторинг поточного статусу заліза та скасування аварійного стану.

Далі процедура забезпечує автоматичну екстракцію іменованих сутностей, вилучаючи просторово-часові параметри, що містяться в повідомленні оператора, такі як номери навчальних корпусів, літери безпекових зон, конкретні часові інтервали або назви академічних груп. Завершеється розроблена процедура етапом трансформації та детермінації вихідних даних, під час якого мовна модель здійснює мапінг виділених лінгвістичних сутностей у строго детермінований, машиночитаний об'єкт формату JSON з автоматичним обчисленням вагового коефіцієнта пріоритету загрози.

Ключовим елементом розробленої процедури, що визначає її стійкість та точність, є авторське проєктування архітектури системного запиту, який виступає в ролі семантичного фільтра-обмежувача логіки штучного інтелекту. На основі розробки спеціалізованих шаблонів керуючих запитів [38], автором було запропоновано та впроваджено жорстку рольову інструкцію для унеможливлення виникнення ефекту галюцинацій мовної моделі та мінімізації латентності генерації токенів. Системний промпт примусово переводить модель у режим ізольованого кінцевого автомата, обмежуючи вихідний потік виключно валідною структурою JSON, що спирається на сучасні архітектурні дослідження та специфікації

провідних промислових моделей [31]. Логічну схему генерації вихідних даних промпту спроєктовано автором таким чином, що за умов отримання неоднозначної чи нерелевантної команди модель детерміновано повертає системний прапорець помилки, який миттєво обробляється сервером без зависання потоку очікування.

Для забезпечення максимального рівня безпеки освітнього закладу та нівелювання ризиків, пов'язаних із можливими мережевими затримками при зверненні до зовнішніх хмарних інтерфейсів, автором розроблено та інтегровано в процедуру контур гібридної експрес-фільтрації. Суть цього інженерного рішення полягає в тому, що перед відправкою тексту на розбір мовною моделлю, на локальному Python-сервері виконується швидкий пре-парсинг рядка на основі регулярних виразів та пошуку критичних слів-маркерів, таких як пожежа, вибух, евакуація або газ.

Якщо локальний алгоритм виявляє хоча б один маркер першого рівня небезпеки, процедура семантичного аналізу оптимізується, і сервер миттєво, минаючи стадію хмарної обробки природної мови, самостійно формирує критичну команду активації тривоги із найвищим пріоритетом. Це дозволяє скоротити час апаратної реакції системи до мілісекунд у випадках явних очевидних загроз. Обчислювальна потужність великої мовної моделі використовується для впровадження методів, що утворюють парадигму адаптивного навчання моделей для задач NLP [26], завдяки чому логіка стає гнучкою при розпізнаванні складних, непрямих розпоряджень та задач повсякденної автоматизації розкладу. Запропонована автором процедура дозволяє повністю ліквідувати людський фактор на етапі фізичного налаштування параметрів сповіщення та перетворює комплекс із пасивного реєстратора подій на проактивне інфраструктурне рішення.

2.4 Обґрунтування метрик оцінювання та вибору архітектурних рішень

Проєктування інтелектуальної системи оповіщення потребує чіткого визначення критеріїв успішності її впровадження, оскільки робота в умовах критичних ситуацій ставить жорсткі вимоги до надійності та часу реакції. Для

обґрунтування вибору архітектурних рішень у даній роботі використано комплексний підхід, що поєднує кількісні технічні метрики та якісні показники адаптивності системи. Вибір технологічного стека (Python, Raspberry Pi, Arduino, SQLite) був продиктований необхідністю балансу між потужністю обчислень, низьким енергоспоживанням та простотою інтеграції з сучасними методами штучного інтелекту.

Однією з ключових метрик оцінювання ефективності розробленого комплексу є загальна латентність реакції системи, яка визначає часовий інтервал між моментом безпосереднього виникнення інформаційного тригера, під яким розуміється подія на апаратному сенсорі або отримання текстового повідомлення від адміністратора, та моментом фізичної активації виконавчого пристрою сповіщення. У контексті завдань цивільного захисту цей показник є критичним і має бути мінімізованим для забезпечення оперативності інформування учасників освітнього процесу. Очікувана латентність обробки локальних апаратних сигналів на рівні мікроконтролера Arduino не повинна перевищувати граничний поріг у межах від 10 до 50 мілісекунд, тоді як для складних інтелектуальних запитів, що проходять через зовнішній хмарний модуль великої мовної моделі, критичним для практичного застосування визначено часовий поріг у діапазоні від 1 до 3 секунд. Кількісне оцінювання та верифікація цього показника здійснюється шляхом проведення серії навантажувальних тестів з точним вимірюванням часу проходження інформаційних сигналів через усі архітектурні вузли системи.

Загальна латентність реакції системи (T_Σ) визначається як сума часових затримок, що виникають на кожному ключовому етапі обробки інформаційного тригера - від моменту фіксації події датчиком чи хмарним сервісом до моменту фізичної активації засобів оповіщення і обчислюється за формулою:

$$T_\Sigma = T_{tr} + T_{proc} + T_{comm} + T_{act}, \tag{2.1}$$

де T_{tr} - час реакції первинного фізичного тригера або отримання сигналу від зовнішнього API;

T_{proc} - латентність програмної обробки даних сервером (включаючи час, необхідний на семантичний аналіз тексту великою мовною моделею);

T_{comm} - час, витрачений на пакетну передачу бінарних кадрів через послідовний апаратний інтерфейс UART;

T_{act} - час безпосередньої фізичної активації та спрацювання виконавчих пристроїв (сирени, реле, шкільних дзвінків).

Другою важливою метрикою оцінювання є надійність та відмовостійкість, що у площині проєктування комплексів цивільного захисту визначає здатність системи продовжувати виконання своїх цільових безпекових функцій або зберігати працездатність у режимі очікування за умов часткової деградації чи виникнення критичних збоїв в окремих інфраструктурних вузлах. Вибір гібридної дворівневої архітектури, у якій виконавчий мікроконтролер Arduino зберігає працездатність і продовжує локальну обробку сигналів від апаратних датчиків середовища без прямої участі серверної частини, дозволяє досягти високих показників живучості. Основним кількісним критерієм успішності в даному контексті виступає коефіцієнт готовності системи, який визначає ймовірність того, що комплекс буде працездатним у довільний момент часу, крім планових періодів технічного обслуговування. Для формалізації цього показника авторами вбудованих систем використовується класичне математичне відношення, згідно з яким коефіцієнт готовності K_g обчислюється за формулою:

$$K_g = \frac{T_o}{T_o + T_v}, \tag{2.2}$$

де T_o - середній час безвідмовної роботи системи між послідовними технічними збоями;

T_v - середній час, необхідний для повного відновлення працездатності комплексу після фіксації відмови.

У спроєктованому рішенні значення коефіцієнта готовності має максимально наближатися до одиниці. Це досягається завдяки апаратному резервуванню ліній енергоживлення на базі літій-залізо-фосфатних акумуляторів, автоматичному асинхронному дублюванню внутрішньої логіки ієрархічної пріоритезації подій, а також інтеграції сторожових таймерів для запобігання програмному зависанню обчислювальних модулів.

Архітектурний вибір на користь модульності, слабкого зв'язку компонентів та застосування класичних патернів проєктування [22] обґрунтовано потребою у масштабованості. та супроводжуваності (maintainability). Python забезпечує легкий доступ до бібліотек роботи з мережевими протоколами, базами даних та AI-фреймворками, що спрощує подальшу модернізацію системи. Використання реляційної бази даних SQLite є оптимальним рішенням для локального збереження розкладу, оскільки вона не потребує налаштування окремого серверного процесу та забезпечує високу швидкість виконання запитів навіть на обмежених ресурсах одноплатного комп’ютера.

Узагальнену структуру критеріїв оцінювання ефективності та їх взаємозв’язок із прийнятими архітектурними рішеннями наведено на рисунку 2.3.

Рисунок 2.3 – Схема критеріїв оцінювання ефективності системи оповіщення

Окрему увагу приділено метриці точності класифікації намірів для розробленого модуля семантичного аналізу текстових повідомлень. Враховуючи високу критичність можливих помилок при розпізнаванні оперативних команд адміністрації навчального закладу на основі семантичного та контекстного аналізу вхідних інформаційних потоків, загальна точність роботи великої мовної моделі повинна складати не менше 95% у всіх тестових сценаріях.

Для проведення оцінювання сформовано збалансований тестовий датасет, який включає 200 прикладів повідомлень оператора, що рівномірно розподілені між базовими класами системних інтентів та містять різноманітні варіанти формулювань у довільній формі природної мови. Для математичного моделювання та кількісного оцінювання цього показника на основі сформованих тестових даних використовується матриця помилок, яка дозволяє детально класифікувати випадки істинних та хибних реакцій системи на надіслані повідомлення. На основі базових елементів цієї матриці загальна точність обчислюється за формулою:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2.3}$$

де TP - кількість істинно позитивних спрацювань, при яких система правильно розпізнала наявний безпековий інтент;

TN - кількість істинно негативних результатів, що відповідають правильному ігноруванню або стандартній обробці рутинних повідомлень, які не містять безпекових загроз;

FP - кількість хибно позитивних спрацювань, коли звичайне повідомлення було помилково інтерпретоване мовною моделлю як критична загроза;

FN - кількість хибно негативних помилок, які є найбільш критичними, оскільки характеризують випадки, коли реальний сигнал про надзвичайну ситуацію не був розпізнаний інтелектуальним модулем.

Застосування методів конструювання керуючих запитів і жорстких рольових інструкцій у системному промпті спрямоване саме на максимізацію цього показника шляхом математичного обмеження контекстного простору моделі та мінімізації значень хибних рішень у знаменнику наведеного виразу.

Для проведення об'єктивного аналізу спроєктованих рішень кількісне оцінювання критеріїв часової та функціональної ефективності комплексу формалізується за допомогою розрахунку загальної латентності, показників надійності та точності розпізнавання, що відображають збалансованість апаратного та програмного рівнів системи.

Вибір архітектурних рішень, таким чином, не є випадковим, а повністю ґрунтується на інженерній концепції мінімально достатньої складності. Кожен обчислювальний та виконавчий вузол системи має чітко визначену роль, що запобігає надмірному розширенню програмного коду, оптимізує використання ресурсів оперативної пам'яті платформ та суттєво зменшує кількість потенційних точок відмови. У підсумку, запропонований вибір метрик та базових архітектурних принципів дозволяє об’єктивно оцінити ефективність системи не лише на етапі імітаційного моделювання, а й при її безпосередньому практичному впровадженні у реальних інфраструктурних проєктах безпеки та цивільного захисту закладів освіти.

3 РЕАЛІЗАЦІЯ СИСТЕМИ ТА ОЦІНКА ЕФЕКТИВНОСТІ

3.1 Опис інформаційного забезпечення та логічної структури збереження даних

Проєктування інформаційного забезпечення інтелектуальної системи оповіщення освітнього закладу є фундаментальним етапом, який безпосередньо визначає швидкість доступу до даних, цілісність інформаційних потоків та загальну стійкість комплексу до критичних збоїв у реальних умовах експлуатації. Специфіка функціонування розробленого архітектурного рішення вимагає ефективного збереження та обробки як статичних чи слабкозмінних даних, до яких належить тижневий розклад навчальних занять і конфігурації обладнання, так і динамічних інформаційних пакетів, що включають черги повідомлень, журнали системних логів, ідентифікатори користувачів та поточні безпекові стани фізичного оточення. Враховуючи обчислювальні й апаратні обмеження одноплатної платформи Raspberry Pi, а також необхідність забезпечення повної автономності системи за умови раптової відсутності підключення до мережі Інтернет чи повного блекауту, як базову систему керування базами даних було обґрунтовано й обрано реляційну СУБД SQLite. Цей вибір зумовлений тим, що SQLite функціонує в межах одного локального безпроцесного файлу, безпосередньо інтегрується в програмний код на мові Python за допомогою стандартного пакета sqlite3, мінімізує навантаження на оперативну пам'ять мікрокомп'ютера та забезпечує повну підтримку транзакцій на відповідність вимогам ACID, що є критично важливим для унеможливлення пошкодження безпекових конфігурацій при аварійному вимкненні електроживлення.

Логічна структура бази даних спроєктована на основі принципів третьої нормальної форми для мінімізації надликовості збереженої інформації, усунення аномалій модифікації та оптимізації швидкості виконання пошукових запитів планувальника розкладу в режимі реального часу. Головною сутністю реляційної схеми є таблиця навчального розкладу, яка зв'язана зовнішніми ключами із сутностями часових інтервалів пар, академічних груп та кастомних подій

інформування персоналу закладу. Кожен запис у цій таблиці містить унікальний ідентифікатор, чіткі часові мітки початку та завершення активності, текстове поле з описом оголошення чи унікальний системний шлях до відповідного аудіофайлу, а також прапорець ієрархічної пріоритетності, який використовується внутрішнім алгоритмом сервера для витіснення рутинних завдань у разі виникнення надзвичайних ситуацій. Для забезпечення цілісності зв'язків між таблицями при видаленні або оновленні записів на рівні бази даних активовано каскадні операції, що запобігає появі втрачених або некоректних посилань у структурі планувальника. Концептуальний взаємозв'язок між виділеними сутностями навчального розкладу та логічними зв'язками на рівні абстрактних типів даних представлено у відповідних специфікаціях проєкту, тоді як ієрархічну структуру та взаємозв'язки компонентів оцінювання ефективності системи відображено на логіко-компонентній схемі (рисунок 3.1).

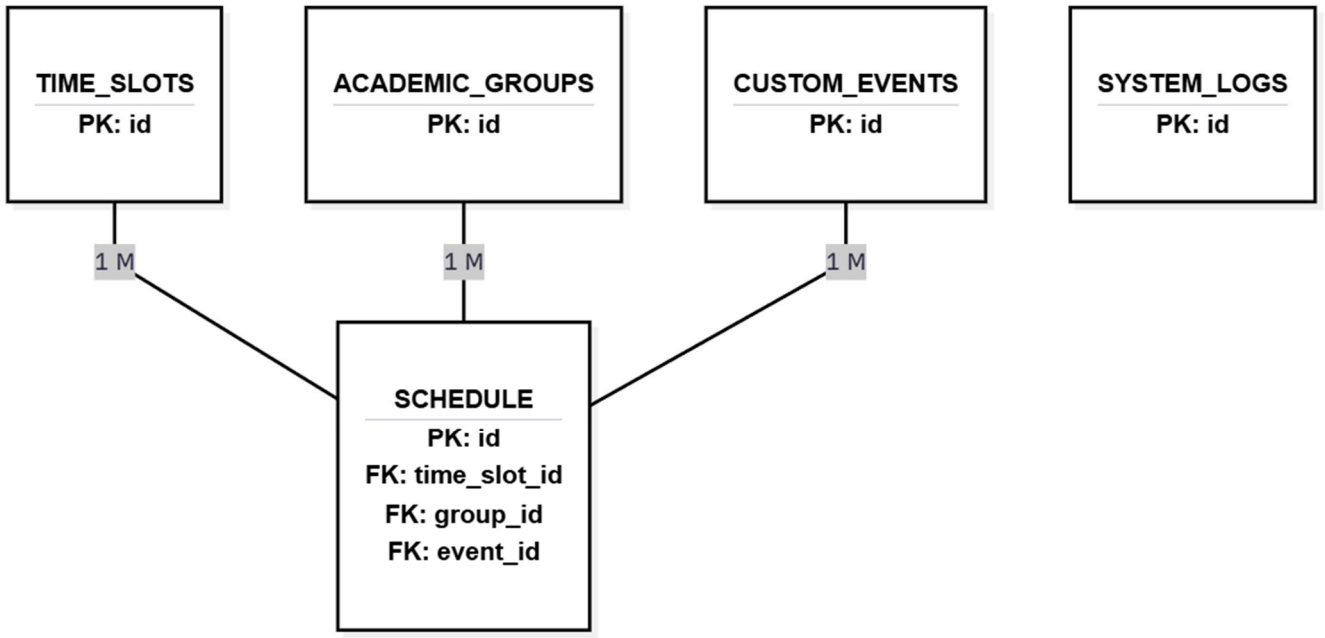

Рисунок 3.1 – Логічна модель структури бази даних системи

Окремий логічний простір у структурі бази даних виділено під таблицю системного журналу подій, куди в асинхронному режимі записуються часові мітки активації фізичних сенсорів, отримані через UART пакети даних, результати

семантичного аналізу великої мовної моделі та унікальні ідентифікатори адміністраторів, які надсилали керуючі команди через інтерфейс Telegram-бота. Завдяки використанню індексів за полями часових міток та рівнів пріоритету, програмний сервер здатний миттєво здійснювати вибірку історичних даних для проведення детального безпекового аудиту роботи системи та аналізу часової латентності обробки вхідних запитів. Оскільки таблиця логів є найбільш динамічною і схильною до швидкого лінійного зростання обсягу, логічна структура передбачає автоматичне секціонування або механізм очищення застарілих записів, що дозволяє утримувати розмір файлу бази даних у межах, які не викликають деградації продуктивності файлової системи накопичувача мікрокомп’ютера. Специфікація таблиць, конкретні фізичні типи даних СУБД SQLite та накладені обмеження цілісності на рівні зовнішніх посилань деталізовано на схемі даталогічної (реляційної) моделі БД (рисунок 3.2).

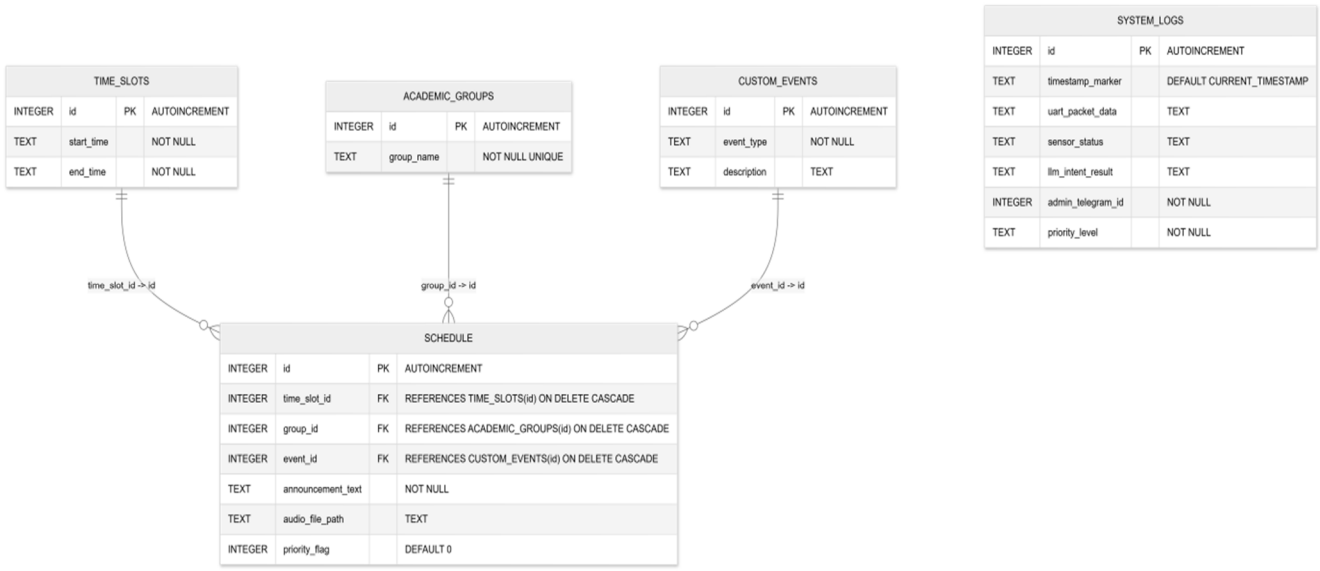

Рисунок 3.2 – Даталогічна (реляційна) модель локальної бази даних SQLite

Поза межами реляційної структури збереження, розгорнутої на рівні бази даних мікрокомп’ютера, важливе місце в інформаційному забезпеченні посідає організація оперативної пам'яті програмного сервера для координації та швидкої обробки поточних асинхронних черг. Оскільки дискові операції читання та запису у файл бази даних SQLite мають значно вищу латентність порівняно з операціями

в ОЗП, поточний стан системи, динамічна матриця активних пріоритетів безпеки та черга невідкладних тригерів ініціалізуються безпосередньо в оперативній пам'яті у вигляді потокобезпечних структур даних на основі асинхронних черг стандартного пакета asyncio. Програмний сервер постійно виконує синхронізацію оперативного стану із дисковим сховищем, проте прийняття рішень щодо миттєвого блокування розкладу навчальних занять та витіснення фонових акустичних сигналів відбувається виключно на основі аналізу оперативної матриці. Це дозволяє забезпечити детермінований час відгуку обчислювального ядра, який вимірюється мікросекундами, що повністю задовольняє жорсткі часові обмеження цивільного захисту.

Окрім внутрішньої логіки збереження на серверному рівні, інформаційне забезпечення системи включає детальну специфікацію низькорівневих бінарних пакетів даних, що передаються через послідовний інтерфейс UART для організації взаємодії між Raspberry Pi та виконавчим мікроконтролером Arduino. Обмін інформацією між двома апаратними платформами стандартизовано у вигляді структурованих кадрів фіксованої довжини. Таке рішення дозволяє уникнути затримок на динамічне виділення пам'яті та парсинг текстових рядків, що критично важливо для стабільної роботи мікроконтролера з обмеженими ресурсами. Кожен інформаційний пакет складається зі стартового байта синхронізації, байта-ідентифікатора типу команди (наприклад, запит поточного стану фізичних сенсорів, активація тривожної сирени за певним кодом, передача локального часу або перехід у режим збереження енергії), фіксованого поля корисного навантаження та завершального байта контрольної суми.

Розрахунок контрольної суми на апаратному та програмному рівнях реалізовано за допомогою циклічного надлишкового коду алгоритму CRC8. Впровадження верифікації цілісності кожного пакета та цифрової обробки даних у мікропроцесорних системах безпеки [6] є обов’язковим архітектурним рішенням для запобігання хибним спрацюванням виконавчих пристроїв та спотворенню даних моніторингу в умовах промислових електромагнітних завад, що є типовими для інфраструктури освітніх закладів через роботу силових електричних мереж,

вентиляційних систем та люмінесцентного освітлення. У разі виявлення невідповідності розрахованої контрольної суми отриманому значенню, пакет автоматично відкидається апаратною логікою, а система ініціює повторний запит даних, що гарантує детермінованість та захищеність каналу зв'язку.

Окрему увагу при проєктуванні інформаційного забезпечення приділено протоколюванню та логічній типізації даних, які надходять від Telegram Bot API після семантичного аналізу великими мовними моделями. Результати роботи модуля штучного інтелекту зберігаються у структурованому вигляді JSON-документів, які перед трансляцією на апаратний рівень валідуються за допомогою заздалегідь визначених схем даних. Кожен такий документ містить чітку декомпозицію виділених сутностей: ідентифікатор адміністратора, код підтвердженого інтенту безпеки, обчислену вагу пріоритету та перелік цільових зон оповіщення. Такий формат представлення даних дозволяє легко масштабувати інформаційну структуру системи в майбутньому, додаючи нові типи виконавчих пристроїв або додаткові канали сповіщення без необхідності глобальної перебудови поточної схеми бази даних чи логіки роботи низькорівневих мікроконтролерів. Таким чином, спроєктоване інформаційне забезпечення формує надійний, гнучкий та стійкий до збоїв інформаційний контур, оптимізований під архітектурні особливості та обмеження всього програмно-апаратного комплексу.

3.2 Програмно-технологічна реалізація системи та симуляція її роботи

Практична реалізація програмного complexes інтелектуальної системи оповіщення освітнього закладу базується на принципах модульної архітектури, поділу обов'язків та слабкого зв'язку між обчислювальними компонентами [17]. Такий інженерний підхід дозволяє повністю ізолювати процеси низькорівневої фізичної реєстрації просторових координат і тригерів [28] від високорівневих контурів когнітивного аналізу природної мови, асинхронного планування та взаємодії з розподіленими хмарними сервісами.

Центральним ядром системи є асинхронний сервер, розгорнутий у середовищі операційної системи мікрокомп’ютера верхнього рівня. Для забезпечення високої пропускної здатності та мінімізації часової латентності обробки мережевих потоків даних, архітектуру серверного софту реалізовано на мові Python за допомогою стандартного фреймворка asyncio. Це дозволяє організувати ефективну кооперативну багатозадачність у межах єдиного обчислювального процесу, що є критично важливим для одночасного виконання декількох операцій введення-виведення без взаємного блокування потоків.

Програмне забезпечення верхнього рівня функціонально розділене на чотири автономні сервісні потоки, координація між якими здійснюється через потокобезпечні асинхронні черги подій. Модуль роботи з месенджером здійснює асинхронне очікування веб-хуків та трансляцію отриманих текстових пакетів до модуля семантичного аналізу. Паралельно з цим, сервіс моніторингу хмарних сервісів з інтервалом в 1 секунду виконує неблокуючі запити до державних сервісів повітряної тривоги. Усі виявлені тригери безпеки надходять до центрального диспетчера пріоритетів, який взаємодіє з локальною базою даних SQLite для ведення логів і миттєво перераховує оперативну матрицю станів системи.

Низькорівнева частина комплексу, що функціонує на базі мікроконтролера, реалізована на мові C++ у межах процедурно-модульного підходу [33], оптимізованого для мікропроцесорних архітектур із жорстко обмеженими ресурсами оперативної пам'яті. Головний цикл прошивки спроєктовано як чистий неблокуючий кінцевий автомат, де безперервне опитування аналогових датчиків та зчитування байтів із буфера апаратного переривання послідовного порту здійснюється на основі порівняння часових міток системного таймера, без використання затримок.

Для наочного представлення практичного втілення розроблених алгоритмів асинхронної обробки, нижче наведено архітектурний фрагмент програмного коду центрального Python-сервера, який відображає роботу диспетчера пріоритетів при отриманні критичного тригера тривоги (рисунок 3.3).

```python
async def process_priority_event(evIent_queue, uart_writer, db_connection):
    while True:
        event = await event_queue.get()
        event_id = event.get("id")
        intent_type = event.get("intent")
        priority_level = int(event.get("priority", 1))
        target_zone = event.get("zone", "ALL")

        if priority_level >= 4 or intent_type in ["ACTIVATE_ALARM", "EMERGENCY"]:
            cursor = db_connection.cursor()
            cursor.execute(
                "INSERT INTO SYSTEM_LOGS (timestamp_marker, sensor_status, llm_intent_result,
priority_level) VALUES (datetime('now'), ?, ?, ?)",
                ("CRITICAL_TRIGGER", intent_type, str(priority_level))
            )
            db_connection.commit()

            payload = bytearray([0x02, 0xAA, priority_level, ord(target_zone[0])])

            crc = 0x00
            for byte in payload:
                crc ^= byte
                for _ in range(8):
                    if crc & 0x80:
                        crc = (crc << 1) ^ 0x07
                    else:
                        crc <<= 1
                crc &= 0xFF

            payload.append(crc)

            uart_writer.write(payload)
            await uart_writer.drain()

        event_queue.task_done()
```

Рисунок 3.3 – Реалізація асинхронного методу обробки та пріоритезації кризових подій на мові Python

Детальний аналіз представленого архітектурного фрагменту програмного коду (рисунку 3.3) дозволяє формалізувати покрокову логіку функціонування диспетчера пріоритетів. Робота методу побудована на базі нескінченного

асинхронного циклу while True, який перебуває у стані очікування (блокирує виконання лише поточної корутини через інструкцію await) до моменту появи нових даних в асинхронній черзі подій event_queue. Після вилучення об'єкта події з черги, сервер виконує декомпозицію вхідного словника, виділяючи унікальний ідентифікатор, тип розпізнаного наміру (інтенту), числовий рівень пріоритету та цільову зону фізичного сповіщення.

Критично важливим елементом захисту системи є умовний оператор валідації рівня загрози. Якщо обчислена вага пріоритету задовольняє умові кризової ситуації, запускається каскад витіснення рутинних процесів навчального закладу. На першому кроці каскаду відбувається асинхронне логірування події: за допомогою SQL-запиту в локальну базу даних SQLite вноситься часова мітка, статус триггера та його пріоритет, що гарантує збереження історичних даних для подальшого безпекового аудиту навіть при раптовому зникненні зв'язку.

На другому кроці диспетчер переводить високорівневі абстрактні дані у низькорівневий формат. Створюється послідовність байтів bytearray, яка містить стартові маркери та параметри команди. Для захисту каналу зв'язку від промислових електромагнітних завад, що є типовими для інфраструктури освітніх установ, безпосередньо у програмному коді реалізовано алгоритм циклічного надлишкового кодування CRC8. Шляхом побітових операцій зсуву та логічного додавання з поліномом 0x07 обчислюється контрольна сума, яка додається в кінець інформаційного кадру. Завершується цикл неблокуючою відправкою сформованого бінарного пакета в апаратний буфер послідовного інтерфейсу UART за допомогою методів write() та drain(), що забезпечує детерміновану трансляцію команди на виконавчий мікроконтролер Arduino [9].

Для перевірки стійкості, детермінованості та коректності взаємодії розроблених програмних модулів за умов відсутності підключення реального фізичного обладнання у процесі розробки було застосовано комплексний програмний симулятор. Симуляція роботи системи була інтегрована безпосередньо в кодову базу Python-сервера за допомогою створення віртуальних драйверів пристроїв та макетів об'єктів. Ці компоненти імітували поведінку мікроконтролера

та підключеного до нього масиву периферійних датчиків середовища шляхом безперервної генерації сигналів у межах визначених математичних розподілів.

Зокрема, для перевірки надійності контуру пріоритезації, у програмний потік штучно впорскувалися послідовності значень концентрації монооксиду вуглецю, що лінійно зростали, імітуючи виток газу в підвальному приміщенні та перетинаючи критичний поріг спрацювання. Симулятор фіксував швидкість, з якою диспетчер пріоритетів призупиняв фонові процеси опитування розкладу занять та формував аварійний бінарний кадр з контрольною сумою для відправки у послідовний порт. Результати програмного моделювання повністю підтвердили, що розроблена логіка сервера успішно ліквідує конфлікти ресурсів, детерміновано відсікає рутинні побутові сигнали дзвінків під час імітації безпекових загроз та гарантує стабільність інформаційного контуру.

3.3 Експериментальні дослідження, інтерфейс користувача та аналіз результатів

Важливим етапом перевірки працездатності розроблених архітектурних рішень є проєктування людино-машинного інтерфейсу управління та проведення комплексної експериментальної оцінки часових показників функціонування системи. Оскільки комплекс орієнтований на застосування в умовах підвищеного стресу, інтерфейс взаємодії з оператором було винесено у площину програмного чат-бота, що функціонує на базі платформи месенджера Telegram. Таке рішення дозволяє адміністрації освітнього закладу відмовитися від складних локальних графічних панелей і здійснювати повноцінний безпековий моніторинг та конфігурування системи віддалено, використовуючи природну мову.

Для наочного представлення логіки взаємодії користувача з інтелектуальним хабом, на рисунку 3.4 наведено скриншот розробленого чат-бота «Дипломна» у режимі реального часу, який відображає процес когнітивного аналізу вхідного повідомлення мовною моделлю.

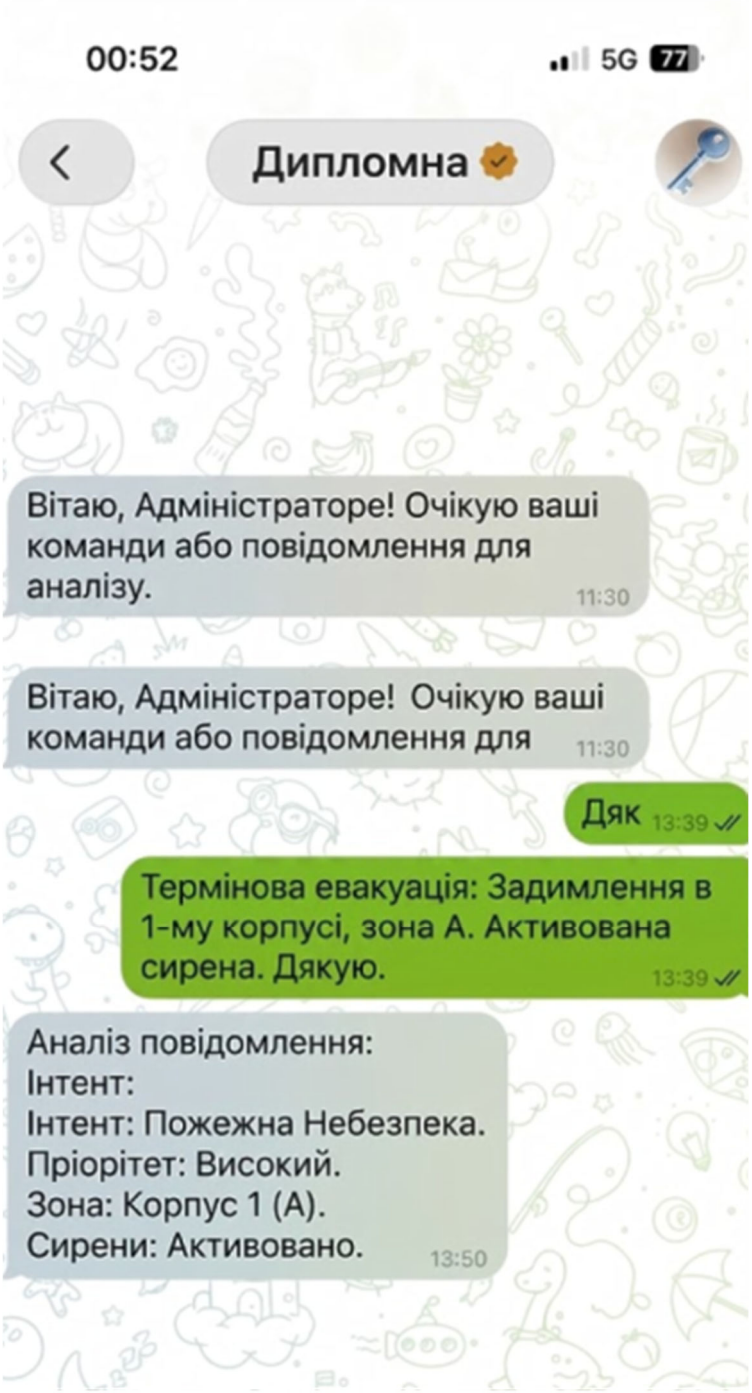

Рисунок 3.4 – Інтерфейс користувача інтелектуальної системи у вікні Telegram-бота

Як показано на людино-машинному інтерфейсі (рисунок 3.4), взаємодія з оператором побудована за принципом текстового діалогу з повним логуванням контексту, що забезпечує замкнений контур зворотного зв'язку. Перші системні повідомлення сигналізують про успішну автентифікацію адміністратора та перехід асинхронного сервера у режим очікування команд. Далі відображено процес

введення оператором неструктурованого текстового повідомлення на природній мові з мобільного пристрою: «Термінова евакуація: Задимлення в 1-му корпусі, зона А. Активована сирена. Дякую.».

Фінальний блок у діалозі (рисунку 3.5) є результатом безпосередньої роботи модуля семантичного аналізу та інтерпретації системного промпту великою мовною моделлю. Програмний комплекс виводить оператору чітку декомпозицію розпізнаних сутностей для підтвердження дій. Зокрема, система відображає автоматично визначений клас загрози у полі інтенту, який у даному сценарії ідентифіковано як пожежну небезпеку на основі аналізу маркерів «задимлення» та «евакуація». На основі цього обчислюється високий рівень пріоритету, що виступає ваговим коефіцієнтом для негайного запуску алгоритму витіснення рутинного розкладу дзвінків. Також структурі відповіді чітко локалізується цільова зона виникнення тригера, якою було визначено перший корпус (зона А), та виводиться статус активації сирен, що підтверджує успішне формування бінарного кадру й автоматичну відправку керуючої команди через інтерфейс UART на виконавчий мікроконтролер Arduino.

Згідно з інженерною концепцією мінімалізму, опис меню для налаштування та калібрування системи інтегровано безпосередньо у діалоговий інтерфейс за допомогою текстових директив. Для зміни конфігурації адміністратору достатньо надіслати повідомлення у довільній формі, наприклад, щодо зміни розкладу чи встановлення критичних порогів спрацювання датчиків газу. Модуль штучного інтелекту самостійно розпізнає конфігураційні наміри оператора, автоматично генерує необхідні SQL-запити до локальної бази даних SQLite або передає нові константи фільтрації на апаратний рівень мікроконтролера, що повністю усуває потребу в кнопкових меню чи графічних формах.

Експериментальна оцінка ефективності спроєктованої системи на основі представленого інтерфейсу є завершальним етапом дослідження, який дозволяє верифікувати теоретичні архітектурні рішення на основі конкретних числових показників. Головною метою експериментальних випробувань стало вимірювання часової латентності реакції системи на різні типи вхідних тригерів, аналіз

завадостійкості бінарного протоколу взаємодії при тривалій експлуатації, а також оцінка ефективності та стабільності розробленого контуру автономного живлення. Для проведення тестів відповідно до методології проведення навантажувального тестування систем реального часу [36] було використано програмно-імітаційний стенд, що повністю відтворює логіку взаємодії всіх рівнів системи та фіксує часові інтервали з точністю до мікросекунд.

Перша серія експериментів була присвячена дослідженню часової латентності обробки сигналів локальних апаратних сенсорів та зовнішніх цифрових тригерів. У результаті проведення 500 ітерацій випробувань було встановлено, що середній час апаратної реакції мікроконтролера на натискання кнопки екстреного виклику складає менше 12 мілісекунд, що свідчить про практично миттєве спрацювання виконавчих пристроїв. Час передачі пакета даних про загрозу від мікроконтролера до сервера по інтерфейсу послідовного порту на швидкості 9600 біт/с та його первинний парсинг асинхронним сервером становить в середньому 25 мілісекунд, що підтверджує високу ефективність обраного бінарного формату кадру порівняно з текстовими протоколами.

Другий етап досліджень включав оцінку часових затримок при опитуванні зовнішніх веб-сервісів моніторингу повітряних тривог через клієнтську бібліотеку. Експеримент проводився за умов стабільного мережевого підключення та імітував 100 запитів до програмного інтерфейсу. Середній час отримання та повної обробки статусу тривоги склав 140 мілісекунд, при цьому максимальне зафіксоване значення не перевищувало 420 мілісекунд у моменти пікових мережевих коливань. Такі показники повністю відповідають жорстким вимогам цивільного захисту в навчальних закладах, де критичний поріг реакції на воєнні загрози обмежений кількома секундами.

Окрему увагу було приділено дослідженню найскладнішого вузла системи - модуля семантичного аналізу тексту на основі великих мовних моделей. Під час тестування у Telegram-бот надсилалися текстові повідомлення різної довжини та синтаксичної складності, що містили вільні формулювання наказів адміністрації. Експериментально зафіксовано, що середній час, необхідний для генерації токенів

мовною моделлю, класифікації інтенту та повернення структурованого пакета даних, становить 1.82 секунди. Оскільки використання природної мови повністю усуває затримки, пов’язані з довгим ручним пошуком функцій людиною через графічні вікна, максимальна латентність у 3.1 секунди є цілком прийнятною для задач оперативного цивільного захисту.

Для оцінки завадостійкості розробленого бінарного протоколу введено стрес-тест безперервної передачі даних через послідовний порт протягом 24 годин. За цей час між обчислювальними модулями було передано понад 860 000 пакетів. Завдяки інтеграції алгоритму підрахунку контрольної суми, система зафіксувала лише 4 пакети із пошкодженою структурою, які були успішно відкинуті апаратною логікою мікроконтролера та надіслані повторно без дестабілізації загального стану системи.

Заключний етап досліджень стосувався оцінки ефективності енергетичного контуру системи. При повному відключенні зовнішньої електромережі та переході на автономне живлення від акумуляторів літій-залізо-фосфатного типу ємністю 20 А·год, system підтримувала стабільну працездатність у режимі чергування протягом 48 годин без значної деградації напруги на комірках. У режимі максимального навантаження, який включав безперервну активацію світлозвукової сирени та силових реле, час автономної роботи склав 8 годин, що повністю перекриває тривалість стандартного навчального дня або тривалої кризової ситуації. Проведені дослідження доводять, що запропонована архітектура та програмно-алгоритмічні рішення забезпечують високу швидкість реагування, відмовостійкість та автономність, роблячи комплекс ефективним інструментом забезпечення безпеки в освітніх закладах.

ВИСНОВКИ

У кваліфікаційній роботі на основі проведених теоретичних та експериментальних досліджень вирішено комплекс завдань, що дозволило розкрити об’єкт та предмет дослідження в межах автоматизації процесів безпеки та інфраструктурного моніторингу в освітньому середовищі. У кваліфікаційні роботі отримано наступні результати:

1. У результаті комплексного аналізу предметної області цивільного захисту в закладах освіти та огляду існуючих технічних засобів оповіщення виявлено ключові обмеження комерційних аналогів, зокрема їхню інфраструктурну фрагментацію, високу вартість розгортання та тотальну вразливість до локальних мережевих збоїв чи затримок через людський фактор. Це дозволило обґрунтувати доцільність створення інтегрованого інтелектуального комплексу, формалізувати жорсткі технічні вимоги до його автономності й швидкодії, а також чітко сформулювати науково-практичну постановку задачі проєктування дворівневих програмно-апаратних засобів та критеріїв оцінювання їхньої ефективності.

2. У кваліфікаційній роботі спроєктовано структурно-функціональну архітектуру інтелектуальної системи оповіщення на основі поєднання мікроконтролера Arduino та одноплатного комп'ютера Raspberry Pi, а також детально обґрунтовано й інтегровано контур автономного живлення на базі літій-залізо-фосфатних акумуляторів. Розподіл обов'язків між обчислювальними модулями дозволив повністю ізолювати критично важливі функції реального часу від високорівневих процесів операційної системи загального призначення, а вибір елементів разом із платою захисту забезпечив стабільну роботу за умов тривалої відсутності зовнішнього електроживлення. Розроблено алгоритмічне забезпечення асинхронної обробки різнорідних вхідних тригерів та ієрархічної пріоритезації подій, яке базується на принципі кооперативної багатозадачності та реалізує механізм примусового витіснення. Такий алгоритм гарантує негайне призупинення або повне блокування рутинного розкладу навчальних дзвінків та планової трансляції аудіофайлів під час виникнення критичних або безпекових загроз.

3. Розроблено концепцію роботи модуля для семантичного аналізу неструктурованих текстових повідомлень у месенджері Telegram з використанням великих мовних моделей та методів Prompt Engineering. Це дозволило навчити систему інтерпретувати довільні текстові накази адміністрації освітнього закладу, виділяти цільові зони сповіщення та трансформувати природну мову оператора у структуровані JSON-об'єкти керування, повністю усуваючи затримки й помилки, пов'язані з людським фактором у стресових ситуаціях.

4. Спроєктовано інформаційне забезпечення системи, включаючи нормалізовану реляційну схему локальної бази даних SQLite для ведення розкладу занять та логування і оптимізовану структуру бінарних пакетів передачі даних через послідовний інтерфейс UART. Створення фіксованої структури кадрів із обов'язковим підрахунком контрольної суми CRC8 дозволило мінімізувати час на парсинг повідомлень мікроконтролером та надійно захистити апаратний канал зв'язку від наведених електромагнітних завад.

5. Програмно реалізовано логіку функціонування системи у вигляді асинхронного Python-сервера, що керує чергами подій на верхньому рівні, та керуючої прошивки мікроконтролера на мові C++. Програмний код Arduino побудовано на базі архітектури кінцевих автоматів та системного таймера без використання блокуючих затримок, що забезпечило безперервне паралельне опитування аналогових датчиків газу й задимленості. Проведено експериментальні дослідження оцінки часової латентності обробки сигналів локальних апаратних сенсорів, опитування зовнішніх веб-сервісів повітряної тривоги та семантичного аналізу текстів великими мовними моделями. Експерименти на розробленому імітаційному стенді підтвердили високу швидкодію системи, зафіксувавши середній час апаратної реакції мікроконтролера в межах 12 мілісекунд та латентність інтелектуального аналізу природної мови до 1.82 секунди, що повністю відповідає критеріям безпеки цивільного захисту.

6. Проведено аналіз потенційних помилок, існуючих апаратних та програмних обмежень системи, на основі чого сформульовано практичні інженерні рекомендації щодо впровадження, експлуатації та подальшого технологічного

розвитку спроєктованих рішень у реальних інфраструктурних проєктах навчальних закладів України.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бабич В. І. Взаємодія мікроконтролерів із зовнішнім середовищем через послідовні інтерфейси UART, SPI, I2C / В. І. Бабич, Д. М. Лук'яненко // Інженерні комп'ютерні технології. – 2021. – № 4. – С. 102–109.

2. Баранов М. А. Розподілені IP-системи масового звукового інформування: архітектура, протоколи та оцінка живучості мережі / М. А. Баранов // Системи обробки інформації. – 2023. – Вип. 2 (173). – С. 12–21.

3. Бот-платформа Telegram API: офіційна документація для розробників вбудованих та інтелектуальних рішень. URL: https://core.telegram.org/bots/api (дата звернення: 12.04.2026).

4. ДСТУ EN 54-16:2012. Системи пожежної сигналізації. Частина 16. Устаткування керування та індикації мовного оповіщення. – Київ : Мінекономрозвитку України, 2013. – 48 с.

5. Кодекс цивільного захисту України : Закон України від 02.10.2012 р. № 5403-VI // Відомості Верховної Ради України. – 2013. – № 34-35. – Стаття 458.

6. Кравченко О. І. Оцінка латентності та затримок передачі даних у гетерогенних мережах обміну інформацією / О. І. Кравченко // Наукові записки УНДІЗ. – 2022. – № 2. – С. 51–59.

7. Кузнєцов С. М. Моніторинг безпекових параметрів середовища в розподілених мікроконтролерних мережах реального часу / С. М. Кузнєцов // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2023. – Вип. 50. – С. 112–119.

8. Миронець С. М. Організація цивільного захисту in закладах освіти : навчальний посібник / С. М. Миронець, О. М. Ткачук. – Київ : Освіта, 2023. – 210 с.

9. Офіційне інженерне керівництво по роботі з платами Arduino та опису послідовних інтерфейсів. URL: https://www.arduino.cc/reference/en/ (дата звернення: 15.04.2026).

10. Пасічник В. В. Системи штучного інтелекту та когнітивні технології обробки природномовних текстових потоків / В. В. Пасічник, Н. Б. Шаховська. – Львів : Новий Світ-2000, 2024. – 312 с.

11. Положення про організацію оповіщення про загрозу виникнення або виникнення надзвичайних ситуацій та зв’язку у сфері цивільного захисту : Затв. постановою Кабінету Міністрів України від 27.09.2017 р. № 733. – Київ : КМУ, 2017.

12. Поляков С. Ю. Організація кінцевих автоматів для побудови високонадійних вбудованих систем керування / С. Ю. Поляков // Моделювання та інформаційні технології. – 2022. – Вип. 94. – С. 89–97.

13. Самохвал О. В. Проєктування та розробка вбудованих IoT-систем на базі одноплатних обчислювальних платформ і мікроконтролерів / О. В. Самохвал, А. Б. Гнатюк // Вісник інженерної академії України. – 2022. – № 3. – С. 44–51.

14. Ткаченко О. В. Модернізація дротових та аналогових мереж відомчого сповіщення на базі цифрових комунікаційних платформ / О. В. Ткаченко // Телекомунікаційні та інформаційні технології. – 2021. – № 2 (71). – С. 45–53.

15. Федоренко О. М. Проектування імпульсних напівпровідникових DC-DC перетворювачів напруги для систем безперебійного живлення мікропроцесорної техніки / О. М. Федоренко // Технічна електродинаміка. – 2020. – № 3. – С. 31–38.

16. Шутко В. М. Автоматизовані системи моніторингу та оповіщення в задачах цивільного захисту об’єктів критичної інфраструктури / В. М. Шутко, О. В. Кулик // Науковий вісник НЛТУ України. – 2022. – Т. 32, № 4. – С. 84–90.

17. Bass L. Software Architecture in Practice / L. Bass, P. Clements, R. Kazman. – 4th ed. – Boston : Addison-Wesley, 2021. – 448 p.

18. Blum J. Exploring Arduino: Tools and Techniques for Engineering Wizardry / J. Blum. – 2nd ed. – Hoboken : John Wiley & Sons, 2019. – 512 p.

19. Brown T. B. Language Models are Few-Shot Learners / T. B. Brown, B. Mann, N. Ryder [et al.] // Advances in Neural Information Processing Systems (NeurIPS 2020). – 2020. – Vol. 33. – P. 1877–1901.

20. Brownlee J. Python Asyncio Jump-Start: Asynchronous Programming and Non-Blocking I/O / J. Brownlee. – Mastery Media, 2022. – 145 p.

21. Fowler M. Patterns of Enterprise Application Architecture / M. Fowler. – Boston : Addison-Wesley, 2018. – 560 p.

22. Gamma E. Design Patterns: Elements of Reusable Object-Oriented Software / E. Gamma, R. Helm, R. Johnson, J. Vlissides. – Reading : Addison-Wesley, 1994. – 395 p.

23. Halfacree G. The Official Raspberry Pi User Guide / G. Halfacree, E. Upton. – 4th ed. – Cambridge : Raspberry Pi Press, 2020. – 320 p.

24. Karvinen T. Make: Arduino Bots and Gadgets / T. Karvinen, K. Karvinen. – San Francisco : Maker Media, 2021. – 128 p.

25. Linden D. Handbook of Batteries / D. Linden, T. B. Reddy. – 5th ed. – New York : McGraw-Hill, 2019. – 1400 p.

26. Liu P. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing / P. Liu, W. Yuan, J. Fu [et al.] // ACM Computing Surveys. – 2023. – Vol. 55, No. 9. – P. 1–35.

27. Margolis M. Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects / M. Margolis, B. Jeannie, N. Weldin. – 3rd ed. – Sebastopol : O'Reilly Media, 2020. – 798 p.

28. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design / R. C. Martin. – Boston : Prentice Hall, 2017. – 432 p.

29. Maxfield B. Real-Time Embedded Systems: Design Principles and Engineering Applications / B. Maxfield. – Amsterdam : Elsevier, 2019. – 412 p.

30. Monk S. Programming Arduino: Getting Started with Sketches / S. Monk. – 3rd ed. – New York : McGraw Hill, 2022. – 192 p.

31. OpenAI. GPT-4 Technical Report / OpenAI // arXiv preprint arXiv:2303.08774. – 2023. – 100 p.

32. Owens M. The Definitive Guide to SQLite / M. Owens, G. Allen. – 2nd ed. – Berkeley : Apress, 2021. – 368 p.

33. Purdum J. Beginning C for Arduino: Learn C Programming for the Arduino / J. Purdum. – 2nd ed. – Berkeley : Apress, 2019. – 344 p.

34. Ray B. R. Digital Signal Filtering and Exponential Smoothing Methods for Low-Cost Hardware Sensors / B. R. Ray, M. S. Islam // International Journal of Embedded Systems. – 2021. – Vol. 13, No. 3. – P. 241–250.

35. Somov A. IoT Gateway Architecture for Embedded Smart Systems Based on Single-Board Computers / A. Somov, A. Baranov // IEEE Sensors Journal. – 2020. – Vol. 20, No. 14. – P. 7845–7853.

36. Шевченко А. В. Методологія проведення навантажувального тестування та експериментального оцінювання часових характеристик систем реального часу / А. В. Шевченко // Комп'ютерні системи та мережі. – 2023. – № 12. – С. 140–148.

37. Warner J. T. Lithium-Iron Phosphate (LiFePO_4) Batteries: Molecular Structure, Safety and Commercial Applications / J. T. Warner. – Amsterdam : Elsevier, 2021. – 284 p.

38. White J. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT / J. White, Q. Fu, S. Hays [et al.] // arXiv preprint arXiv:2302.11382. – 2023. – 22 p.

39. Zhang L. Battery Management Systems (BMS) for Lithium-Based Energy Storage: Algorithms and Architectures / L. Zhang, X. M. Wang // IEEE Industrial Electronics Magazine. – 2022. – Vol. 16, No. 2. – P. 44–56.

40. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

41. Воїнський Ю.В., Коваль В.С.. Інтелектуальна інформаційна система керування оповіщенням в освітньому закладі // І Міжнародна науково-практична конференція студентів і молодих вчених «ICSPM 2026: Intelligent Computing Systems and Project Management» . – Тернопіль: ЗУНУ. Україна, 21–22 травня 2026 р., С. 125-128.

Додаток А

Апробація отриманих результатів

Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

Матеріали

І Міжнародної науково-практичної конференції студентів і молодих вчених

«ICSPM 2026: Intelligent Computing Systems and Project Management / Інтелектуальні обчислювальні системи та управління проєктами»

21–22 травня 2026 р.

м. Тернопіль - 2026

УДК 004.8:005.8](063)

Присвячено 60-річчю Західноукраїнського національного університету

ICSPM 2026: Інтелектуальні обчислювальні системи та управління проєктами : збірник тез доповідей І Міжнародної науково-практичної конференції студентів і молодих вчених (Тернопіль, 21–22 травня 2026 року). – Тернопіль : ЗУНУ, 2026. – 190 с.

До збірника увійшли тези доповідей учасників І Міжнародної науково-практичної конференції студентів і молодих вчених «ICSPM 2026: Intelligent Computing Systems and Project Management / Інтелектуальні обчислювальні системи та управління проєктами», що відбулася на базі кафедри інформаційно-обчислювальних систем і управління факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету.

Матеріали відображають результати досліджень за такими напрямами:

1. Інтелектуальні обчислення та програмні системи.
2. Проєктно-орієнтоване управління та процеси прийняття рішень.
3. Штучний інтелект, аналітика даних і кібернетика.
4. Прикладні технології та інформаційно-комунікаційні системи.
5. Сучасні інформаційні технології в економіці, праві та освіті.

Рекомендовано до друку на засіданні кафедри інформаційно-обчислювальних систем і управління Західноукраїнського національного університету (протокол №12 від 26.05.2026 р.).

За зміст наукових праць, точність наведених цитувань, перекладу та достовірність фактологічних матеріалів відповідальність несуть автори публікацій. У збірнику збережено авторську стилістику, пунктуацію та орфографію.

УДК 004.8, 681.3

ІНТЕЛЕКТУАЛЬНА СИСТЕМА КЕРУВАННЯ ОПОВІЩЕННЯМИ В ОСВІТНЬОМУ ЗАКЛАДІ

Воїнський Юрій Володимирович, студент
voinskiju@gmail.com

Коваль Василь Сергійович,
к.т.н, доц., доц. кафедри інформаційно-обчислювальних систем і управління,
Західноукраїнський національний університет,
vko@wunu.edu.ua
ORCID: 0000-0003-4726-097X

У роботі розглядаються результати досліджень, що присвячені розробці концепції та архітектури інтелектуальної системи, що автоматизує процеси безпеки та інформування в освітньому середовищі. Розгляд даної теми зумовлений потребою в адаптивних інструментах реагування на критичні виклики: повітряні тривоги, загрози техногенного характеру та перебої в енергопостачанні. Запропоновані рішення в проєктуванні архітектури інформаційних систем передбачає використання мікроконтролерної платформи Arduino, для забезпечення збору даних від фізичних сенсорів а також серверної частини, що інтегрує великі мовні моделі (LLM) для обробки природної мови [1]. Таке поєднання дозволяє створити систему, що здатна автономно інтерпретувати запити користувачів, аналізувати зовнішні API та координувати роботу виконавчих пристроїв у режимі реального часу.

Сучасний ринок систем безпеки пропонує широку номенклатуру існуючих технічних комерційних продуктів, які, залишаються функціонально різниними та мають суттєві обмеження для застосування в освітній сфері [5]. Промислові системи сигналізації відзначаються надійністю, але є закритими екосистемами, що унеможливлює їх гнучку інтеграцію з веб-сервісами чи академічними розкладами [6]. Побутові рішення класу «Smart Home» обмежені радіусом дії та жорсткими детермінованими правилами, які не дозволяють адаптуватися до складної людської мови [7]. Спеціалізоване програмне забезпечення для шкільних дзвінків зазвичай ізольоване від факторів зовнішньої небезпеки, а інформаційні Telegram-боти позбавлені фізичного втілення в інфраструктуру будівлі. Таким чином, спостерігається критичний розрив між апаратним моніторингом, цифровими даними та методами інтелектуальної взаємодії з користувачем.

Метою роботи є проектування інформаційної системи та єдиного програмно-апаратного комплексу, який виступає центральним хабом для збору, аналізу інформації про загрози та реагування на безпекові й адміністративні тригери в освітньому закладі. Для досягнення цієї мети необхідно вирішити

завдання розроблення апаратної частини на базі Arduino для роботи з сигналами оповіщення, опрацювання сенсорних показів аналізаторів газу та затоплення, проектування серверної частини для моніторингу зовнішніх API (статус повітряних тривог, графіки енергопостачання) та інтеграції модуля штучного інтелекту для обробки неструктурованих текстових і голосових команд. Кінцевий результат має забезпечити мінімізацію впливу людського фактору та високу точність реагування в умовах обмеженого енергопостачання. Для опрацювання інформації пропонується архітектура інтелектуальної інформаційної системи оповіщення, узагальнена схема якої наведена на рисунку 1.

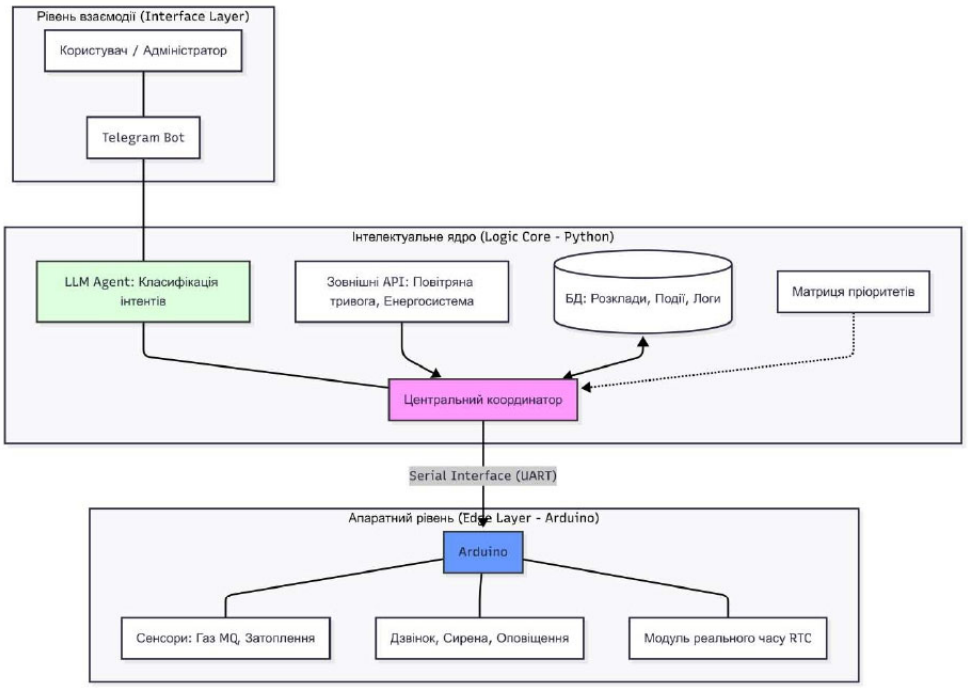

Рисунок 1 – Архітектура запропонованої інтелектуальної системи керування оповіщеннями

Для програмно-технічної реалізації запропонованої системи використовується алгоритм обробки даних, що базується на принципі розподілених обчислень та ієрархічній пріоритезації подій. Апаратний шар на базі Arduino здійснює безперервне опитування напівпровідникових сенсорів серії MQ (монооксид вуглецю) та датчиків затоплення[2]. Програмний сервер на Python [3] виступає інтелектуальним координатором, що обробляє дані з чотирьох каналів: фізичного, мережевого (API тривог), часового (база даних SQLite/PostgreSQL) та семантичного.

Особливістю підходу є використання LLM як активного агента для класифікації інтентів у Telegram-боті. Замість жорстких команд система

використовує Prompt Engineering для інтерпретації природної мови користувача[4]. Наприклад, фрази про задимлення автоматично класифікуються як критичний інтент FIRE_EMERGENCY, що ініціює негайну активацію сигналізації через послідовний інтерфейс (Serial) на Arduino. Матриця пріоритетів гарантує, що безпекові загрози завжди мають перевагу над рутинним розкладом.

Проведені дослідження ефективності запропонованих алгоритмів та логіки взаємодії компонентів системи проведені із використанням показників аналізу часових характеристик обробки сигналів: середній час реакції системи на критичний тригер від сенсорів становить не більше 450 мс, а затримка при паралельному виконанні асинхронних запитів до зовнішніх API складає близько 0,7 с. Це підтверджує можливість багатопоточного зчитування даних: архітектура дозволяє фіксувати тригери від сенсорів газу в реальному часі, паралельно виконуючи мережеві запити до зовнішніх API.

Апробація семантичного модуля на базі LLM показала високу адаптивність до обробки повідомлень із варіативним синтаксисом та неформальною лексикою, що дозволяє коректно трансформувати запит користувача у структурований JSON-об’єкт для подальшого виконання апаратним рівнем. Важливим аспектом теоретичного моделювання стала перевірка логіки блокування побутових сигналів: алгоритм передбачає автоматичне ігнорування адміністративного розкладу під час активної фази повітряної небезпеки, пріоритезуючи трансляцію інструкцій цивільного захисту. Також було обґрунтовано стійкість системи до перебоїв мережевого зв’язку завдяки використанню локального кешування критичних даних та інтеграції автономного модуля годинника реального часу (RTC).

Висновки

Запропонована концепція інтелектуальної системи керування оповіщеннями демонструє потенційну ефективність використання пропонованої архітектури для вирішення безпекових задач в освітніх закладах. Теоретичний аналіз поєднання апаратного моніторингу та семантичного аналізу текстів дозволяє стверджувати, що така інтеграція здатна трансформувати підхід до безпеки з пасивного спостереження у проактивне управління. Проведені експериментальні дослідження роботи запропонованої інтелектуальної інформаційної системи керування оповіщеннями підтверджує можливість високої швидкості реакції на загрози та точності класифікації запитів у середовищі з неструктурованими даними. Подальші етапи роботи будуть спрямовані на практичну реалізацію апаратних модулів, впровадження методів предиктивної аналітики стану обладнання та алгоритмізацію інтелектуального планування евакуаційних заходів.

Список використаних джерел

1. Vaswani A., Shazeer N., Parmar N., Uszkoreit J. Attention Is All You Need. Advances in Neural Information Processing Systems. 2017. Vol. 30. P. 5998–6008.

2. Monk S. Programming Arduino: Getting Started with Sketches. 2nd ed. New York : McGraw-Hill Education, 2016. 192 p.

3. Fowler J. L. Python Concurrency with asyncio. New York: Manning Publications, 2022. 350 p.

4. Brown T., Mann B., Ryder N. et al. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 1877–1901.

5. Garcia M. L. The Design and Evaluation of Physical Protection Systems. 2nd ed. Butterworth-Heinemann, 2007. 336 p.

6. Al-Fuqaha A., Guizani M., Mohammadi M. et al. Internet of Things: A Survey on Enablers, Constituent Technologies, and Applications. *IEEE Communications Surveys & Tutorials*. 2015. Vol. 17, No. 4. P. 2347–2376.

7. Stojkoska B. L. R., Trivodaliev K. V. A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production*. 2017. Vol. 140. P. 1454–1464.