МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп’ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувачка кафедри,
д-р техн. наук, проф.

_____Ірина ЖУРАВСЬКА

« __ » _____ 202__ р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

Система сповіщення щодо потрапляння в надзвичайну ситуацію на базі GSM модуля

Спеціальність 123 Комп’ютерна інженерія

Освітня програма «Комп’ютерна інженерія»

Здобувач _____Артем ЧАУС

підпис

«__» _____ 2025р.

Керівник PhD, старший викладач _____Євген ДАРНАПУК

підпис

«__» _____ 2025р.

Миколаїв – 2025

Факультет	Комп’ютерних наук
Кафедра	Комп’ютерної інженерії
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	123 Комп’ютерна інженерія
Освітня програма	Комп’ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри комп’ютерної інженерії
_____ Ірина ЖУРАВСЬКА
«_____» _____ 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Чауса Артема Дмитровича
(прізвище, ім’я, по батькові здобувача)

1. Тема кваліфікаційної роботи
Система сповіщення щодо потрапляння в надзвичайну ситуацію на базі GSM модуля

Затверджена наказом по ЧНУ ім. Петра Могили від 30.10.2024 № 297.

2. Строк представлення кваліфікаційної роботи «____» _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні
Очікуваним результатом роботи є апаратне та програмне забезпечення для системи сповіщення населення про надзвичайні ситуації, з можливістю автоматичного реагування. Вхідними даними роботи є специфікація вимог, що описує характеристики зазначеного апаратного та програмного забезпечення.

4. Перелік питань, що підлягають розробці:
1) проаналізувати методи оповіщення про небезпечні ситуації;
2) проаналізувати існуючи рішення в галузі оповіщення населення;
3) обгрунтовано здійснити вибір апаратних компонентів системи;
4) розробити програмне забезпечення мовою Arduino C++;
5) реалізувати виведення інформаційних повідомлень щодо стану системи на Serial монітор;
6) перевірити систему, протестувавши основні ситуації.

5. Перелік графічних матеріалів
Презентація

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Керівник роботи _____ __Євген ДАРНАПУК_____
Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Здобувач _____ __Артем ЧАУС_____
Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Дата видачі завдання «_____» _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної бакалаврської роботи

Тема: ___Система сповіщення щодо потрапляння в надзвичайну ситуацію на базі GSM модуля _____

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КБР	30.10.2024	30.10.2024	Виконано
2	Огляд літератури за темою роботи	01.11.2024	14.01.2025	Виконано
3	Складання календарного плану КБР	16.01.2025	01.02.2025	Виконано
4	Аналіз предметної області	03.02.2025	20.02.2025	Виконано
5	Розробка проєктних рішень	21.02.2025	21.03.2025	Виконано
6	Моделювання та конструювання АПК	29.03.2025	20.04.2025	Виконано
7	Перевірка працездатності, тестування та апробація розробленого АПК, аналіз результатів тестування	21.04.2025	15.05.2025	Виконано
8	Відгук керівника КБР	28.04.2025	13.06.2025	Виконано
9	Оформлення КБР та презентації	28.04.2025	03.06.2025	Виконано
10	Попередній захист	21.05.2025	04.06.2025	Виконано
11	Рецензування	08.06.2025	09.06.2025	Виконано
12	Завершення оформлення КБР та презентації	10.06.2025	17.06.2025	Виконано
13	Захист кваліфікаційної бакалаврської роботи	24.06.2025	24.06.2025	Виконано

Керівник роботи _____ __Євген ДАРНАПУК_____

Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Здобувач _____ __Артем ЧАУС_____

Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

«____» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи
«Система сповіщення щодо потрапляння
в надзвичайну ситуацію на базі GSM модуля»
Студент 405 гр.: Чаус Артем Дмитрович
Керівник: PhD, старший викладач Дарнапук Євген Сергійович

У сучасних умовах підвищеної техногенної небезпеки, системи автоматичного оповіщення населення набувають особливої актуальності. Об’єктом дослідження є процес попередження людей про надзвичайні ситуації, а предметом – апаратно-програмний комплекс з використанням Global System for Mobile Communications (GSM)-модуля для інформування у разі небезпеки. Метою даної кваліфікаційної роботи було розробити функціональну систему на базі Arduino, яка здатна самостійно виявляти загрозливі фактори, як пожежа, витік газу, підвищення температури чи рух, та миттєво реагувати відповідно до заданого сценарію.

У першому розділі роботи було проведено аналітичний огляд тематики. Було проаналізовано методи оповіщення про надзвичайні ситуації, визначено ключові технології, що використовуються у сучасних системах сповіщення. Також розглянуто існуючі рішення у різних країнах світу, їх ефективність, переваги та недоліки, що дозволило визначити напрямок власної розробки. У другому розділі розглянуто математичні методи та алгоритми, необхідні для роботи системи сповіщення. Було сформульовано основне завдання, проведено оцінку ефективності обробки даних, побудовано алгоритми реагування на небезпеку.У третьому розділі описано програмну реалізацію системи. Було створено програму мовою Arduino C++, яка забезпечує автоматичну реакцію на виявлені загрози: надсилає сигнал тривоги, відкриває двері, вмикає світлову індикацію та виводить повідомлення на монітор. Окремо реалізовано функціональність, яка в майбутньому дозволить підключити модуль GSM для надсилання Short Message Service (SMS)-повідомлень на телефон користувача.

У висновках підсумовано результати розробки апаратно-програмної системи для виявлення надзвичайних ситуацій на базі Arduino з підтримкою GSM. Досягнута мета – створити робочу модель системи, що самостійно реагує на загрози: виявляє дим, температуру, газ або рух, вмикає тривогу, відкриває двері та інформує користувача.

Робота складається зі 76 сторінок (без додатків), містить 15 таблиць, 17 рисунків, 25 джерел та 1 додаток з програмним кодом.

Ключові слова: *GSM, Arduino, тривога, сповіщення, сенсори, пожежа, аварія, Tinkercad, автоматизація, система безпеки.*

ABSTRACT

of the Bachelor's Thesis

«Notification system for in case of emergency based on GSM module»

Student: Artem Chaus

Supervisor: PhD, senior lecturer Yevhen Darnapuk

In today's environment of increased man-made hazards, automatic public warning systems are of particular relevance. The object of research is the process of warning people about emergencies, and the subject is a hardware and software complex using the Global System for Mobile Communications (GSM) module for informing in case of danger. The purpose of this qualification work was to develop a functional system based on Arduino, which is able to independently detect threatening factors such as fire, gas leakage, temperature rise or movement, and instantly respond according to a given scenario.

The first section of the paper provides an analytical overview of the topic. The methods of emergency notification were analyzed, and the key technologies used in modern notification systems were identified. Existing solutions in different countries of the world, their effectiveness, advantages and disadvantages were also considered, which allowed us to determine the direction of our own development. The second section discusses the mathematical methods and algorithms necessary for the operation of the notification system. The main task was formulated, the efficiency of data processing was evaluated, and algorithms for responding to danger were built.The third section describes the software implementation of the system. We created an Arduino C++ program that provides an automatic response to detected threats: sends an alarm, opens the door, turns on the light indication, and displays messages on the monitor. A separate functionality has been implemented that will allow connecting a GSM module to send Short Message Service (SMS) messages to the user's phone in the future.

The conclusions summarize the results of the development of a hardware and software system for detecting emergencies based on Arduino with GSM support. The goal achieved was to create a working model of a system that responds to threats independently: detects smoke, temperature, gas, or motion, raises an alarm, opens the door, and informs the user.

The work consists of 76 pages (excluding appendices), contains 15 tables, 17 figures, 25 references and 1 appendix with program code.

Keywords: *GSM, Arduino, alarm, notification, sensors, fire, accident, Tinkercad, automation, security system.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

АПК – апаратно-програмний комплекс
ДСНС – Державна служба України з надзвичайних ситуацій
ДТ – датчик температури
ЗЗ – захист від замикань
МК – мікроконтролер
ПЗ – програмне забезпечення
ШІ – штучний інтелект

API – Application Programming Interface
GPS – Global Positioning System
GSM – Global System for Mobile Communications
HTTP – HyperText Transfer Protocol
IoT – Internet of Things
LTE – Long-Term Evolution
SIM – Subscriber Identity Module
SMS – Short Message Service
WI–FI – Wireless Fidelity

ВСТУП

У світі часто стаються небезпечні ситуації, такі як землетруси чи аварії на виробництвах, тому потрібно швидко повідомляти людей, щоб якомога менше було шкоди, а головне – зберегти життя населення та їх майно. Звичайні способи попередження, наприклад, сирени, нерідко підводять не стабільною роботою: обривається зв’язок, зникає живлення, саме тоді коли треба терміново дати сигнал про небезпеку.

У роботі йде аналіз, як Global System for Mobile Communications (GSM) – модулі роблять системи сповіщення надійнішими. Вони дозволяють швидко передавати тривожні сигнали, коли стається катастрофа – наприклад, повінь у селі чи витік газу на заводі. У статті Азарова С. І. та Єременка С. А. знайдено, що ці модулі допомагають стежити за ситуацією без затримок. Датчик фіксує проблему, і за секунди повідомлення вже на телефонах людей. Це дає шанс вчасно втекти чи захистити майно. Пригадуючи Чорнобильську катастрофу – тоді через повільне попередження катастрофа стала ще більшою. Якби були такі системи, можливо, вдалося б врятувати більше життів і природу [1].

У світі вже є приклади, коли такі системи рятують людей. У США через телефони дають повідомлення, якщо приближається ураган, а в Японії датчики попереджають про землетрус, навіть тоді, коли він ще не почався. В Україні GSM для таких випадків тільки починають використовувати, але якщо це розвинути, можна добре допомогти і людям, і природі. Тому варто розібратися в цій темі, щоб зробити систему, що швидко попереджає всіх про небезпеку. Ця робота показує, як краще попереджати про небезпеку.

План створити пристрій, який через GSM-мережу надсилатиме людям попередження, і написати для нього програму, щоб увесь процес йшов добре й не запізнювався.

Щоб це запрацювало, потрібно виконати кілька пунктів:

1) проглянути, як діють системи сповіщення, що вже є, вияснити, що в них не виходить, і подумати, як можна підключити до них GSM-зв’язок;

2) скласти список вимог до пристрою – що має вміти GSM-модуль, які датчики потрібні й як усе це має працювати разом;

3) написати програму, яка сама розбиратиметься з сигналами й надсилатиме повідомлення, без потреби втручання людей;

4) перевірити роботу системи в умовах, наближених до реальних, щоб оцінити її ефективність;

5) оцінити, як добре працює пристрій, та виявити його слабкі і сильні сторони, щоб була можливість модифікувати його у майбутньому.

Об’єкт дослідження: процес попередження людей про техногенні аварії чи інші небезпеки.

Предмет дослідження: апаратно-програмний комплекс (АПК) із GSM-модулем, з автоматичним інформуванням про небезпечні ситуації.

Мета роботи: метою роботи було створити і перевірити в симуляції систему оповіщення на базі Arduino, яка автоматично помічає небезпечні ситуації, такі як пожежа, витік газу чи рух, і одразу реагує: вмикає світлові сигнали, відкриває двері та показує повідомлення на моніторі.

Завдання:

1) проаналізувати методи оповіщення про небезпечні ситуації;
2) проаналізувати існуючі рішення в галузі оповіщення населення;
3) обгрунтовано здійснити вибір апаратних компонентів системи;
4) розробити програмне забезпечення мовою Arduino C++;
5) реалізувати виведення інформаційних повідомлень щодо стану системи на Serial монітор;
6) перевірити систему, протестувавши основні ситуації.

Розробка є доцільною, оскільки існуючі системи сповіщення часто не встигають або погано працюють із GSM-модулями. Багато з них коштують дорого, щоб їх ставили скрізь, або швидко розряджаються чи не сумісні із новими телефонами. У книжці Андрусенка В. П. і Коваленка О. Г. [3] пишуть,

що системи, які стежать за довкіллям, дуже допомагають, бо можуть одразу помітити проблему й передати про неї інформацію, не втрачаючи часу.

Сфера застосування результатів:

1) у Державній службі України з надзвичайних ситуацій (ДСНС) для надання інформації населенню;
2) на підприємствах для захисту тих хто там працює;
3) у побуті (розумні будинки);
4) в транспорті для попередження про аварії чи погодних умов.

У час, коли техногенні аварії стають частішими, важливо мати системи, які швидко попереджають людей про небезпеку. У вступі пояснено, чому ця тема актуальна, визначено мету, завдання, об’єкт і предмет роботи. Запропоновано технічне рішення, яке може стати простим і доступним способом сповіщення під час надзвичайних ситуацій.

1 АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Розкриття об’єкта і предмета дослідження

Ця робота досліджує, як GSM-технології допомагають попереджати людей про небезпеку. В теперішній час небезпека може з’явитися у будь який момент, наприклад: повінь у селі, масштабна пожежа у лісі чи вибух на заводі. У такі моменти кожна секунда має значення. Системи на базі GSM дають шанс негайно зреагувати, вони швидко передають сигнал, щоб люди встигли евакуюватися та захистити своє майно.

Швидкість та надійність саме головне при створені системи. Сирени та проводові мережі вже пройденим етап у нинішні реалії, бо вони часто бувають безсилими у кризових ситуаціях. Наприклад катастрофа сталося в віддаленому районі країни, де немає доступу до інтернету та сигналу тривоги. GSM-технологія допоможе вирішити такі проблеми.

У роботі розбирається, як GSM-модулі дають системам сповіщення працювати без інтернету. Все, що треба – це мобільний зв’язок, до якого є доступ навіть у селах. Якщо сталася небезпека, наприклад, річка затопила дорогу чи сталася пожежа, люди отримують сигнал тривоги на телефон за кілька секунд.

Це допомагає швидко реагувати, врятувати своє життя і речі, та віддалитися з небезпечної зони. У статті Азарової О. В., Гудзя О. В. і Блонського О. В. за 2020 рік пишуть, що GSM-мережі можна налаштувати так, щоб вони стабільно працювали навіть у погану погоду. Верещак Д. С. разом із Калашніковим О. О. зазначили, що такі системи зручні для термінових повідомлень, бо не залежать від Wireless Fidelity (Wi-Fi) чи кабелів. Ці матеріали разом доводять: з такими системами люди мають шанс не пропустити небезпеку та вчасно реагувати на неї [2, 5].

Також для стабільної роботи GSM-систем сповіщення, важливо не лише наявність покриття мобільного зв’язку, а й рівень розвитку мережі. Світ почав активно переходити до 5G, що дає можливість системам покращити свої

характеристики: швидкість передачі сигналу тривоги, стійкість до перенавантажень, більша надійність та саме головне зменшення затримки.

На рис. 1.1 з дослідження [20] зображено прогноз використання 5G до кінця 2025 року. Згідно графіку видно, що розвинені регіони, Азія та Північна Америка, матимуть найбільш з’єднань з 5G, але в країнах Африки та Центральної Азії домінують 3G технології. Це говорить про те, що розробка систем на базі GSM, залишається актуальною, особливо при нестабільному та слабкому інтернеті.

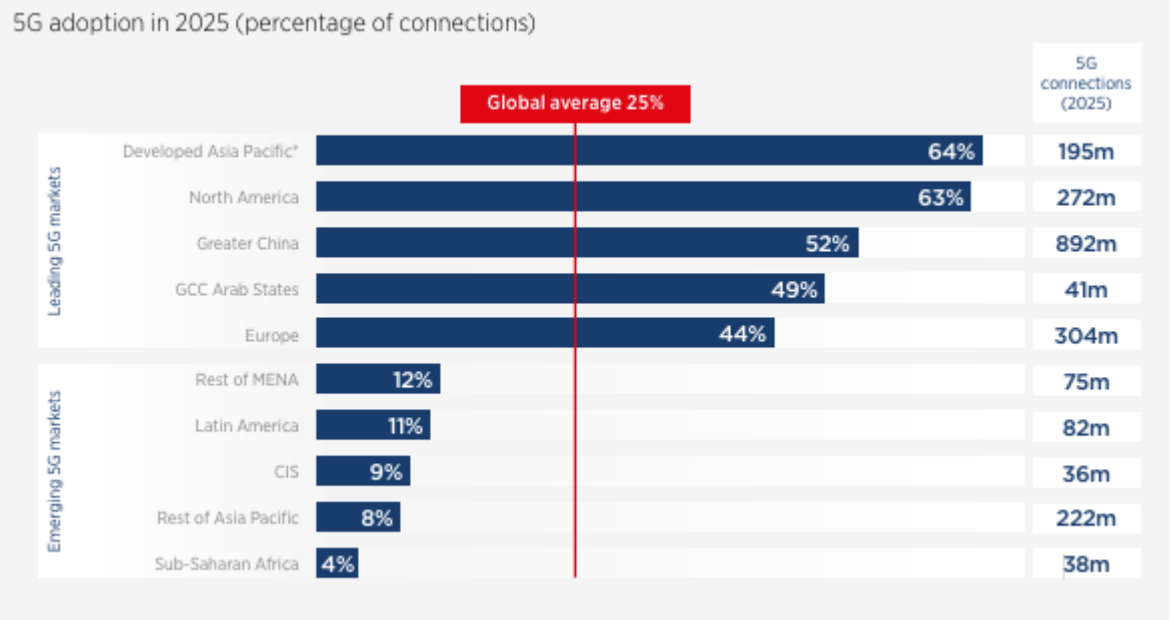

Рисунок 1.1 – Прогноз рівня використання 5G до кінця 2025 року [20]

Виходячи з цих даних стає зрозумілим, що GSM-технології в Україні будуть актуальні й після 2025 року, тому що до кінця 2025 лише 25 % у світі будуть мати 5G, а інші 75 % матимуть 4G/3G/2G з’єднання.

Ще треба звернути увагу на те, як компанії інвестують у розвиток мережі до 5G, як показано на рис. 1.2, більшість капіталу операторів з 2022 по 2025 роки було вкладено саме у 5G, до 85 % від обсягу усіх інвестицій. Але рівень цих інвестицій відрізняється в залежності від регіону. В Північній Америці, Європі та Азії це 80 % і більше, а в Африці, СНД та Латинської Америці менше ніж 75 %.

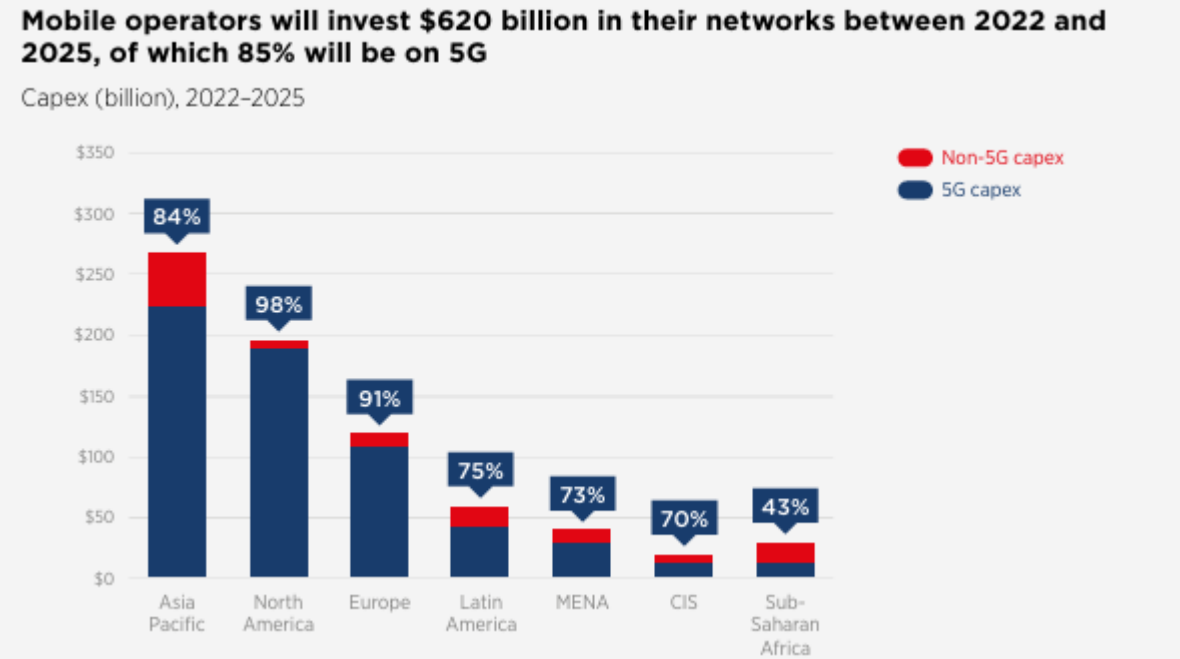

Рисунок 1.2 – Інвестиції у 5G з 2022 до 2025 [20]

Це ще раз підтверджує, що розвиток 5G у світі буде йти поетапно, а отже, системи з GSM/2G/3G будуть актуальні, особливо в тих регіонах, де менше розвинута інфраструктура. В наявних умовах, створення недорогих систем з GSM, досі є актуальним.

Також роль таких систем щодня стає дедалі значущішою у нашому світі, де зміни клімату та техногенні небезпеки приносять нові виклики. Мережі оповіщення є важливі у таких сферах:

1) на заводах і фабриках: система допомагає стежити за небезпечними місцями;

2) у містах та селах: попередження людей про землетруси, цунамі та пожежі, щоб вони могли швидко реагувати на небезпеку;

3) транспорт: повідомляти про будь які аварії, наприклад у метро чи залізничних станціях, щоб пасажири могли швидко зреагувати;

4) у будинках: домашні сигналізації, які рятують життя, при пожежі та витоку газу в квартирі.

Ця робота показує як працює система, що попереджає про можливу небезпеку для людей, які пристрої та програми для цього використовують.

Наприклад у лісі почалася пожежа та датчик фіксує дим, а додаток миттєво про це сповіщає служби та населення поблизу катастрофи. Усе це працює в команді, від отримання даних до подачі інформації на смартфон, щоб люди могли знати про загрозу та могли зберегти своє життя.

Буде досліджено, як пристрій надсилає сигнал людям вчасно, щоб вони могли на це реагувати. Наприклад:

1) програма, яка обробляє інформацію з датчиків;
2) поєднання датчиків (температури, диму, газу тощо) в одну систему;
3) створення способів для передавання і отримання даних.

У дослідження входе те, як зробити систему за допомогою сучасних технологій, яка зможе працювати в різних ситуаціях. Навіть при слабкому мобільному сигналі чи сильних опадах вона має передавати попередження.

1.2 Опис і аналіз структурних та функціональних особливостей об’єкта дослідження

У роботі розбирається, як працює система оповіщення населення, що складається з кількох частин, які разом стежать за небезпекою і надають інформацію, коли стає катастрофа. Наприклад, є сенсорний блок – це група датчиків, які стежать за ситуацією навколо. Їх функції:

1) датчики диму: реагують, коли починається пожежа;
2) датчики температури: включають тривогу, якщо температура вище зазначених показників;
3) датчики руху: реагують на рух;
4) газоаналізатори: якщо в повітрі з’явився чадний газ, починається тривога.

Також є центральний блок – МК, який бере сигнали від датчиків і вирішує, що з ними робити далі. Він мозок усієї системи. GSM-модуль, наприклад, Subscriber Identity Module (SIM) 800L чи Quectel, відправляє повідомлення на телефони людей – чи Short Message Service (SMS), чи дзвінок. Для того щоб, усі

деталі не вимкнулися, коли пропаде світло, ставиться джерело живлення. Зазвичай це акумулятор, але може бути також, що підключають сонячні панелі, якщо місце дозволяє. Така система дає шанс не розгубитися, при надзвичайних ситуаціях [18].

Інтерфейси взаємодії: дають користувачам отримувати дані.

Наприклад:

1) Serial монітор;
2) SMS-повідомлення;
3) NeoPixel (WS2812).

Якщо все це працює разом, система одразу помічає небезпеку й попереджає користувачів.

Функціональні особливості

Система стежить за навколишнім середовищем та аналізує дані які до неї надходять, щоб не пропустити потенційну загрозу. Датчики завжди перевіряють, що діється навколо – можливо аномальна спека, чи витоки газу або вибухи на заводі. Далі мікроконтролер (МК) проглядає ці дані й аналізує, чи є небезпека для населення, щоб вирішити, чи слати сигнал тривоги. Якщо трапляється катастрофа, то через GSM-модуль іде SMS або дзвінок, також йде вивід на Serial монітор та через NeoPixel йде візуальна інформація про небезпеку. Азаров С. І. каже, що такі системи мають бути чіткими, щоб точно виявляти небезпеку [1]. Ще можна зберігати записи подій, щоб потім аналізувати та зробити роботу над помилками.

Трохи окремо про GSM-модуль, бо він є невід’ємною частиною системи оповіщення населення. Він працює майже без обмежень, навіть якщо сигнал слабкий, усе одно відправляє сповіщення про небезпеку. Верещак Д. С. згадує, що такі модулі швидко передають повідомлення, і це дуже зручно [5]. Також, його ціна не дуже висока, що теж плюс.

1.3 Огляд наявних рішень для систем сповіщення

Системи для сповіщення про небезпеку у теперішній час актуальні, щоб люди могли вчасно дізнатися про катастрофу й не постраждати. Через різні небезпеки, наприклад, аварії, сильні повені чи пожежі, важливо, щоб попередження приходили одразу за кілька секунд. Белюченко Д. Ю. з колегами пише, що важливо проаналізувати, які технології краще використовувати, щоб система працювала у місті чи в далекому селі [4]. Огляд того, що вже існує та працює, допомагає зрозуміти, як зробити все надійно та без помилок.

Системи оповіщення населення використовують у багатьох місцях – для безпеки людей, щоб одразу знати про катастрофу і повідомити заздалегідь про це користувачам. Кожна система має свої плюси й мінуси, бо технології та принцип роботи різний. У джерелі [25] кажуть, що головне – це щоб система справлялася з різними ситуаціями. А Закора О. В. додає, що від технологій залежить, наскільки швидко й точно все спрацює [8].

Найкращі та актуальні є дві категорії:

1) системи з GSM – вони популярні, бо працюють навіть там, де погано ловить мобільна мережа. Вони надсилають SMS через GSM-модулі, що зручно для далеких місць, наприклад, у селах;

2) системи на основі Wi-Fi – ці системи швидкі і працюють через інтернет. Але вони підходять лише там, де є Wi-Fi, наприклад, у місті.

Для створення власної системи сповіщення населення, було проведено аналіз існуючих міжнародних рішень, які вже активно використовують в різних країнах. Це дало розуміння, які системи є найефективнішими, які переваги вони мають, та які труднощі можуть виникнути. У табл. 1.1 наведено кілька прикладів систем, та їх характеристика.

Таблиця 1.1 – Системи сповіщення населення і їх різниця

Система сповіщення	**Країна**	**Переваги**	**Недоліки**
WEA (Wireless Emergency Alerts)	США	Безкоштовні повідомлення на телефони, не вимагає додатків	Працює лише на підтримуваних пристроях
NHK Alert System	Японія	Сповіщає про землетруси, до того як вони трапляються	Працює лише на території Японії
eCall	ЄС	Вбудована система в авто, яка викликає допомогу при аварії	Обмежена лише інформуванням про аварії
Cell Broadcast (CB)	Використовується у багатьох країнах	Миттєва розсилка повідомлення про небезпеку у зоні покриття, незалежно від SIM	Потрібна підтримка від мобільних операторів та сумісність з пристроями
Системи оповіщення ДСНС	Україна	Транслює про небезпеку, через сирени, гучномовці, мобільні додатки, радіо чи телевізор.	Застаріле обладнання в деяких регіонах, обмежене покриття
SAGRN Alert SA	Австралія	Надсилає повідомлення при пожежах та техногенних катастрофах	Потребує налаштування пристроїв, SMS може мати затримку

Крім аналізу готових систем, також важливо враховувати сучасні наукові дослідження. По принципу eCall технології, яку використовують в країнах ЄС, у джерелі [15], було розглянуто реалізацію прототипу, щодо використання GSM та Global Positioning System (GPS), на базі Arduino, для скорішого повідомлення

оперативних служб про аварію у реальному часі, та це показало ефективність цієї системи та довело її користь на практиці.

Після спрацювання всіх датчиків, система сповіщення отримувала координати з GPS і передавала їх поліції через SMS. На рис. 1.3 було показано прототип та приклад повідомлення яке відправляє система.

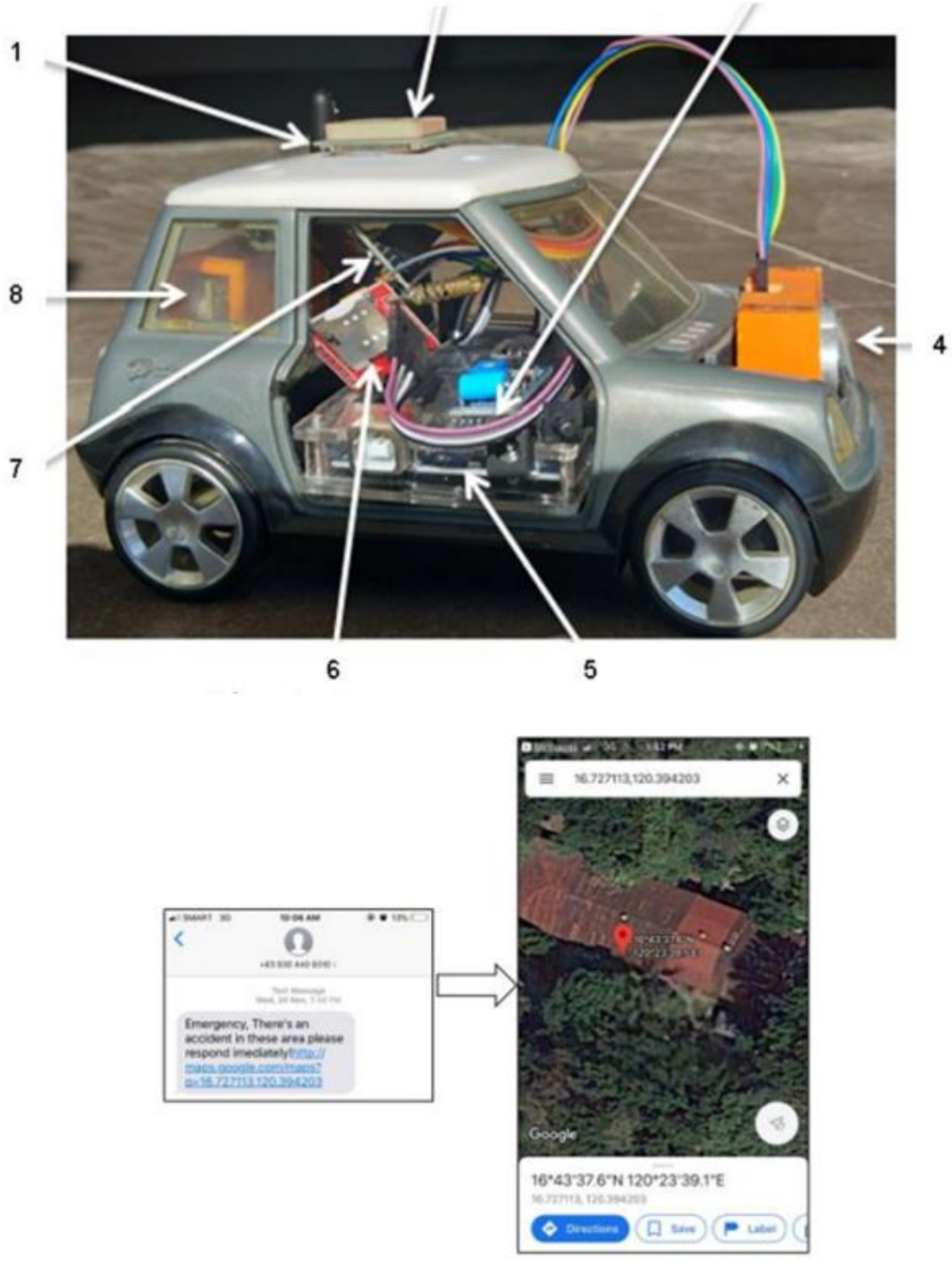

Рисунок 1.3 – Прототип девайсу та SMS повідомлення про аварію через GSM [15]

Також у дослідженні Гоутамі С. було описано та зроблено систему на базі GSM і GPS для сповіщення про небезпеку, яка показана на рисунку 1.4. Основою для системи став МК Arduino UNO, який планується використовуватися при розробці, ще модуль GSM, GPS, 16x2 LCD-дисплей та звуковий сигналізатор. Початком роботи є натискання кнопки тривоги, після якої система визначає місцезнаходження та відправляє SMS повідомлення на мобільний номер телефону [18].

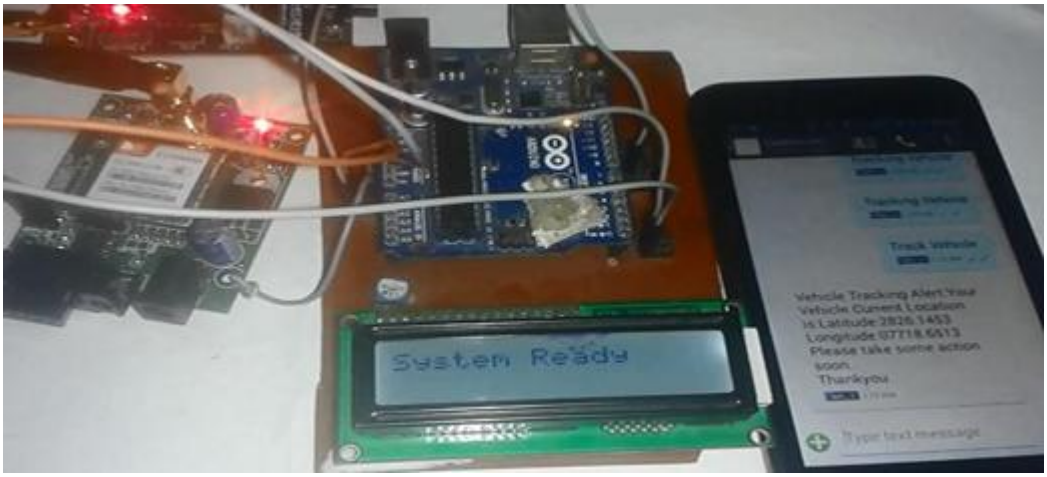

Рисунок 1.4 – Система оповіщення з LCD, GSM, GPS і передачею координат [18]

Виходячи з аналізу існуючих систем сповіщення та наукових досліджень, було виявлено, що система має поєднувати можливість роботи без інтернету, надійний канал зв’язку та автоматичне реагування. Враховуючи досвід сповіщення населення у таких країнах, як США та Японія, а також успішний приклад реалізації на Arduino з GSM модулем, можна сказати, що такі системи доцільно використовувати і в Україні.

1.4 Аналіз та вибір компонентів для системи

Аналіз уже готових систем оповіщення допоміг вибрати потрібні компоненти, щоб зробити свою систему. Обрані елементи повинні бути доступними, енергоефективними та сумісними з Arduino.

Основні деталі системи сповіщення:

1) датчики, які знаходять небезпеку (пожежу, витік газу, високу температуру);

2) контролери, які перевіряють дані від датчиків і вирішують, що робити;

3) модулі зв’язку (GSM, Wi-Fi, Bluetooth), які надсилають повідомлення.

Частіше за все використовують такі компоненти:

1) MQ-2 – щоб знаходити гази;

2) DHT22 – щоб вимірювати температуру і вологість;

3) SIM800L – щоб надсилати sms повідомлення.

У табл. 1.2 наведено опис призначення основних компонентів системи сповіщення та приклади їх застосування на практиці.

Таблиця 1.2 – Основні елементи системи сповіщення та сфери їхнього використання

Компонент	**Опис**	**Приклад використання**
MQ-2	Датчик газу, знаходить чадний газ	Попередження про витік газу
TMP 36	ДТ (датчик температури) і вологості	Виявлення, якщо температура зависока
SIM800L	GSM-модуль, надсилає SMS	Надсилання SMS про небезпеку
Фоторезистор	Датчик освітленості	Автоматичне регулювання яскравості індикації
Arduino UNO	Основний контролер системи	Централізоване керування логікою пристрою
Serial Monitor	Виводить повідомлення користувачу	«Normal mode», «FIRE», «Motion detected» тощо

Також для кращого розуміння, чому були вибрані саме ці компоненти, було створено табл.1.3, де показано їх порівняння з аналогами.

Таблиця 1.3 – Порівняння компонентів з аналогами

Компонент	**Аналог**	**Переваги обраного**	**Недоліки альтернативи**
MQ-2	MQ 135, MQ 3	Чутливий до чадного газу, та дешевше за аналоги	MQ-135 ширший у функціоналу, але дорожчий.
TMP 36	DHT11	Більший діапазон та вища точність	Нижча точність у DHT11
SIM800L	SIM900A, A6	Малий розмір, дешевший	SIM900 складніший у налаштуванні
Фоторезистор	УЗ сенсори	Менше споживає енергії	Уз сенсори дорожче
Arduino UNO	ESP32, STM32	Простота, підтримка Tinkercad	ESP32 потребує складнішого коду
Serial Monitor	LCD-екран	Доступний у Tinkercad, швидкий вивід	LCD не підтримується у Tinkercad

Трохи окремо про Arduino UNO, представлений на рис. 1.5, який було обрано, як основний контролер системи. Він служить центральним МК для виявлення небезпеки, обробки даних та зв’язку з GSM модулем. Arduino UNO полегшує передачу сповіщень про аварії та даних про місцезнаходження в режимі реального часу, що забезпечує швидке реагування на надзвичайні ситуації та підвищення безпеки як зазначено у [18].

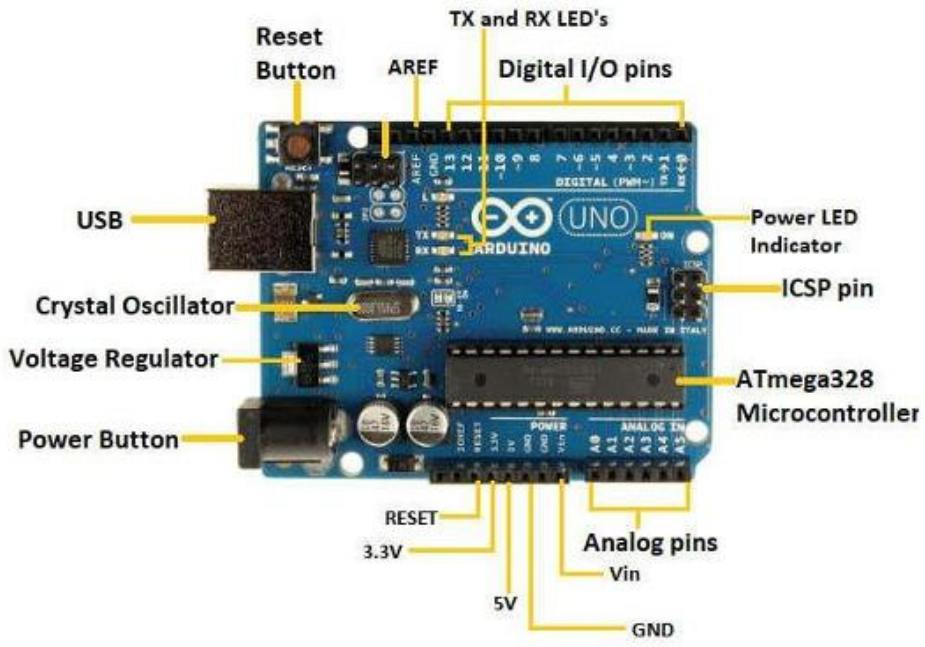

Рисунок 1.5 – Arduino UNO [18]

Детальне пояснення вибору GSM модуля:

Як зазначено у табл. 1.2 для системи сповіщення було обрано GSM модуль SIM800L, але це не єдиний варіант який міг би бути у системі, ще є SIM900A, який був використаний у дослідженні [18] при розробці системи, обидва варіанта показані на рис. 1.6. Обидва модулі мають подібний функціонал, але вибір пав саме на SIM800L, тому що він працює у 4 частотних діапазонах (850/900/1800/1900 МГц), що робить його універсальним для усього світу, коли SIM900A, має лише 900/1800 МГц. Також обраний варіант має меньші габарити та нижче енергозатрат, що сходиться з цілями дослідження. Ще SIM800L показує стабільнішу роботу з Arduino UNO, який також був обраний для дослідження систем сповіщення. І саме головне в нього краща підтримка в спільноті розробників Arduino, це дає змогу робити легшим процес розробки, тому що без проблем можна знайти потрібну документацію, приклади коду, та бібліотеки для модуля. Тому SIM800L частіше використовують у освітніх роботах і проєктах.

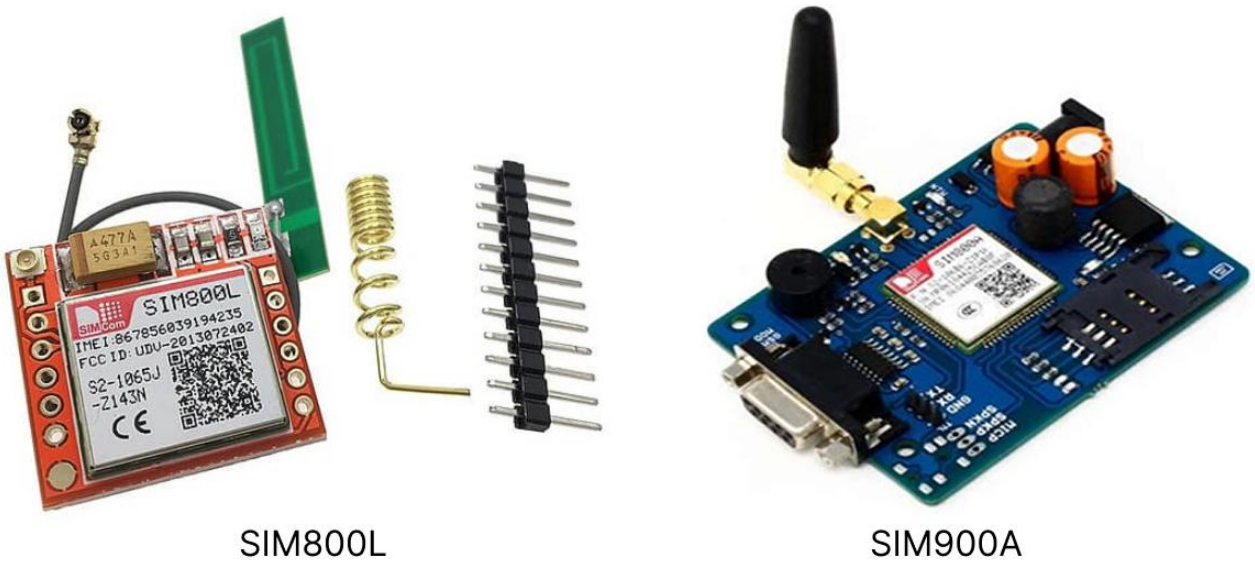

Рисунок 1.6 – GSM модулі SIM800L та SIM900A[18]

Обгрунтування зробленого вибору:

Компоненти для системи сповіщення було обрано не лише через їх доступність, а й для можливості іхнього моделювання у Tinkercad, де буде проводитися тестування системи. Також у майбутньому система може бути масштабована заміною датчику газу MQ 2 на MQ 135 або MQ 3, для більшого роспізнавання газів або ДТ TMP 36 на BME 280 для додаткових налаштувань.

Отже, компоненти вибрано, дивлячись на те, що вже використовують в інших системах. Кожен із них потрібен, щоб система працювала швидко і без збоїв. Наприклад, MQ-2 вибрали, бо він недорогий і добре знаходить гази, а SIM800L – бо він простий у роботі. Усе разом допомагає системі швидко попереджати про небезпеку, наприклад, надсилати SMS, якщо почалася пожежа [20].

1.5 Вибір підходу до виконання завдань

Для стабільної роботи системи оповіщення населення треба вибрати правильний підхід, для її розробки. Вона має бути легкою для людей у використовуванні і стабільно працювати – у місті чи в селі, де зв’язок поганий або зовсім відсутній. Борісова Л. з колегами пише, що важливо все грамотно

спланувати, щоб система не підвела, коли буде небезпека [16]. Якщо подивитися на готові приклади, які були розібрані у підрозділі 1.3 , видно, що такий підхід часто застосовують, і він добре себе показує, як пишуть у [18].

Для створення системи обрано кілька головних та цікавих ідей. Спочатку продумано, щоб кожну частину можна було замінити окремо. Наприклад, якщо ДТ вийде з ладу, його просто міняють на інший, а решта системи продовжує працювати та забезпечувати безпеку. Також треба використовувати відомі технології, такі як GSM і Wi-Fi, їх можна додати у майбутньому, після тестування системи у віртуальному середовищі. Перед запуском систему треба перевірити, щоб упевнитися, що вона реагує на дим чи газ, і робить це вчасно, без великих затримок. Ще треба додати можливість налаштування, щоб користувачі самі вибирали, як працюватимуть датчики.

Такий підхід дозволяє зробити систему практичною та універсальною. Завдяки тому, що частини заміняються окремо, ремонт не стає проблемою. GSM і Wi-Fi забезпечують роботу в різних умовах – від міста до віддалених районів. Перевірка перед запуском дає впевненість, що система не підведе, наприклад, під час пожежі чи витоку газу. А налаштування допомагають кожному користувачу адаптувати її під свої потреби та роблять використання легшим та зрозумілішим, як зазначається в джерелі [20].

1.6 Вимоги до техніки та програм системи

Система сповіщення потребує технічної частини та програм, щоб повідомлення доходили швидко до людей. Сигнал тривоги мусить прийти за кілька секунд, навіть якщо світло вимкнене. Джерело [5] зазначає, що зв’язок має бути надійним, без збоїв. Ці вимоги потрібні, щоб попередження встигали захистити людей та їх майно.

У роботі буде створено модель системи в Tinkercad, яка дозволяє перевірити, як працюють датчики і Arduino, без реальних пристроїв. Хоч GSM-модуля там немає, система все одно буде реагувати на небезпечні ситуації.

Баутіста Дж. із співавторами пишуть, що віртуальні моделі допомагають тестувати системи на Arduino [15].

Функціональні вимоги:

1) автоматично знаходити небезпечні ситуації за заданими параметрами (наприклад, висока температура);
2) швидко створювати і надсилати повідомлення;
3) виведення повідомлень користувачу через серійний монітор.

Нефункціональні вимоги:

1) швидкість роботи(реакція системи не повина бути більше 3 секунд);
2) програму можна запускати в Tinkercad;
3) система дає можливість подальшого розширення у майбутньому, наприклад додати GSM та WI-FI;
4) просте використання.

Спочатку було заплановано додати GSM-зв’язок, але в Tinkercad це не вийде зробити через обмеження програми. У майбутньому, якщо зібрати справжній пристрій, можна підключити GSM-модуль, щоб система стала ще кориснішою.

У цьому підрозділі описано, що потрібно для роботи системи, та для реалізації це в моделі. Система буде працювати сама на Arduino Uno, одразу реагуючи на зміни даних із датчиків. У Tinkercad буде перевірено, як усе діє: як система читає датчики, вмикає світлові сигнали, керує сервоприводом і показує дані на моніторі. Вона повинна працювати без збоїв і готова до покращень, наприклад, до додавання GSM-модуля чи інших частин на справжньому пристрої. Гоутамі С. пише, що віртуальні тести допомагають підготувати системи до реальної роботи [18].

1.6.1 Загальний опис системи

Система сповіщення населення задумана так, щоб одразу повідомляти про небезпеку з мінімальною затримкою у 3–5 секунд, для цього треба дослідити, як

вона буде влаштована. Це дає змогу не втрачати часу, коли стається пожежа чи ламається газова труба, і треба рятувати людей. Завдяки цьому можна зробити так, щоб катастрофа не наробила багато шкоди. У цьому підрозділі пояснено, із чого система складається. Кожна її частина потрібна, щоб усе працювало стабільно та без проблем. Є програма, яка розбирає інформацію, технічна частина, для стабільної роботи, і простий вигляд на Serial моніторі, щоб кожен міг швидко отримати інформацію.

Система має такі частини:

1) програма, яка бере дані, перевіряє їх і показує так, щоб усе було видно;
2) техніка, яке надає безперебійну роботу системи;
3) вигляд на екрані, який не плутає, а допомагає легко розібратися в програмі.

У табл. 1.4 нижче детально розказано, що робить кожна частина системи.

Таблиця 1.4 – Загальна структура АПК

Компонент	**Опис**
МК	Arduino Uno – головна частина для керування всією системою
Датчики	Датчики температури (TMP36), газу (MQ-2), руху (IK), освітлення
Світлодіодна система	Світлодіоди NeoPixel для візуальних попереджень
Сервопривід	Відкриває двері при евакуації або при виявленні руху
Serial монітор	Виводить повідомлення про небезпеку для користувача
Програмна логіка	Алгоритм обробки даних, реагування на тривоги, режим «спокій»

Компонент	Опис
Середовище тестування	Tinkercad – віртуальна перевірка логіки роботи
Потенційні розширення	GSM-модуль, резервне живлення, віддалене керування

Отже, систему продумано так, щоб усі її частини працювали разом й без затримок. Програма має збирати дані, розбирати їх і виводити так, щоб усе одразу стало зрозуміло. Інтерфейс треба зробити легким, щоб кожен без проблем із ним розібрався. Завдяки такому рішенню система не підводить, навіть коли дані надходять у великій кількості.

1.6.2 Вимоги до функціональних характеристик

Для стабільної роботи система має виконувати певні завдання. Вона безперервно стежить за датчиками температури, газу, світла і руху, визначає, чи все гаразд, і реагує залежно від обставин. Усі дані і повідомлення про небезпеку користувач бачить на Serial моніторі. Система працює сама, без потреби з нею взаємодіяти, але одразу повідомляє людину, якщо щось не так. У табл. 1.5 описано, що має вміти система. Азарова О. В. З колегами пише, що автоматична обробка даних робить системи сигналізації зручними і надійними [2].

Таблиця 1.5 – Функціональні вимоги

Функція	Опис	Користувач
Перевірка датчиків	Система весь час зчитує дані з датчиків температури, газу, світла і руху.	Автоматично

Функція	**Опис**	**Користувач**
Оцінка ситуації	Перевіряє, чи є небезпека, аналізуючи показники датчиків.	Автоматично
Світлові сигнали	Показує небезпеку через світлодіоди	Користувач
Керування дверима	Відкриває двері, якщо є тривога чи хтось підійшов до дверей.	Користувач
Показ даних	Виводить на монітор температуру, рівень газу, рух тощо.	Користувач
Попередження про небезпеку	Пише на моніторі, наприклад, «FIRE!», якщо щось не так.	Користувач
Звичайний режим	Показує «Normal mode – EXIT», коли все спокійно.	Користувач
Реакція на рух	Відкриває двері, якщо хтось є біля них і немає тривоги.	Користувач
Плани на майбутнє	Можна додати GSM-модуль або запасну батарею для автономної роботи.	Адміністратор/Розробник

Нефункціональні вимоги:

1) бути швидкою: система повинна швидко виявляти небезпеку та затримка у сповіщенні не повинна бути більшею за 3–5 секунд;

2) стабільно працювати: система повинна весь час спостерігати та стабільно працювати, щоб поломки траплялися не часто, і люди не боялися, що вона раптом перестане робити;

3) ставати на різні комп’ютери: програму треба придумати так, щоб вона без проблем запускалася і на Windows, і на Linux, і будь-хто міг її налаштувати й почати використовувати.

Отже, створена система має робити усе, що потрібно: читати дані з датчиків і попереджати людину, якщо є небезпека. Усе повинно парцювати автоматично, тому зручно і надійно. Її можна буде легко покращити в майбутньому, щоб додати нові функції. Верещак Д. С. пише, що автоматизовані системи стають ефективнішими, якщо їх можна вдосконалювати [5].

1.6.3 Вимоги до зовнішніх інтерфейсів

Система оповіщення населення має бути зручною і зрозумілою для користувача, щоб процес використання системи був без проблем. Для цього потрібні прості способи, щоб зчитувати дані з датчиків, керувати пристроями і показувати інформацію користувачу одразу. Оскільки систему буде протестовано Tinkercad, основну увагу приділено тому, як вона отримує сигнали від датчиків і вмикає світлодіоди чи сервопривід.

Інтерфейс потрібно зробити так, щоб його зрозумів кожен, навіть без технічних знань, це було представлено у табл. 1.6. Повідомлення треба щоб було видно одразу на Serial моніторі, а на небезпеку система реагує сама. Репановичі Р. із співавторами пишуть, що прості інтерфейси роблять системи безпеки доступними для всіх [25].

Виходячи з цього, можна виділити ось такі основні вимоги:

1) зручність: легке та зрозуміле меню;
2) підключення: може підключатися к датчикам та іншим пристроям;
3) надійність: стабільна робота з підключеними деталями у віртуальному середовищі.

Таблиця 1.6 – Зовнішні інтерфейси

Інтерфейс	**Опис**	**Тип**
Апаратні підключення	Підключення датчиків	Аналогові/Цифрові
Світлодіодні стрічки	Передача сигналів керування кольорами та яскравістю	Цифровий (NeoPixel)
Сервопривід	Відкриття/закриття дверей	PWM
Serial монітор	Виведення повідомлень користувачу в реальному часі	UART (Serial)
Tinkercad середовище	Віртуальний інтерфейс тестування системи	Вбудований симулятор

Загалом система повинна працювати з користувачем і навколишнім середовищем через найпростіші з’єднання. Усі частини системи пов’язані між собою проводами і швидко передають дані без помилок. У майбутньому, після віртуального тестування системи в Tinkercad можна додати GSM-модуль або Bluetooth, щоб керувати системою здалеку чи надсилати повідомлення. Чоухан П. зазначає, що GSM-модулі роблять системи оповіщення зручнішими для віддаленої роботи [17].

1.7 Формування специфікації вимог

Формування специфікацій вимог є одним із головних етапів, перед розробкою системи оповіщення населення. На цьому етапі визначаються функціональні та нефункціональні характеристики, яких треба дотримуватися для досягнення мети.

Щоб система працювала стабільно і без проблем, у специфікації враховано важливі деталі, наприклад, як швидко вона реагує і чи довго працює без електроживлення. У табл. 1.7 розказано, які саме вимоги було вибрано, щоб система була зручною і не підвела у важливий момент, наприклад, у місті чи селі, коли буде небезпека. Також додано вимоги до безпеки, щоб дані ніхто не перехопив.

Таблиця 1.7 – Специфікація вимог до системи

Розділ	**Опис**
Призначення	Автоматичне сповіщення про наявність небезпеки
Межі використання	Внутрішні приміщення, моделювання на макеті
Вимоги до апаратного забезпечення	Arduino, датчики газу, температури, освітлення, ІЧ-датчик, сервопривід
Вимоги до програмного забезпечення	Програма на Arduino C/C++, обробка даних у режимі реального часу
Надійність системи	Висока доступність і мінімальний час простою системи
Безпека	у майбутньому – через GSM/шифрування
Енергозабезпечення	Працює від USB або батареї; підтримка PowerBank для автономності

Розділ	Опис
Чутливість	Визначення небезпеки за 1-3 секунди після зміни значень
Масштабованість	Визначення небезпеки за 1-3 секунди після зміни значень
Інтерфейс користувача	Серійний монітор (Serial Monitor) Arduino IDE
Сумісність	Підключення до Arduino Uno/Nano, емуляція в Tinkercad

Отже, треба щоб увесь процес дослідження був детально продуман, для того щоб, зробити систему сповіщення та вона не виходила з ладу і була максимально ефективною. Наприклад, завдяки GSM-зв’язку вона може надсилати повідомлення навіть там, де погано працює сигнал. Це допомагає в різних випадках. А якщо система економить заряд, то працюватиме довше без підключення до живлення. Обладнання має бути міцним, щоб не псувалося від спеки, дощу чи поганого зв’язку. Від цього залежить, як довго система буде використовуватися.

Висновки до розділу 1

Зробивши аналіз, як працюють сучасні системи сповіщення, стає ясно, що вони дуже потрібні, щоб швидко реагувати, коли стає катастрофа. Вони одразу помічають проблеми та дають сигнал оперативним службам. Завдяки цьому можна встигнути попередити всіх людей, щоб жертв було як умога менше.

Програми й пристрої, які використовують GSM-мережі, добре справляються з тим, щоб повідомляти про небезпеку. Це дозволяє завжди бути на зв’язку і надсилати повідомлення без проблем, що особливо важливо, коли мало часу.

Ознайомившись з різними системи сповіщення, то GSM-модулі виглядають краще за всіх. Їх легко вдосконалювати, вони зрозумілі для людей і

коштують недорого. Такі модулі дають змогу тримати зв’язок навіть у місцях, де інші способи не здатні функціонувати. Тож можна зробити систему, яка і не підведе, і не коштує дорого.

Щоб система вийшла надійною, треба спочатку розібратися, що вона має вміти. У роботі описано головне: щоб сама попереджала про небезпеку, могла зв’язатися різними способами, наприклад, через повідомлення чи дзвінки, і працювала разом із датчиками чи іншими програмами, щоб усе відбувалося швидко. Ще подумано про те, щоб вона не зависала, не витрачала багато електроживлення і була простою для всіх. Історія з аварією на Чорнобильській АЕС нагадує, що такі системи дуже допомагають, коли стається якась небезпека.

На даний момент вже визначено основні характеристики, на яких буде працювати система. Виявилося, що краще використовувати комбінований підхід, де GSM-технології поєднуються з іншими способами передачі даних, наприклад, через Wi-Fi або Bluetooth. Це допоможе швидше реагувати і зробити так, щоб повідомлення доходили без помилок.

Те, що придумано й перевірено, стане основою для подальшої роботи. Ці ідеї пояснюють, якою має бути система, і дають змогу створити її макет, щоб усе протестувати. Наприклад, якщо продумати, щоб система не ламалася і не брала багато енергії, вона зможе діяти без проблем, навіть коли ресурсів мало.

Завдяки GSM-технологіям система може швидше попереджати про катастрофу і залишатися на зв’язку, навіть якщо умови зовсім погані, а звичайні способи зв’язку не працюють.

Отже, дослідження дало теоретичну базу для наступних етапів розробки системи. Завдяки аналізу структури і головних функцій було сформовано чітке завдання для створення програми. Визначені вимоги допоможуть зробити наступні етапи проєктування і тестування простішими, щоб у підсумку вийшла система, яка швидко реагує на небезпеку і вчасно попереджає людей.

2 МАТЕМАТИЧНІ МЕТОДИ ТА АЛГОРИТМИ РОЗРОБКИ СИСТЕМИ

2.1 Постановка завдання

У розділі буде визначено, як зробити систему, щоб вона швидко й без проблем попереджала людей про небезпеку. Розроблено план, який враховує, де і як використовуватимуть алгоритми, та спосіб, щоб система робила все автоматизовано, без втручання людей. Якщо з’єднати пристрій з програмою, вийде надійна система яка буде простою у використанні для всіх. Це дуже потрібно, коли треба терміново повідомити про катастрофу. Так інформація дійде швидко, і вдасться вчасно відреагувати на будь-яку небезпеку [4, 5, 18].

Треба використовувати сучасні способи зв’язку, наприклад, GSM, адже вони дають змогу надсилати повідомлення чи телефонувати навіть із місць, де сигнал слабкий [18]. Оскільки ці деталі стають усе кращими, можна навчити систему самій повідомляти про катастрофу. Це допомагає швидше діяти, коли щось стається. Тому розробка плану й способи використання для таких автоматичних повідомлень – це великий крок до створення систем, які не підведуть у складний момент.

Головне завдання створити систему, яка буде:

1) точно обробляти дані та робити вірні розрахунки;
2) стабільно працює навіть у складних умовах;
3) ефективно використовує обладнання;
4) зручний та зрозумілий інтерфейс.

Система, після дослідження, має бути добре захищеною за новими правилами, працювати в різних умовах і давати можливість її поліпшувати потім у майбутньому.

Чому це потрібно: Така робота з’явилася, бо є кілька великих проблем, які треба вирішити:

1) старі методи, багато компаній досі використовують застарілі методи сповіщення, і це виходить повільно й не дуже точно;

2) не автоматизована робота, деякі функції у сповіщенні про небезпеку люди роблять самі, без використання програми, цей процес довший і можуть бути помилки;

3) проблеми з програмами, ті програми, що є зараз, не завжди підходять для роботи і їх важко з’єднати між собою.

Створення нової системи допоможе позбутися цих труднощів, покращить функціонал і зекономить гроші на обладнанні.

Що треба зробити: Потрібно придумати спосіб, щоб система працювала за допомогою нових комп’ютерних програм і розумних розрахунків. Цей спосіб має вміти:

1) визначення даних: з'ясувати, які дані потрібні для вирішення завдання;

2) перевірка даних: перевірити, чи правильні ці дані;

3) обробка даних: зробити аналіз інформації за допомогою формул і методів;

4) отримання результатів: показати результати у зрозумілому вигляді;

5) автоматизація: зменшити роль людини в роботі системи.

Для виконання цих функцій створено алгоритм, який показаний у вигляді блок-схеми.

Щоб розібратися, як усе влаштовано, зроблено схему, яку можна розглянути на рис. 2.1. Там показано, послідовність та які етапи проходить система. Схема допомагає зрозуміти, що потрібно робити системі.

Блок-схема має такий зміст:

1) початок: включення системи і перевірка усіх налаштувань;

2) введення даних: отримання інформації від користувача або зовнішніх джерел;

3) обробка: перевірка чи правильні дані, обчислення результатів;

4) аналіз результатів: перевірка чи відповідають вони вимогам;

5) показ результатів: передача інформації користувачу;

6) завершення: закриття програми та збереження результатів.

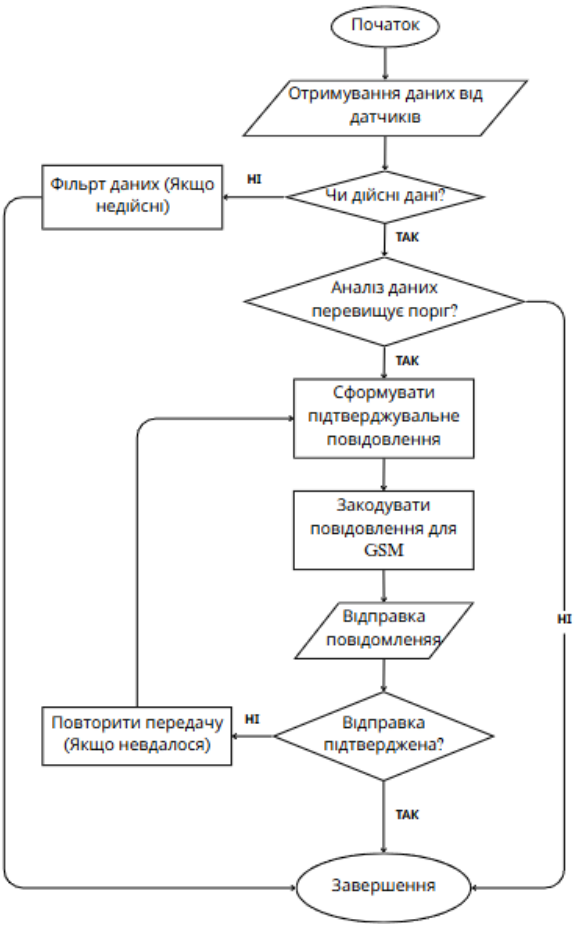

Рисунок 2.1 – Блок-схема алгоритму роботи системи

***Детальний опис створеного алгоритму*:**

Система запускається та отримує дані від датчиків. Спочатку вона перевіряє, чи вони правильні. Якщо дані показуються з помилкою, то їх фільтрують. Якщо все гаразд, система аналізує, чи перевищують вони допустимий рівень.

Коли перевищення не має, робота завершується. Але у разі виявлення порушень робиться закодоване повідомлення та надсилається через GSM та Serial монітор. Далі система очікує підтвердження відправки.

Якщо отримано підтвердження, то процес буде завершуватися. При невдачі виконується повторна передача, яка триває поки, поки не буде успіху або не вичерпається ліміт спроб.

Отже розглянуто, як зробити систему, яка працює з GSM-технологіями, щоб автоматично вирішувати потрібне завдання. Як усе відбувається: від того,

як система збирає дані, до того, як видає результат, який влаштовує користувачів і не дає системі зависнути. Азарова О. В. із колегами пише, що такі етапи важливі для стабільної роботи [2].

Запропонований алгоритм показано у вигляді блок-схеми, де видно, які кроки треба пройти, щоб перевірити дані й отримати результат. Верещак Д. С. зазначає, що такі схеми допомагають зробити аналіз даних швидшим і точнішим [5]. Ця схема буде основою для програми, яка працюватиме швидко й зручно для людей.

Отже, усе, що описано в підрозділі – завдання, функції й алгоритм, – дає змогу рухатися далі й почати розробляти методи й моделі для системи. Gowthami S. пише, що використання закордонного досвіду й нових технологій, типу Internet of Things (IoT), може сильно покращити систему [18].

2.2 Оцінка ефективності обробки даних у системі сповіщення

Оцінку ефективності обробки даних використовують, коли йде розробка системи. Вона допомагає розібрати процес обробки, коли умови різні, і перевірити, чи система взагалі добре функціонує. Борісова Л. пише, що оцінка ефективності дає змогу краще зрозуміти, як система себе покаже у різних випадках, і це потрібно, щоб вона не мала затримку й працювала надійно [16].

Також, це дозволить не витрачати багато часу на реальні тести, бо можна просто перевірити на комп’ютері, як все буде функціювати. Дивень В. І. зазначив, що це економить час і допомагає попробувати різні варіанти без проблем [6]. У статті від GSMA ще додають, що моделі спрощують керування системою й дають змогу швидше реагувати, якщо відбувається катастрофа [20]. Моделювання показує, як виглядає завдання в реальності й що в системі найважливіше. Загалом, це показує, як система має працювати, щоб усе було без помилок.

Оцінка ефективності обробки допомогає вирішувати такі завдання:

1) створити основи для розробки алгоритмів обробки даних;

2) передбачити, як система буде реагувати при зміні даних;
3) краще використовувати ресурси системи.

За цілями проєкту, обробка має відповідати двом головним вимогам: точно показувати основні частини системи і при цьому бути простою у розробці. Закора О. В. у своїй роботі пише, що можна зробити модель і точною, і не дуже складною, якщо поетапно її покращувати [8]. На старті моделювання треба добре розібратися з даними, які система отримує, бо від їхньої якості залежить, чи буде вона стабільно працювати. Усе це розписано в табл. 2.1, де видно, як дані впливають на систему, як зазначено в [18].

Дані мають наступні властивості:

1) змінність: дані можуть змінюватися залежно, як працює система;
2) невпорядкованість: інформація може бути в різних форматах;
3) обмеженість: є допустимі значення для даних.

Таблиця 2.1 – Характеристики вхідних даних для математичного моделювання

Параметр	**Тип даних**	**Опис**	**Приклад значення**
Температура	Числовий	Значення з температурного сенсора (TMP36)	28 °C
Рівень газу	Числовий	Напруга або цифрове значення з газового сенсора (MQ-2)	600 умовних значень
Освітленість	Числовий	Значення з фоторезистора, конвертоване в яскравість	150 умовних значень
Наявність руху	Логічний	Значення з ІЧ-датчика руху (1 – рух виявлено, 0 – ні)	1 умовних значень
Стан системи	Текстовий	Опис стану системи залежно від ситуації	FIRE!, EXIT

Після аналізу вхідних даних визначається завдання та описуються основні процеси.

Етапи вирішення завдання виглядають так:

1) збір даних: введення даних користувачем або імпорт з інших джерел;
2) попередня обробка: перевірка правильності даних і видалення зайвих записів;
3) основна обробка: обчислення або аналіз даних за заданими правилами;
4) результати: збереження даних і передача їх користувачеві.

Кожен етап системи розбито на окрему частину алгоритму, а потім усе це зроблено в одну модель. Треба придумати алгоритми, які б ясно показували, послідовність дій системи, щоб усе працювало без помлок. Андрусенко В. П. пише, що без нормальних алгоритмів інформаційна система не буде стабільно працювати [3].

При оцінки ефективності обробки важливо зауважити:

1) швидкість обчислень: спрощення алгоритмів для прискорення роботи;
2) масштабованість: забезпечення роботи системи з великими об'ємами даних.

Для забезпечення стійкості, масштабованості та ефективності моделі можна використовувати такі підходи:

1) хеш-таблиці для швидкого пошуку;
2) сортування для зменшення кількості порівнянь;
3) паралельна обробка для прискорення роботи.

Система повинна враховувати реальні проблеми, наприклад, коли дані змінюються або GSM-мережа повільно передає інформацію. У статті [20] пишуть, що це може впливати на роботу. Дивень В. І. пояснює, що адаптивні алгоритми корисні, бо вони допомагають системі швидко підлаштуватися під зміни в мережі й краще обробляти дані [6].

Щоб зрозуміти, наскільки швидко працює система сповіщення про катастрофи, можна використати просту формулу. Припустимо, (N) – це скільки

сигналів система обробляє за якийсь час, а $T_{обр}$ – скільки секунд іде на одну заявку. Тоді загальний час для всіх заявок $T_{заг}$ визначається за формулою 2.1:

$$T_{заг} = N \cdot T_{обр}. \quad (2.1)$$

Наприклад, якщо система обробляє 50 подій (N=50), а середній час обробки однієї заявки становить 2 секунди ($T_{обр}$ = 2), то загальний час обробки дорівнює $T_{заг} = 50 \times 2 = 100$ секунд, тобто 1 хвилина 40 секунд. Це показує, чи встигає система обробляти все швидко. Якщо заявки бувають термінові і звичайні, можна додати коефіцієнт (P), щоб врахувати їхню важливість. Для термінових P = 0.5, а для звичайних P = 1. Тоді час заявки з урахуванням її типу та особливості $T_{пріор}$ буде рахуватися за формулою 2.2:

$$T_{пріор} = T_{обр} \cdot P. \quad (2.2)$$

Наприклад, якщо звичайна заявка обробляється 2 секунди, то для термінової вийде $T_{пріор} = 2 \times 0.5 = 1$ секунда. У джерелі [5] пишуть, що це допомагає системі швидше реагувати на важливі ситуації, не витрачаючи зайвих ресурсів. Щоб система могла працювати з великою кількістю даних, треба порахувати, скільки часу займе їхня обробка. Нехай (V) – це скільки всього записів у базі, а (S) – скільки записів система обробляє за секунду. Тоді час для всіх даних $T_{дані}$ рахується за формулою 2.3:

$$T_{дані} = \frac{V}{S}. \quad (2.3)$$

Наприклад, якщо записів 10 000 (V = 10 000), а система обробляє 500 записів за секунду (S=500), то виходить $T_{дані} = \frac{10000}{500} = 20$ секунд. Це показує, як багато даних впливає на швидкість роботи, що важливо для реальних умов, як сказано у джерелі [23]. Такі розрахунки допомагають зрозуміти, як налаштувати систему, щоб вона не гальмувала навіть при великих навантаженнях, і забезпечити її стабільну роботу в різних ситуаціях.

У цьому підрозділі розібрано, що треба для алгоритмів системи сповіщення, яка працює з GSM і GPS. Верещак Д. С. у своїй роботі пише, що

оцінка ефективності обробки даних допомагає розібратися, як система буде діяти, і прискорити обробку даних [5]. З’ясовано, що для цього треба кілька кроків: спочатку визначити, що за небезпека трапилась, потім розставити, що важливіше, і вибрати найшвидший спосіб відреагувати.

Дані треба налаштувати так, щоб система не мала затримку, коли приходить багато заявок. Коваленко Р. І. пояснює, що якщо правильно розподілити завдання, то можна швидко реагувати на надзвичайні ситуації [10]. Щоб алгоритми не підводили, кожен етап обробки даних має бути чітко розписаний. Коваленко Р. І. ще додає, що це допомагає системі тримати навантаження, навіть коли багато запитів [10].

2.3 Алгоритми обробки даних

У підрозділі 2.3 показано, які мають бути алгоритми для системи екстреного сповіщення, що працює через GSM і GPS. Моделі допомагають зрозуміти, як усе буде працювати, і зробити обробку даних швидшою, як пишуть у роботі [18]. Щоб система реагувала швидко, дані треба обробляти точно, а датчики – наприклад, температури чи диму – мають одразу відправляти інформацію через GSM, без затримок.

Збір і відправка даних відіграють важливу роль, щоб система могла стабільно реагувати на різні катастрофи, про це пишуть Кутусов М.В. [11]. Алгоритми мають бути готові до того, що зв’язок може бути нестабільним чи мережа перевантажена. Верещак Д. С. і Калашніков О. О. пишуть, що тут виручають методи, які швидко й чітко обробляють багато даних [5].

У цьому підрозділі розібрано, як ці алгоритми допомагають із екстреними сповіщеннями, щоб система реагувала на надзвичайні ситуації, наприклад, природні катастрофи чи аварії, якомога швидше й без помилок.

Як пише Коваленко Р. І. [10], системи, які самі аналізують дані з датчиків, дуже важливі, бо вони швидко знаходять загрози і допомагають вирішувати, що робити.

Основні етапи обробки даних включають:

1) ***збір даних*** – це перший і важливий етап, бо інформація приходить від різних датчиків, наприклад, температури, диму чи руху. Вони стежать за тим, що відбувається навколо. У віснику студентської конференції Азаров С. І. [1] пише, що якщо правильно зібрати і обробити дані, то можна швидко зреагувати на небезпеку. Це правда, бо вчасний збір допомагає одразу помітити проблему, навіть якщо зв’язок поганий;

2) ***передача даних –*** це процес коли інформація відправляється через GSM-мережу до центру або служб. Вона дійти швидко і без збоїв, навіть якщо мережа перевантажена, щоб усе спрацювало вчасно [20];

3) ***обробка даних*** – система аналізує дані і вирішує, яка небезпека сталося, наприклад, це пожежа, аварія, чи інше. Алгоритми мають зрозуміти, що відбувається, і вирішити, що робити далі. У матеріалах [23]. пишуть, що обробка в реальному часі – це основа, щоб швидко реагувати на проблеми, а датчики допомагають усе правильно розпізнати;

4) ***оповіщення*** – це процес попередження людей або служби про небезпеку, наприклад, через SMS або дзвінки по GSM. Верещака Д. С. і Калашніков О. О. [5] кажуть, що швидкість і надійність тут дуже важливі, бо це може врятувати багато життів;

5) ***обробка помилок –*** система стабільно працює, навіть якщо сталося перенавантаження. Алгоритми мають самі знаходити і виправляти проблеми, особливо коли зв’язок поганий або багато даних, бо це важливо для нормальної роботи [2].

Усі етапи обробки даних пов’язані між собою й разом роблять систему, яка може швидко реагувати на небезпеку. Спочатку сенсори збирають інформацію, щоб зробити аналіз. Потім дані йдуть через GSM-мережу, щоб швидко дістатися, до користувача. Обробка й сповіщення допомагають службам одразу включатися у роботу. Ще система вміє справлятися з помилками, щоб не зупинятися, навіть якщо є якісь проблеми. Наприклад, якщо сенсори помічають щось дивне, система

автоматично починає діяти, щоб не витрачати час й передати все без помилок. Це дуже важливо в реальному житті, бо кожна секунда може багато вирішити.

Для більшого розуміння, на рис. 2.2 намалювано частину алгоритму обробки даних. Вона показує, послідовність процесів.

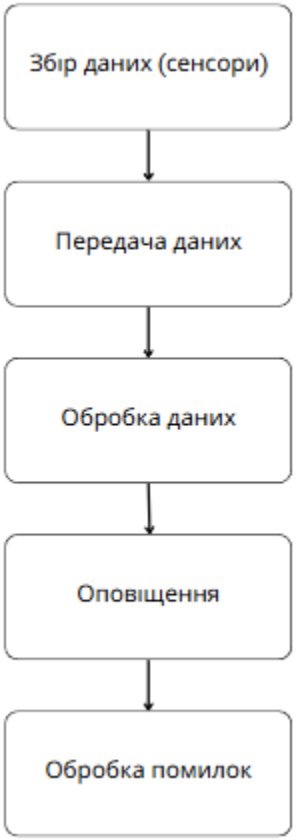

Рисунок 2.2 – Частина алгоритму збору та обробки даних

Деталізована схема ключових процесів:

Друга схема розбирає на рис. 2.3, як у системі йдуть основні процеси з обробкою даних. Вона показує все у послідовності: як інформація надходить з сенсорів, відправляється через GSM, як виправляються помилки й як алгоритми вирішують, що робити при різних небезпеках.

Ця схема показує, як система справляється, коли треба терміново обробити дані й відправити їх. Наприклад, якщо трапляється небезпека, алгоритм швидко визначає, чи слати сигнал про проблему на заводі, чи попереджати про повінь. Через це система діє не повільно, а швидко й по ділу, що в критичних випадках може багато змінити.

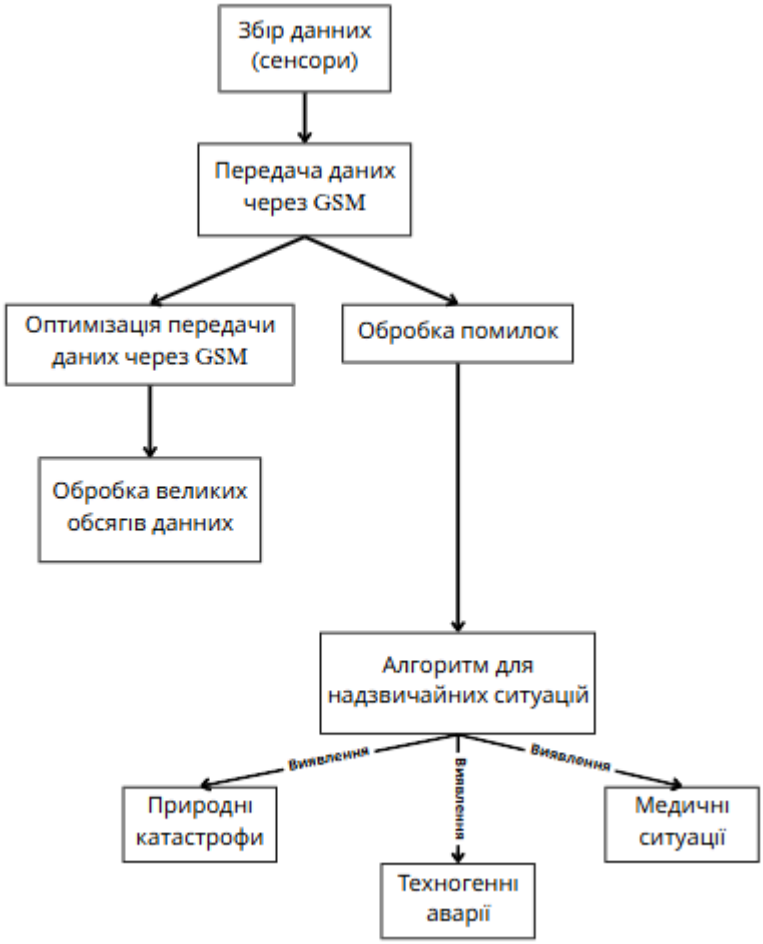

Рисунок 2.3 – Деталізована схема ключових процесів

Логіка збору даних сенсорів:

Сенсори в системах моніторингу грають ключову роль, бо вони збирають дані з різних датчиків – температуру, дим, рух чи тиск. Завдяки сенсорам одразу видно, якщо щось трапилось, і можна швидко почати діяти, коли вони фіксують рух, тиск чи інші варіанти небезпеки на старті роботи. Сенсори одразу отримують сигнал, якщо стає катастрофа і треба одразу діяти. У статті про GSM-систему для попередження про природні біди сказано, що з сенсорами все виходить швидше й чіткіше [18]. Усю ігформацію, що вони зібрали, направляють у головну систему, щоб там аналізували ці дані. Ще в статті про екстрені сповіщення через GSM зазначають, що коли дані з сенсорів об’єднують, то легше зрозуміти, що трапилось, і швидше можна зреагувати на тривогу [18].

Передача даних через GSM:

У системах екстреного сповіщення дані з сенсорів йдуть через GSM-мережу до головного центру чи служб, які повині реагувати на катастрофу. Швидкість і надійність тут дуже важливі, бо від цього залежить, як швидко інформація дійде до потрібних осіб. Якщо зв’язок не стабільний чи погана мережа, система має мати алгоритми, щоб дані все одно доходили до відповідальних осіб. Верещак Д. С. із Калашніковим О. О. пишуть, що це допомагає не втрачати часу [5]. Gowthami S. додає, що такі алгоритми дозволяють системі працювати без зупинок, коли кожна хвилина на вагу золота [18].

Обробка помилок:

Коли йде процес передачі даних, можуть виникнути проблеми, такі як втрата зв’язку чи зіпсована інформація. Щоб система не відмінила передачу інформації, використовують програми, які знаходять і виправляють помилки. Такі програми можуть ще раз відправити дані або попередити відповідальну людину, про небезпеку. У статті про екстрені сповіщення через GSM пишуть, що це зменшує шанси втратити дані й робить систему надійнішою [17]. Ще в дослідженні про сповіщення при природних катастрофах додають, що це допомагає тримати все під контролем [2].

Особливо важливо розібратися з помилками, коли мережа не стабільна або перевантажена. Для цього є методи, які перевіряють дані, відправляють їх ще раз, якщо щось не вийшло, і стежать за потоком інформації. Іноді додають технології, наприклад GPS чи інші канали зв’язку, щоб проблем було менше. У статті про систему сповіщення для аварій на основі Arduino зазначають, що це дозволяє системі працювати стабільніше [17].

Алгоритм 1: Збір даних та передача через GSM

1) отримання даних від сенсорів (температура, дим, тиск, рух);
2) перевірка даних на правильність;
3) передача через GSM до відповідної служби для подальших дій;

Алгоритм 2: Обробка помилок

1) аналіз на помилки;
2) виправлення помилок;
3) оповіщення оператора, якщо неможливо виправити помилки.

Алгоритми для різних варіантів надзвичайних ситуацій

Природні катастрофи:

Щоб виявляти природні катастрофи, такі як землетрусів, повеней чи сильних вітрів, використовують алгоритми, які розбираються з даними від сейсмічних датчиків, приладів для рівня води й метеостанцій. Це допомагає передбачити, що буде далі, і вчасно попередити служби. Андрусенко В. П. із Коваленко О. Г. пишуть, що для кращої точності додають методи машинного навчання й моделі, які прогнозують події [3]. У матеріалах [13] і статті [15] теж кажуть, що це робить прогнози надійнішими.

Техногенні аварії:

Для реагування на техногенні аварії (вибухи, витоки газу, хімічне забруднення) застосовуються сенсори, які відстежують температуру, тиск і концентрацію шкідливих речовин у повітрі. Алгоритми аналізують ці дані, визначають проблему і передають повідомлення через GSM-мережу відповідним службам. Для забезпечення надійності передачі в умовах проблем зі зв’язком система використовує кілька каналів зв’язку [20, 25].

Медичні ситуації:

Для стеження за медичними проблемами, наприклад серцевих нападів, інсультів чи травм, ставлять сенсори, які слідкують за пульсом, тиском і іншими показниками. Якщо трапляється критична ситуація, система одразу відправляє дані через GSM до швидкої, та пише, де людина знаходиться. Верещак Д. С. із Калашніковим О. О. кажуть, що це допомагає вчасно приїхати й врятувати людину [5].

Обробка великих обсягів даних:

Алгоритми мають швидко розбиратися з великою кількістю даних від різних сенсорів і не витрачати часу на реакцію. Це особливо потрібно, коли до

системи надходить багато сигналів одночасно, наприклад, під час катастрофи. Як зазначають Верещак Д.С. і Калашніков О.О. , система має бути налаштована на оперативну обробку великих обсягів інформації та надійне реагування в будь-яких умовах [5].

Отже, алгоритми для екстрених сповіщень дуже допомогають, бо дозволяють швидко розібратися з катастрофою й не пропустити небезпеку. Вони дбають, щоб дані правильно оброблялися, без затримок передавалися через GSM-мережі. У Законі України «Про мобільний зв’язок та телекомунікації в Україні» пишуть, що мережі мають чітко працювати, щоб екстрені повідомлення доходили без проблем [7]. Якщо до GSM додати IoT, то обробка даних виходить кращою, а алгоритми, заточені під різні аварії чи катаклізми, роблять реакцію швидшою й точнішою.

2.4 Технічна реалізація алгоритму

Щоб алгоритм системи оповіщення правильно функціонував, треба з’єднати технічну частину з програмою так, щоб увесь процес йшов без збоїв. Основна мета – з’єднати датчики, контролер, модулі зв’язку і систему сповіщення, щоб вони працювали разом як єдина система. У цьому підрозділі розібрано, як влаштована технічна частина системи, як її елементи взаємодіють, а також як передаються дані і захищається інформація.

Апаратна структура системи

Основними компонентами системи є:

Датчики (сенсори). Вони потрібні, щоб виявляти небезпеку на об’єкті, який охороняється. Ось які датчики зазвичай використовують:

1) pir-сенсори (пасивні інфрачервоні датчики руху). Реагують на тепло від об’єктів, що рухаються. Про це пишуть Дивень В. І. і Ковалишин В. у [6];

2) газовий сенсор MQ-2 – реагує на дим та чадний газ, при перевищенні порогу включається тривога;

3) температурний сенсор TMP36 – аналізує рівень темпаратури навколо та якщо буде перевищення показників то система включає аварійний режим;

4) фоторезистор – визначає рівень освітлення та регулює яскравість підсвічування системи.

Основна частина, яка все обчислює, – це МК Arduino. Він робить такі функції:

1) отримує сигнали від датчиків і обробляє їх;
2) дивиться на дані і вирішує, чи є загроза;
3) вибирає, чи вмикати сповіщення і що робити далі;
4) система дбає, щоб повідомлення дійшли до людей, які її використовують, і до тих, хто нею керує;
5) виводить всю інформацію на серійний монітор.

Яку платформу вибрати, залежить від того, наскільки складний алгоритм і скільки датчиків потрібно під’єднати. Для простих систем можна взяти ESP8266 або Arduino (який було обрано для дослідження та описано у розділі 1), бо вони економлять енергію і їх легко налаштувати, про що пише Верещак Д. С., [5]. А якщо треба щось потужніше для складніших завдань, краще взяти Raspberry Pi або STM32.

GSM-модуль потрібен, щоб швидко повідомити власника чи охорону через мобільну мережу.

Що він вміє:

1) надсилає SMS, якщо помітили загрозу;
2) автоматично дзвонить на номери, які заздалегідь записали;
3) передає дані в хмару чи на сервер для подальшої обробки, як пишуть Верещак Д. С. і Калашніков О. О. у [5].

Живлення системи:

Щоб усе працювало без перерв, систему підключають до розетки (220В) і ставлять запасний акумулятор. Зазвичай беруть:

1) літій-іонні акумулятори – вони допомагають системі працювати, якщо світло вимкнули;

2) системи безперебійного живлення (UPS) – підтримують стабільну роботу головного контролера.

Принципи функціонування системи

Основні кроки роботи системи оповіщення:

1) ***моніторинг середовища:*** система весь час перевіряє дані від датчиків. Контролер аналізує, чи щось змінилося, і робить перші висновки про те, що відбувається на об’єкті;

2) ***перевірка сигналів:*** дані порівнюють із заданими значеннями. Ще йде аналіз, чи кілька датчиків спрацювали одночасно – це допомагає не реагувати на помилки;

3) ***оцінка рівня загрози:*** якщо система бачить реальну небезпеку, то далі йдуть інші дії згідно зробленому алгоритму у підрозділі 2.1, якщо ні – знаходиться в режимі моніторингу;

4) ***передача сповіщень:*** вивід інформації для користувача на Serial монітор;

5) ***продовження стеження:*** система далі стежить за обстановкою. Якщо загроза минає, вона сама повертається в звичайний режим.

Захист даних у системі оповіщення населення

Система оповіщення населення відправляє дані через GSM-мережі, ці дані мають бути захищені. У Законі України «Про мобільний зв’язок та телекомунікації" (№1280-IV) написано, як зробити мережі безпечними » [7].

До основних методів захисту належать:

1) шифрування повідомлень (AES-128 або AES-256), це показано на рис. 2.4;

2) перевірка пристроїв у мережі;

3) захист від DoS-атак.

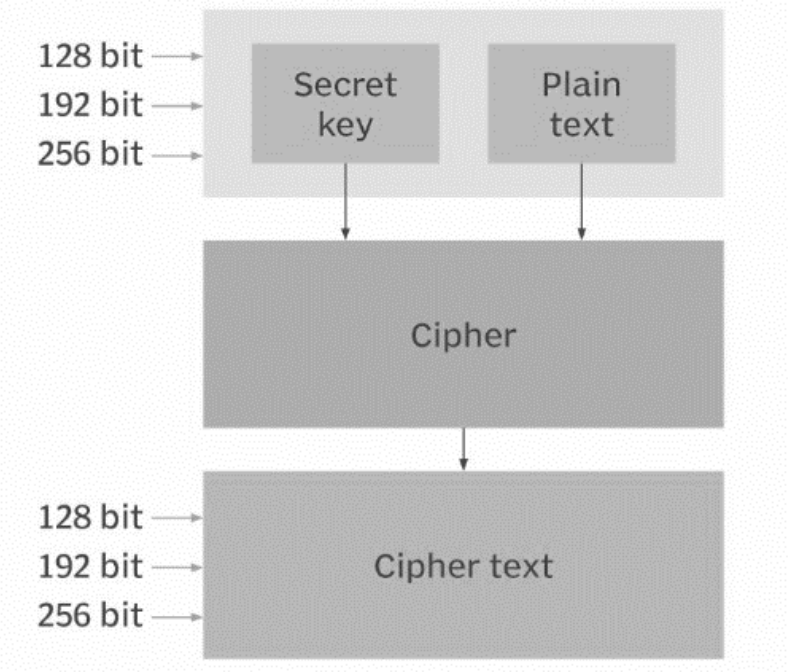

Рисунок 2.4 – Принцип роботи AES [20]

На рис. 2.4 показано, як працює шифрування AES:

На вхід алгоритму подається:

1) секретний ключ (128, 192 або 256 біт);
2) звичайний текст (Plain text) – це дані, які хочемо зашифрувати.

Як відбувається шифрування (Cipher)

Звичайний текст обробляють у кілька етапів: його замінюють (SubBytes), перемішують (ShiftRows), змішують (MixColumns) і додають ключ (AddRoundKey). На виході утворюється: зашифрований текст такої ж довжини, як і початкові дані, 128, 192 або 256 біт.

У підрозділі 2.5 розібрано, як на практиці реалізувати алгоритм для системи оповіщення. Для цього технічну частину і програму поєднують так, щоб усе працювало стабільно. Основні елементи системи – датчики, контролер, GSM-модуль і живлення – разом стежать за ситуацією, перевіряють загрози і швидко реагують.

Виявилося, що вибір техніки залежить від того, наскільки складна система. Для простих підійдуть ESP8266 чи Arduino, а для складніших краще брати

Raspberry Pi або STM32. Важливу роль грає GSM-модуль – він надсилає повідомлення власнику чи охороні через мобільну мережу. Азарова О. В. у своїй статті 2020 року пише, що дані в таких системах можуть перехопити, тому їх треба захищати шифруванням, наприклад, AES-128 чи AES-256, і перевіряти, чи пристрій надійний [2].

Ще важливо, щоб система не включала тривоги без причини. Для цього беруть дані одразу з кількох датчиків, потім додають нові алгоритми для обробки сигналів, іноді навіть із штучним інтелектом.

Отже, для нормальної роботи треба добре з’єднати техніку з програмами, щоб усе було безпечно й надійно. Шифрування, економія заряду і чітка перевірка загроз допомагають системі працювати, навіть якщо погані умови.

2.5 Оптимізація алгоритмів прийняття рішень у системах сповіщення населення

При розробці системи сповіщення населення, треба, щоб алгоритм не просто функціюнував, а працював швидко і без помилок. Щоб він не затягував процес обробки, не витрачав забагато енергії і не вмикав сирену, коли загрози немає. У цьому підрозділі розібрано, як зробити алгоритми кращими, щоб процес попередження був швидким, точним і зручним.

Прискорення роботи системи

Для того щоб системи сповіщення ефективно працювали треба, щоб обробка всіх сигналів від датчиків відбувалася без затримок. Коли буде знайдена небезпека, система має дуже швидко зробити аналіз сигналу , прийняти рішення та сповістити користувача. Якщо ж алгоритм працює повільно, реакція може бути невчасною, що знижує ефективність сигналізації.

Для прискорення роботи системи є кілька варіантів, які допомагають обробляти інформацію від датчиків без затримок. Наприклад, можна робити не багато обчислень, а брати готові таблички з даними, а не перевіряти все по декілька разів, як пишуть у [22], це дозволяє витрачати меньше часу. Також варто

замінювати повільні алгоритми на швидші: якщо в коді є алгоритм зі складністю $O(n^2)$, то треба замінити на O (n log n), якщо є можливість. Ще треба перевіряти все одночасно, якщо датчиків багато, то не по черзі, а разом – так буде ефективніше.

Як зробити, щоб не було помилкових тривог

Система іноді вмикається без причини, наприклад, коли датчик руху реагує на якийсь об’єкт чи тінь, приймаючи це за тривогу. Якщо таке відбувається часто, користувач просто перестає звертати увагу на сповіщення. Щоб таких ситуації не було, є кілька варіантів. Треба налаштувати датчики так, щоб вони самі визначали, що підозріло, а що ні, і не реагували на все підряд, як пишуть у [22]. Також допомагає перевірка кількома датчиками одразу: якщо спрацював лише один, це ще не привід для тривоги, але якщо два чи три датчики фіксують проблему, тоді вже точно трапилась небезпека, як зазначає Азарова О. В. [2]. Ще один спосіб – додавати затримку перед увімкненням сирени, щоб система могла переконатися, що це не випадковість.

Економія енергії

Системи сигналізації часто працюють на пристроях з малою потужністю, які не можуть витрачати багато енергії, тому алгоритми мають бути легкими, щоб пристрій не розряджався швидко. Для цього є кілька способів. Можна використовувати зручні способи зберігання даних, а не прості списки. Також допомагає перевірка датчиків не весь час, а лише за потреби: якщо нічого не відбувається, перевірки робляться рідше, а якщо з’являється щось підозріле – частіше. Ще один варіант – стискати дані перед відправкою, щоб не перевантажувати зв’язок, як радить Верещака Д. С. [5]. МК перевіряє датчики кожні 100 мс, але якщо 5 хвилин усе спокійно, перевірки робляться раз на 500 мс, що економить енергію.

Алгоритми з самонавчанням

Деякі алгоритми здатні підлаштовуватися до нових умов, що покращує роботу оповіщення. Нейронні мережі, як зазначає Андрусенко В. П. [3], можуть навчитися визначати, коли є реальна небезпека, і вирішувати, чи вмикати сирену. Байєсівські алгоритми аналізують минулу інформацію, щоб оцінити ймовірність загрози. Еволюційні алгоритми поступово покращюють свої налаштування для кращого рішення.

Баланс роботи системи

Створити алгоритм, який би був швидким, точним і було мінімальне споживання енергії, неможливо, тому треба обирати що більш важливе, залежно від завдання. Один із підходів – спочатку запускати швидкий алгоритм для перевірки, а при виявленні підозрілої активності переходити до точнішого аналізу. Другий варіант – налаштувати систему на повільніший режим у спокійний період і прискорювати її роботу, коли з’являються небезпеки.

Покращувати алгоритми для систем сповіщення населення дуже важливо, бо від цього залежить, як швидко система зреагує, чи не буде помилок. Під час дослідження було виявлено, як зробити виявлення інформації швидше, уникнути неправильних тривог, економити енергію і додати розумні алгоритми, які самі вчаться, за допомогую аналізу подій які вже відбувалися раніше. Дуже чудово, що нейронні мережі допомагають системі краще працювати, як пише Андрусенко В. П. [3]. А ще в статті [20] пишуть, що такі алгоритми можуть самі підлаштовуватися під нові ситуації.

Висновки до розділу 2

У розділі 2 розглянуто основні математичні методи та алгоритми, які використовуються для створення систем оповіщення населення. Проведено аналіз сучасних способів покращення обробки сигналів, підвищення точності знаходження небезпеки та зменшення помилкових тривог.

Аналіз показав, що ефективність роботи систем сигналізації залежить від того, наскільки правильно підібрані алгоритми для обробки даних. Наприклад, адаптивні методи порогового аналізу, описані у джерелах, допомагають зменшити кількість помилкових тривог. А використання нейронних мереж і методів машинного навчання, дозволяє системі краще розпізнавати справжні загрози.

Окремо розглянуто, як економити ресурси, адже багато сучасних систем працюють на пристроях із невеликою потужністю. У дослідженнях показано, що використання зручних структур даних і алгоритмів допомагає зменшити затримки в обробці подій, що дуже важливо для систем, які працюють у реальному часі.

GSM-технології залишаються важливими для систем безпеки. Звертається увага на необхідність балансу між швидкістю відправки сигналу і його точністю, особливо для мобільних систем сигналізації.

Також перспективним напрямком є розробка алгоритмів, які можуть працювати в умовах обмеженої інформації, наприклад, коли деякі датчики тимчасово недоступні. Такі алгоритми, здатні адаптуватися до нестачі даних, використовуючи методи машинного навчання для прогнозування загроз. Це дозволить системам сигналізації залишатися ефективними навіть у складних або нестандартних ситуаціях.

Отже, у розділі 2 доведено, що для створення системи оповіщення потрібно використовувати комплексний підхід, який включає:

1) алгоритми для швидкої обробки даних;
2) адаптивні способи виявлення небезпеки;
3) економію ресурсів;
4) сучасні методи машинного навчання.

У майбутньому дослідження можуть бути спрямовані на створення гнучкіших алгоритмів, які самі вчаться і підлаштовуються під різні умови. Це допоможе зробити системи надійнішими і ефективнішими в реальних умовах.

3 РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ GSM-СИГНАЛІЗАЦІЇ

3.1 Архітектура розробленої системи

У цьому підрозділі розібрано, як влаштована система з технічної частини й програми, яка потрібна, щоб помічати надзвичайні ситуації, наприклад, пожежу чи коли хтось чужий потрапив в приміщення без дозволу, і повідомляти про це користувача. Оскільки було обмеження у ресурсах, систему зібрано й перевірено в симуляторі Tinkercad, щоб побачити, як усі частини працюють разом і чи добре справляється це рішення. На рис. 3.1 нижче зображено взаємодію основних компонентів системи.

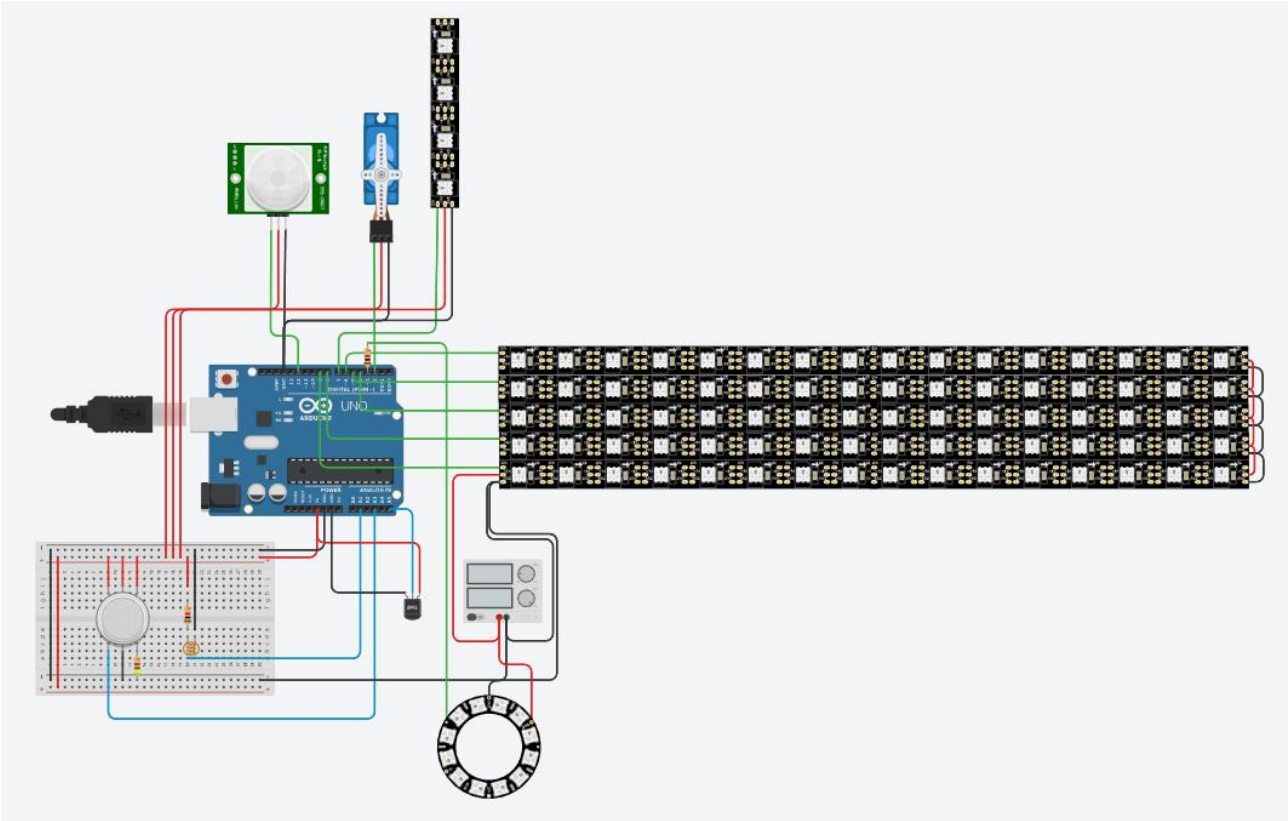

Рисунок 3.1 – взаємодія основних апаратних модулів системи

Система має датчик руху PIR, який бачить, якщо хтось з’явився в зоні. Є датчик MQ-2, що реагує на дим чи небезпечні гази, і сенсор TMP36, який міряє температуру в приміщенні. Ще є фоторезистор, щоб перевіряти, рівень світла. Для показу роботи системи поставлено світлодіодне кільце NeoPixel Ring та

NeoPixel Strip 4 і 8. Мікросервопривід відповідає за деякі механічні дії. Усе це живиться від звичайного блока живлення через макетну плату.

Усі частини системи під’єднані до Arduino Uno R3 за схемою, яку зібрано в Tinkercad. Програма в контролері читає дані з датчиків, обробляє їх і вирішує, що робити, коли стається небезпека чи інша ситуація, все це працює в реальному часі.

Система влаштована так, що якщо датчик руху помічає людину у зоні, одразу вмикається світло, яке показує, куди йти при евакуації. Якщо температура стає занадто високою чи датчик газу знаходить щось небезпечне в повітрі для людей, система переходить у режим тривоги: дисплей показує попередження, NeoPixel починає блимати червоним, а сервопривід може повернутися та відкрити двері, або показати напрямок руху при евакуації як запрограмовано. Якщо фоторезистор зчитує показники і вони нижче норми, система сама вмикає м’яке підсвічування для шляху евакуації. Верещак Д. С. пише, що такі алгоритми допомагають сигналізації чітко реагувати на небезпеку [5].

Схема на рисунку 3.1 показує, як влаштована система, де всі технічні й програмні частини працюють разом, щоб швидко помітити небезпеку і повідомити про неї користувача.

Усе влаштована так, що крім основного завдання – виявляти небезпеку й реагувати – її можна легко розширити. Наприклад, за потреби до системи додаються нові датчики чи пристрої без зміни головного алгоритму. Якщо використати кілька сервоприводів, система може вказувати різні шляхи евакуації залежно від обставин, або можна поставити додаткові екрани для повідомлень у різних частинах приміщення.

Особлива увага приділена тому, щоб сповіщення було помітним. Завдяки NeoPixel-компонентам, які підтримують різні кольори, створено світлові сигнали, що змінюються залежно від ситуації. У звичайному режимі світло зелене й рівне, а при небезпеці воно блимає червоним. Це робить тривогу більш видимою і допомагає людям які мають проблеми слуху.

Аналіз структури розробленої системи показує, що вибрані технічні й програмні частини дозволяють їй виконувати основні завдання системи сповіщення та повідомляти користувача про небезпеку. Під час тестування в Tinkercad усі датчики – руху, температури, диму й освітленості – нормально запрацювали разом із пристроями, такими як NeoPixel-світло, сервоприводи та дисплей. Це допомогло зробити просту, але робочу систему, де всі частини взаємодіють між собою. Верещак Д. С. зазначає, що чітка взаємодія компонентів підвищує надійність системи оповіщення [5].

Особливо приділено увагу над тим, щоб евакуаційний маршрут було добре видно. Світлодіодні елементи NeoPixel міняють кольори й блимають залежно від ситуації, що робить попередження дуже помітними, особливо для людей із вадами слуху. Сервопривод не тільки показує, як усе працює, але й може у майбутньому допомогти, наприклад, автоматично відкривати двері чи вказувати, куди йти при евакуації. Система зроблена так, що до неї легко додати нові частини без переробки основної програми. Це означає, що її можна покращувати надалі. Загалом, розроблене рішення працює злагоджено, реагує на небезпеку одразу й дає користувачу зрозумілу інформацію про те, що відбувається.

3.2 Програмна реалізація системи

Програмну частину системи створено на платформі Arduino, використовуючи мову C++. Програма постійно бере дані з датчиків, аналізує їх і дає команди пристроям, щоб система могла одразу відреагувати, якщо трапилась небезпека. Борісова Л. пише, що такі програми допомагають автоматизованим системам працювати чітко й без збоїв [16].

Мову C++ вибрано для програми на Arduino, бо вона добре працює з цією платформою і дозволяє чітко керувати всіма частинами системи. Завдяки C++ програма швидко обробляє дані з датчиків, що дуже важливо для сигналізації. Також ця мова допомагає писати окремі функції, через що код виходить

зрозумілим і його легко змінити. Репановичі Р. пише, що для таких систем, як ця, C++ добре підходить, бо економить ресурси й забезпечує надійну роботу [25].

Для програми системи використано бібліотеки Adafruit_NeoPixel.h, щоб керувати кольоровими світлодіодами NeoPixel, і Servo.h, щоб управляти сервоприводом, який відкриває чи закриває двері. На початку коду прописано, як під’єднані всі частини: піни для фотодатчика, сенсора температури, датчика диму MQ-2, датчика руху PIR, сервоприводу та п’яти ліній підсвітки. Також задано межі, при яких спрацьовує тривога: температура 55°C і рівень загазованості 500 умовних одиниць.

Готові бібліотеки полегшують процес розробки, бо з ними розробка стає швидше і без помилок. Наприклад, бібліотека Adafruit_NeoPixel дає змогу налаштувати світлодіоди так, щоб вони блимали, як потрібно, без зайвих складнощів у коді. А бібліотека Servo дозволяє упростити керування сервоприводом та указати, що потрібно робити, і він працює як було запрограмновано, без проблем. Кустов В. із колегами пишуть, що такі бібліотеки дуже допомагають, коли створюєш програми для подібних систем [19].

У табл. 3.1 наведено перелік усіх підключених пінів Arduino, їх тип і функціональне призначення в системі.

Таблиця 3.1 – Призначення цифрових і аналогових пінів

Назва	**Позначення в коді**	**Тип піну**	**Призначення**
photoSensor	A1	Аналоговий	Фоторезистор, визначає рівень освітлення
tempPin	A5	Аналоговий	Температурний сенсор
gasPin	A3	Аналоговий	Газовий сенсор (MQ-2)
IKDatchik	12	Цифровий	Інфрачервоний датчик руху (PIR)
doorPin	2	Цифровий	Сервопривід дверей

Назва	Позначення в коді	Тип піну	Призначення
doorPixelsPin	7	Цифровий	LED-підсвітка шляху евакуації
lusterPin	3	Цифровий	Основне освітлення приміщення
pixels[1-5]Pin	4, 5, 6, 8, 9	Цифровий	П’ять LED-ліній для текстових повідомлень

У програмному коді є функції, які розбирають дані з датчиків, керують сервоприводом і визначають, як система реагує в різних випадках. У табл.3.2 коротко описано, що робить кожна функція. Белюченко Д. Ю. із співавторами пишуть, що правильно налаштовані програмні функції забезпечують надійну роботу автоматизованих систем [4].

Таблиця 3.2 – Ключові функції програми

Назва функції	Призначення
getTemperature()	Зчитує температуру навколишнього середовища і повертає її у °C
getPhoto()	Зчитує рівень освітлення з фоторезистора
getGas()	Зчитує рівень диму або газу з MQ-2
getIK()	Перевіряє наявність руху поблизу дверей
openDoor()	Відкриває двері за допомогою сервоприводу
closeDoor()	Закриває двері
turnOnWayFull()	Анімація шляху евакуації + текст «ПОЖАР» + червоне освітлення
textExit()	Виводить повідомлення «EXIT» на LED-матрицях у нормальному режимі
textOFF()	Вимикає усі світлодіоди на текстових матрицях
goodBye()	Анімація виходу: відкриття дверей та підсвічування залежно від освітлення

Назва функції	**Призначення**
choiseScript()	Визначає, який сценарій запускати: пожежа, рух або звичайний режим

Уся робота системи побудована на перевірці умов у головному циклі *loop()*. У табл. 3.3 показано, які умови запускають різні сценарії і що система робить у відповідь. Андрусенко В. П. зазначає, що чітко прописані умови в програмі дозволяють сигналізації правильно реагувати на події [3].

Таблиця 3.3 – Умови активації сценаріїв

Сценарій	**Умова запуску**	**Дія**
Пожежа (1)	Температура > 55°C або газ > 500	Відкриття дверей, світлова індикація «ПОЖЕЖА»
Рух біля дверей (2)	ІК-датчик фіксує рух	Відкриття дверей, підсвічування «EXIT»
Звичайний режим (0)	Все в нормі	Статичне підсвічування «EXIT»

Якщо датчики помічають щось небезпечне, наприклад, високу температуру чи дим, система переходить в аварійний режим (case 1). У ньому двері відкриваються, світлодіоди яскраво блимають, а на дисплеї з’являється напис «ПОЖЕЖА». Це допомагає людям швидко зрозуміти, що робити, навіть якщо погано видно.

Якщо небезпеки немає, але датчик руху бачить, що хтось підійшов до дверей (case 2), система відкриває двері й підсвічує шлях до виходу червоним. Яскравість світла залежить від того, наскільки темно, – це економить енергію, коли світло не потрібне.

У звичайному режимі (default) система просто тримає увімкненим напис «EXIT» на світлодіодах, щоб люди знали, де вихід. Так система навіть без тривоги допомагає орієнтуватися в приміщенні.

Програма системи дозволяє їй швидко реагувати на зміни в приміщенні. Код написаний так, що його легко змінити чи додати щось нове, бо всі функції розділені. Усі сценарії продумані, щоб у разі небезпеки люди могли швидко і безпечно вийти. Програма – це основа роботи системи, яка робить її самостійною і надійною. Андрусенко В. П. із співавторами зазначають, що добре структуроване програмне забезпечення (ПЗ) забезпечує стабільну роботу таких систем [3].

3.3 Тестування в середовищі Tinkercad

Систему протестовано у віртуальному середовищі Tinkercad, де було перевірено, як працює плата Arduino і всі підключені до неї частини. Головне було переконатися, що програма правильно реагує на сигнали від датчиків і що всі сценарії спрацьовують, як задумано. У Tinkercad можна самому виставляти значення для датчиків, наприклад, температуру, газ чи рух, і дивитися, як система на це реагує прямо в процесі. Завдяки цьому вийшло протестувати всі основні функції без реальних пристроїв. Махмудова С. зазначає, що такі перевірки допомагають переконатися в надійності системи сповіщення [23].

Щоб перевірити, як система реагує на потенційну небезпеку, у програмі відтворено різні ситуації, які можуть статися насправді. У табл. 3.4 перелічено основні тести, описано, що саме перевірено, і показано, як система на це реагувала. Азарова О. В. пише, що такі тести потрібні, щоб переконатися в правильній роботі системи [2].

Таблиця 3.4 – Результати тестування програми

№	Тестова ситуація	Початкові умови	Очікувана реакція системи	Результат у Tinkercad
1	Пожежа (висока температура)	tempPin = 55°C	Відкриття дверей, сигнал "ПОЖЕЖА", червоне світло	Виконано успішно
2	Загазованість	gasPin > 500	Відкриття дверей, сигнал "ПОЖЕЖА", червоне світло.	Виконано успішно
3	Рух біля дверей	ІК-датчик активний (HIGH)	Відкриття дверей, напис "EXIT", червона підсвітка	Виконано успішно
4	Темно, але без руху	Освітлення < порогового, ІК-датчик неактивний	LED "EXIT" світиться, двері закриті	Виконано успішно

Усі тести в Tinkercad пройшли успішно. Система без проблем визначала, коли треба вмикати тривогу, а коли залишатися в звичайному режимі, і реагувала, як було заплановано. Це показує, що код написаний правильно і всі частини системи працюють разом без збоїв. Особливо перевіряно, як система перемикається між різними режимами – вона не гальмувала і швидко переходила з одного стану в інший.

Ще перевірено, щоб система не вмикала тривогу коли загроза відсутня. Коли небезпеки не було, вона спокійно чекала і не подавала сигналів. Це показує, що система працює стабільно і не помиляється.

Під час тестів, що показані на рис.3.2 випробувано, як працює сервопривід, що відкриває двері. Коли спрацьовувала тривога, він повертався на потрібний кут, ніби справді відкриваючи двері, як у реальному житті. Гапон Ю. з колегами

пише, що такі механізми важливі для автоматизованих систем [21]. Також система виводила всі дані в послідовний Serial Monitor, де показувала температуру, рівень газу, освітлення і причину тривоги. Це дозволяло записувати, що відбувалося, щоб потім легше було знайти дані, для їх аналізу.

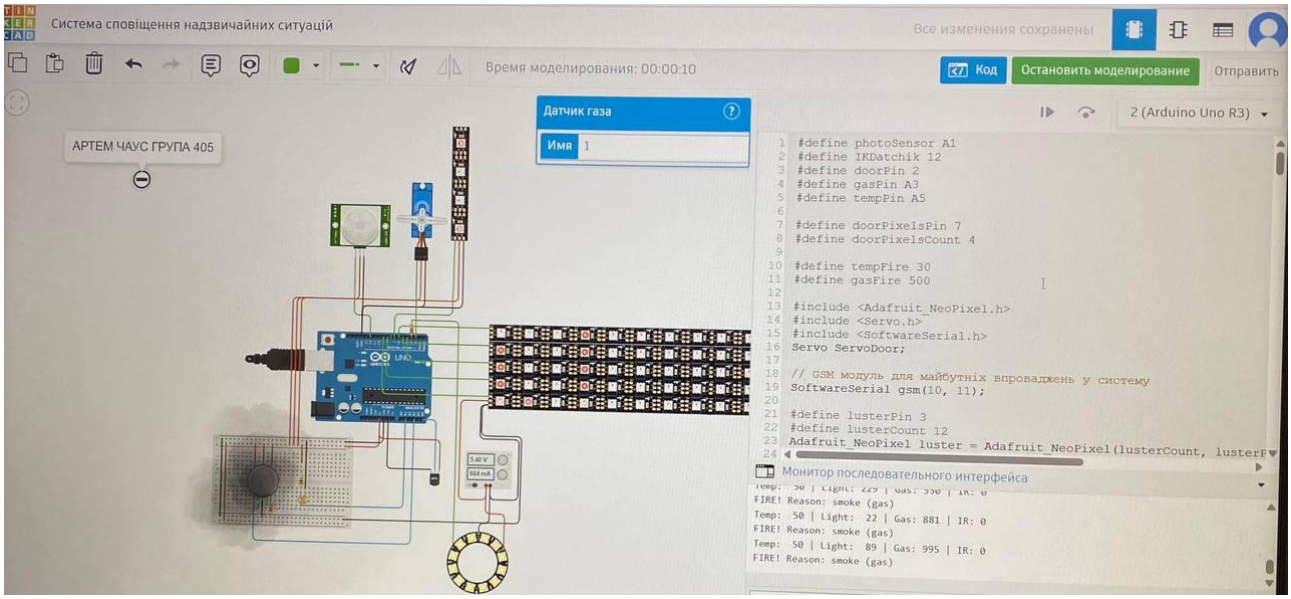

Рисунок 3.2 – Тестування системи сповіщення

У тестах перевірено, як система впорається, якщо одночасно буде висока температура і газ. Система все правильно аналізувала, визначила, яка загроза важливіша, і одразу повідомила про неї. Азаров С. І. пише, що сигналізації повинні правильно реагувати на кілька проблем одразу [1].

Перед тестами для датчиків задано такі значення: температура 50°C, газ більше 500, а освітлення нижче 100. Це допомогло налаштувати систему так, щоб вона правильно працювала в потрібних умовах.

Як показано на рис. 3.3, у випадку загазованості у моніторі відображено тривогу газової небезпеки:

```
FIRE! Reason: smoke (gas)
Temp:  50 | Light:  89 | Gas: 995 | IR: 0
FIRE! Reason: smoke (gas)
Temp:  50 | Light:  45 | Gas: 707 | IR: 0
FIRE! Reason: smoke (gas)
```

Рисунок 3.3 – Газова небезпека

Тестування в Tinkercad було зручним, але не без проблем. По-перше, віртуальна програма не завжди показує, як усе працюватиме в реальності – наприклад, сервопривід чи датчики можуть трохи гальмувати чи давати похибки, а в симуляції цього не видно. По-друге, якщо в схемі забагато елементів, Tinkercad починає підвисати або взагалі не запускається. Також там не можна перевірити, як система реагує на повільні зміни, наприклад, коли температура поступово росте. Ще один мінус – не всі бібліотеки Arduino там працюють, тому доводиться спрощувати код. Але попри це, Tinkercad дуже допомагає для перших перевірок програми і того, як компоненти працюють разом. Верещак Д. С. зазначає, що віртуальні тести корисні, але мають свої обмеження [5].

Перевірка в Tinkercad показала, що система сповіщення працює правильно, навіть без реальних датчиків і пристроїв. Тести підтвердили, що програма нормально реагує на дані від датчиків температури, газу, освітлення й руху, а всі сценарії спрацьовують, як задумано. Особливо добре, що система може одночасно впоратися з кількома проблемами й не має затримки у повідомленні. Послідовний монітор (Serial Monitor) показував усі дані, наприклад, що відбувається і чому, що допомагало стежити за роботою системи прямо під час тестів. Загалом, тести довели, що система готова до використання як нормальна система сповіщення.

3.4 Оптимізація системи після тестування

Після тестів у Tinkercad стало ясно, що система працює стабільно і реагує на всі сигнали від датчиків, але дещо треба було оптимізувати. Деякі моменти доопрацювано, щоб програма працювала краще, зручніше. Андрусенко В. П. із співавторами пишуть, що такі доопрацювання допомагають системам бути надійнішими [3].

Спочатку в коді система реагувала на точні значення, наприклад, температура 55°C або газ більше 500. Але під час тестів виявлено, що якщо значення знаходяться біля цих меж, система може зайвий раз увімкнути тривогу.

Щоб це виправити, додано буфер, який не дає системі часто перемикатися через дрібні зміни. Завдяки цьому система стала працювати спокійніше і надійніше.

Ще один момент – код програми зробили простішим і акуратнішим. Раніше в ньому було багато однакових частин, а деякі частини виглядали занадто заплутаними. Код для виведення даних у Serial Monitor, керування світлодіодами і сервоприводом винесли в окремі функції. Після цього код стало легше читати і правити, якщо треба. Борісова Л. пише, що чітко організований код допомагає уникати помилок і робить систему ефективнішою [16].

Один із прикладів доопрацювання – функція *goodBye()*, яка відкриває двері, коли датчик помічає рух. Раніше цей код був прямо в головному циклі loop(), і через це програму було важко розібрати. Тепер усе зібрано в окрему функцію, що зробило код зрозумілішим. На рисунку 3.4 наведено фрагмент цієї функції у її новому вигляді. Кустов В. пише, що такі зміни полегшують роботу з програмами [19].

```
void goodBye() {
  while (getIK()) {
    openDoor();
    int photo = getPhoto();
    wayPixels.setBrightness(photo);
    for (int i = 0; i < sizeLines; i++) {
      lines[i].setBrightness(photo);
      lines[i].show();
    }
    for (int i = 0; i < doorPixelsCount; i++) {
      wayPixels.setPixelColor(i, wayPixels.Color(255, 0, 0));
      wayPixels.show();
    }
    delay(100);
  }
  wayPixels.setBrightness(255);
  for (int i = 0; i < doorPixelsCount; i++) {
    wayPixels.setPixelColor(i, wayPixels.Color(0, 0, 0));
  }
  wayPixels.show();
  closeDoor();
}
```

Рисунок 3.4 – Функція *goodbye()*

Ще одне доопрацювання – функція *turnOnWayFull()*, яка вмикається, коли датчики фіксують дим чи перегрів. Раніше код для аварійного світла і тексту FIRE був написаний хаотично, через що його було важко зрозуміти чи змінити. Після доопрацювання весь код для світлодіодів і тексту перенесли в окрему функцію, де спочатку вмикається підсвітка, а потім з’являється попередження. Це зробило програму простішою і зручнішою для подальших змін.

На рисунку 3.5 показано частину функції *turnOnWayFull()*, яка вмикає аварійну сигналізацію, якщо стається небезпека.

```
void turnOnWayFull() {
  for (int i = 0; i < lusterCount; i++) {
    luster.setPixelColor(i, luster.Color(233, 250, 47));
  }
  luster.show();

  while (getGas() > gasFire || getTemperature() > tempFire) {
    int turn = -1;
    for (int i = 0; i < 27 - pixelsCount; i++) {
      for (int l = 0; l < sizeLines; l++) {
        for (int j = 0; j < pixelsCount; j++) {
          lines[l].setPixelColor(j, lines[l].Color(255 * TEXT_FIRE[l][i + j], 0, 0));
        }
        turn = (turn + 1) % doorPixelsCount;
        wayPixels.setPixelColor(turn, wayPixels.Color(255, 0, 0));
        lines[l].show();
        wayPixels.show();
        delay(100);
        wayPixels.setPixelColor(turn, wayPixels.Color(0, 0, 0));
        wayPixels.show();
      }
    }
  }

  closeDoor();
}
```

Рисунок 3.5 – Функція *turnOnWayFull()*

Коли систему було виправлено, вона почала працювати набагато краще, навіть якщо ситуація складна. Наприклад, коли перевірено уявну пожежу чи витік газу, усе йшло як заплановано: спочатку загорялося аварійне світло, на екрані писалось попередження, а сервопривід тримав двері відкритими. Через те, що код розкладено по окремих частинах, стало простіше щось змінити – наприклад, швидко змінити рівень тривоги чи додати нові команди, не змінивши всю програму. Це зручно, якщо у майбутньому систему вдосконалити.

Азарова О. В. пише, що добре продумана структура робить системи оповіщення надійнішими [2].

Після тестів і доопрацювань система стала працювати краще і надійніше, її можна використовувати в реальних умовах. Прибрано проблему, коли система реагувала на дрібні зміни показників, приведено код до ладу і розділено його на окремі функції. Це зробило програму зручнішою для роботи й легшою для змін у майбутньому. Усі правки показали, що добре тестувати і структурно писати код на фінальному етапі дуже важливо.

3.5 Можливості покращення системи у майбутньому

Після тестів і доопрацювань у майбутньому систему можна покращити. Сучасні системи безпеки весь час оновлюють, додаючи нові деталі чи програми. У цьому підрозділі розказано про ідеї, як зробити систему оповіщення про катастрофи ще кращою, зручнішою і кориснішою.

Одна з ідей для покращення – поставити GSM-модуль, наприклад, SIM800L який було досліджено та порівняно з аналогами у розділу 1, щоб система могла відправляти SMS на телефон, якщо почалася пожежа чи витік газу. Це зручно, коли власник далеко від будинку, бо повідомлення дійде навіть без Wi-Fi чи інтернету. Але в Tinkercad є обмеження, бо програма не підтримує GSM-модулі, і без реальних пристроїв їх не протестуєш. Баутіста Дж. пише, що віртуальні програми часто не дають змоги перевірити складні апаратні системи [15]. Все ж, якщо робити справжній пристрій на Arduino, GSM-модуль можна легко додати. Після налаштування він би відправляв повідомлення такі як: «Увага! Пожежа!» або «Газ! Перевірте!». Це зробило б систему набагато кориснішою, особливо коли треба реагувати швидко.

На реальному макеті системи у майбутньому можливо буде з’єднати GSM модуль SIM800L з Arduino через бібліотеку SoftwareSerial.h. При спрацюванні системи буде викликатися функція *sendAlertSMS()* яка спеціально реалізована під

майбутню модернізацію і показана на рис. 3.6 та буде надсилати SMS користувачу про небезпеку.

```
void sendAlertSMS(const char* phoneNumber, const char* message) {
  gsm.println("AT");
  delay(1000);

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

  gsm.print("AT+CMGS=\"");
  gsm.print(phoneNumber);
  gsm.println("\"");
  delay(1000);

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

  gsm.write(26);
  delay(5000);
}
```

Рисунок 3.6 – Функція для подальшого модернізування *sendAlertSMS()*

Ще одна корисна функція від додавання GSM-модуля – система може працювати автономно. Якщо через пожежу чи аварію пропаде інтернет або світло, GSM-модуль, наприклад, SIM800L, усе одно відправить SMS, бо живиться від батареї. Таке часто ставлять у «розумних будинках», щоб сповіщення точно дійшло. Плюс до того, SIM800L дозволяє не лише відправляти повідомлення, а й слухати команди від власника. Наприклад, можна надіслати SMS «Вимкни сигналізацію» чи «Відкрий двері», і система це зробить. Чоухан П. пише, що системи, які можуть і приймати, і відправляти сигнали, стають усе популярнішими через їхню зручність і безпеку [17].

У майбутньому можна додати бездротові модулі, наприклад, Wi-Fi чи Bluetooth. Скажімо, із модулем ESP8266 система могла б відправляти дані в інтернет або в додаток на телефоні. Так можна було б дивитися температуру, рівень газу чи вмикати-вимикати систему, не будучи вдома чи в іншому місці де система працює, наприклад на заводі. Можна також зробити сайт для керування. А з модулем HC-05 можна було б керувати системою через Bluetooth прямо зі

смартфона. Белюченко Д. Ю. і Стрілець Д. М. пишуть, що такі речі дозволяють зробити системи гнучкішими й розумнішими [4].

Також важлива ідея – додати резервне живлення. Якщо світло пропаде, система може просто вимкнутися, а це в критичний момент нікуди не годиться. Щоб такого не було, можна під’єднати акумулятор чи PowerBank, який триматиме Arduino, датчики й GSM-модуль у робочому стані. Це дуже потрібно під час аварій. Верещак Д. С. пише, що запасне живлення робить сигналізації набагато надійнішими [5]. З такою батареєю система працюватиме, навіть якщо електрику відключать чи доведеться використовувати її десь у полі.

Щоб розвивати проєкт у майбутньому, було розібрано, що краще: тестувати систему в програмі чи на реальному пристрої. У табл. 3.5 нижче показано, які плюси й мінуси є у віртуального тестування порівняно з фізичним макетом.

Таблиця 3.5 – Порівняння тестування в Tinkercad і на реальному макеті

Критерій	**Tinkercad**	**Реальний макет**
Доступність	Доступний онлайн, не потрібні деталі	Потрібні деталі
Швидкість перевірки	Дуже швидке тестування	Потрібен час на збирання
Реалістичність	Не точна передача, як у реальності	Повна відповідність реальним умовам
Підтримка модулів	Обмежена (немає GSM, Bluetooth тощо)	Підтримка всіх модулів
Надійність результатів	Теоретичні	Теоретичні та практичні
Зворотній зв'язок	Не має звуків, вібрацій та затримок	Реальна поведінка модулей, звуків, руху
Вартість	Безкоштовно	Потрібне фінансування

Надалі систему можна перенести з Tinkercad у реальне життя, зібравши її на макетній платі чи навіть на друкованій платі. Так можна перевірити, як працюють усі модулі разом – GSM, Bluetooth, запасна батарея й усе інше. Це зробить систему готовою до використання вдома, на заводі чи в школі. Реальні тести покажуть у майбутньому, чи є затримки, перешкоди чи проблеми зі зв’язком, які в програмі не побачиш. Кустов В. пише, що перехід від віртуальних тестів до справжнього пристрою допомагає зробити систему надійнішою і краще підготувати її до роботи [19].

Робота над системою сповіщення про техногенні аварії показала, що її можна зробити набагато кращою і кориснішою. Додавання GSM-модуля, SIM800L, дозволить системі слати SMS навіть без інтернету чи світла, що дуже важливо під час аварій. Якщо поставити Wi-Fi чи Bluetooth, можна буде стежити за температурою, газом чи іншими показниками через телефон або комп’ютер, а ще вмикати чи вимикати систему віддалено. Запасна батарея буде корисною, якщо пропаде електро живлення, бо система зможе працювати далі, у критичні моменти. А перенесення з Tinkercad на справжню плату Arduino допоможе перевірити, чи все працює без затримок, і виявити проблеми, яких у програмі не видно, наприклад, слабкий сигнал. Тести в Tinkercad були зручними для старту, але показали, що віртуальна перевірка – це тільки перший крок. Реальний пристрій потрібен, щоб усе протестувати по-справжньому, бо тільки так можна знайти перешкоди чи проблеми з живленням. Ці доопрацювання зроблять систему не просто проєктом, а готовою для використання вдома, на заводах чи в школах.

3.6 Керівництво користувача та інтерфейс системи

Для правильної роботи системи користувачу потрібна чітка інструкція, де розписано, як із нею працювати. Система слідкує за датчиками і, якщо щось змінюється, вмикає світлодіоди, відкриває двері сервоприводом і показує інформацію на серійному моніторі. Щоб краще зрозуміти порядок дій на рис. 3.7

показана блок схема логіки роботи системи. Андрусенко В. П. із співавторами пишуть, що зрозумілі пояснення допомагають людям краще використовувати технічні пристрої [3].

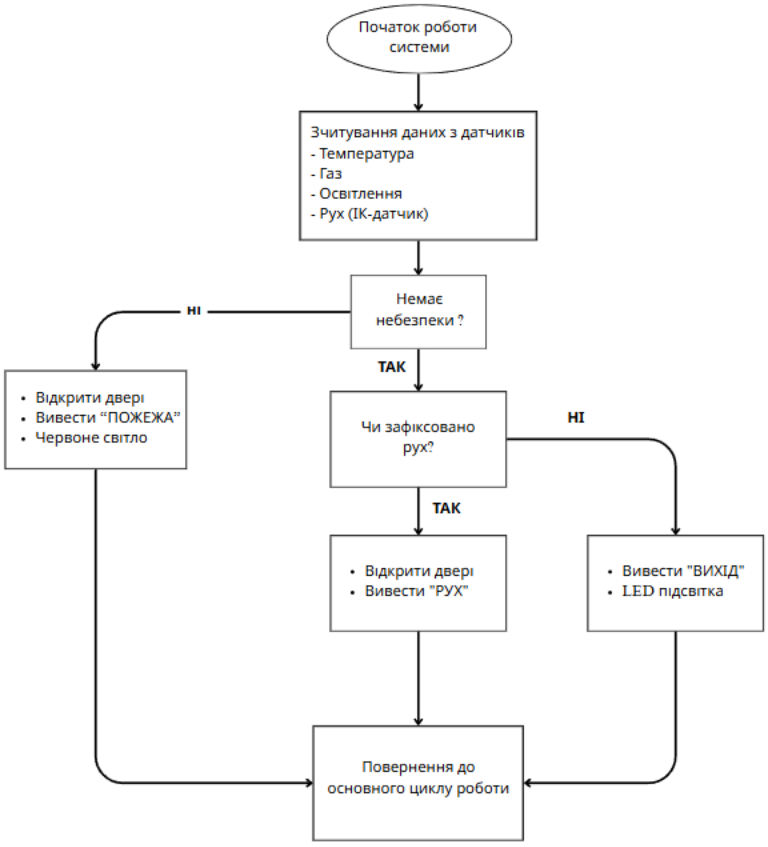

Рисунок 3.7 – Блок-схема логіки по керуванні системою

Блок-схема пояснює, як система працює. Спочатку програма запускає всі деталі: сервопривід, світлодіоди і датчики. Потім вона постійно перевіряє дані з датчиків температури, газу, освітлення і руху. Якщо температура перевищує показники чи є газ, система вмикає аварійний режим: загоряються червоні світлодіоди, двері відкриваються, а на серійний монітор виводиться повідомлення про небезпеку. Якщо ж датчик руху помічає людину біля дверей і немає інших проблем, двері відкриваються для виходу, а на екрані з’являється напис «EXIT». Коли все спокійно, система просто показує на моніторі поточні дані і вмикає зелене або червоне світло залежно від освітлення. Користувачу не треба нічого натискати – усе робиться автоматично, але потрібно стежити за повідомленнями на моніторі. Якщо там пише про загрозу, треба на це одразу

реагувати. А якщо додати GSM-модуль, то буде на телефон надіслано повідомлення. Белюченко Д. Ю. пише, що автоматичні системи полегшують контроль за безпекою [4].

Щоб користувач розумів, що відбувається в системі, усі дані й події показуються на Serial моніторі в Tinkercad. Там видно температуру, рівень газу, освітлення і чи є рух, причому все оновлюється одразу. Якщо стається щось небезпечне, система пише про тривогу, наприклад, про пожежу чи дим, і пояснює, що не так.

Нижче подано приклади повідомлень, які система показує під час роботи:

1) Temp: 28 | Light: 150 | Gas: 300 | IR: 0
   Normal mode – displaying EXIT;
2) Temp: 35 | Light: 140 | Gas: 650 | IR: 0
   FIRE! Reason: smoke (gas);
3) Temp: 55 | Light: 130 | Gas: 400 | IR: 0
   FIRE! Reason: high temperature;
4) Temp: 29 | Light: 180 | Gas: 280 | IR: 1
   Motion detected – opening door.

Користувачу потрібно уважно стежити за повідомленнями на Serial моніторі, бо вони показують, що зараз відбувається з системою. Наприклад, якщо з’являється напис про пожежу чи газ, треба терміново реагувати. Якщо система помітила рух біля дверей, вона їх відкриває, і це видно за відповідним текстом. Коли все нормально, на моніторі пишеться «Normal mode», що означає, що жодних проблем немає.

Для зручності користувача в табл. 3.6 нижче наведено пояснення основних повідомлень, які система виводить на серійний монітор, та дій, які слід виконати.

Таблиця 3.6 – Повідомлення системи та інструкції для користувача

Повідомлення	**Пояснення**	**Дія користувача**
Normal mode – displaying EXIT	Система працює нормально. Немає загроз.	Можна продовжувати перебування
FIRE! Reason: smoke (gas)	Виявлено підвищений рівень газу	Терміново евакуюватися
FIRE! Reason: high temperature	Температура вища за допустиму	Терміново евакуюватися
Motion detected – opening door.	Система побачила рух і відкрила двері	Можна виходити

Інструкція для користувача:

1) ***увімкнення системи:*** після підключення до живлення система сама запускається і починає перевіряти навколишнє середовище.;

2) ***відкриття серійного монітора:*** під час тестування в Tinkercad потрібно увімкнути серійний монітор, щоб бачити дані й повідомлення;

3) ***стеження за повідомленнями:*** якщо на моніторі з’являється текст «FIRE» або, користувачу необхідно терміново покинути приміщення;

4) ***реакція на рух:*** якщо система фіксує рух біля дверей, вона відкриває їх, а на моніторі з’являється відповідне повідомлення;

5) ***нормальний стан:*** коли загроз немає, система виводить поточні показники датчиків, і ніяких дій не потрібно.

Інтерфейс системи зроблено максимально простим, щоб користувач одразу зрозумів, що і як працює. Усі повідомлення автоматично з’являються на серійному моніторі, тож не треба нічого зайвого робити. Завдяки блок-схемі, таблиці повідомлень і прикладам користувач легко розбирається, як працює система. Такий підхід дозволяє використовувати її в надзвичайних ситуаціях навіть без спеціальних знань.

3.7 Аналіз результатів дослідження

Після написання програми систему перевірено у віртуальному середовищі Tinkercad, де можна імітувати роботу Arduino і основних датчиків. Хоча Tinkercad не працює з деякими модулями, наприклад, GSM чи Wi-Fi, він дозволяє протестувати головну логіку системи сповіщення на віртуальному макеті. Верещак Д. С. зазначає, що віртуальні тести корисні для перевірки базової роботи пристроїв [5].

Систему протестовано за різними ситуаціями: звичайна робота, виявлення руху, занадто висока температура, підвищений рівень газу та кілька тривог одночасно. Під час тестів записувано, як система реагує на зміни даних, які повідомлення з’являються на серійному моніторі та які пристрої вмикаються. Чоухан П. пише, що тестування за різними сценаріями підвищує надійність сигналізації [17].

Для перевірки системи вибрано функціональне тестування, тобто перевірено, чи працює вона так, як має за задуманим алгоритмом. У Tinkercad вручну змінювано показники температури, газу, освітлення та руху, щоб перевірити, як система реагує. Борісова Л. пише, що функціональні тести допомагають переконатися в правильній роботі пристроїв [16].

Для наочного показу результатів тестування в табл. 3.7 зібрано приклади ситуацій, які перевірено в Tinkercad, із зазначенням, які дані надано і як реагувала система.

Таблиця 3.7 – Результати тестування системи в Tinkercad

№	Температура, °C	Рівень газу, ум. од.	Освітлення, ум. од.	Рух, ІК	Реакція системи	Повідомлення в Serial Monitor
1	25	300	150	0	Нормальний режим	Normal mode – displaying EXIT
2	36	540	140	0	Тривога через газ	FIRE! Reason: smoke (gas)

№	Температура, °C	Рівень газу, ум. од.	Освітлення, ум. од.	Рух, ІК	Реакція системи	Повідомлення в Serial Monitor
3	55	400	130	0	Тривога через високу температуру	FIRE! Reason: high temperature
4	28	270	180	1	Виявлено рух, відкрито двері	Motion detected – opening door.
5	50	610	160	1	Комбінована тривога (газ + рух)	FIRE! Reason: smoke (gas)

З табл. 3.7 видно, що система правильно поводиться в різних ситуаціях: у звичайному режимі показує «Normal mode», якщо є газ чи висока температура – умикає аварійний режим і пише «FIRE!», а коли помічає рух – відкриває двері. Якщо стається кілька загроз одразу, наприклад, рух і газ, система все одно робить усе, що треба, що підтверджує правильну роботу алгоритму. Тести в Tinkercad показали, що система працює стабільно, без помилок. Усі основні частини програми – перевірка датчиків, прийняття рішень, керування світлодіодами і сервоприводом – спрацьовують так, як задумано. Це означає, що програма надійна і її можна використати в реальному пристрої майже без змін.

Перевірка в Tinkercad показала, що система реагує на все, як і задумано в програмі. Протестувано різні випадки: коли все спокійно, коли може бути пожежа, коли хтось рухається, і коли кілька проблем одразу. Усе працювало як треба, а повідомлення на моніторі були прості й зрозумілі. Це говорить про те, що програма складена правильно і працює без збоїв. Якщо зробити справжній пристрій, система буде готова до роботи.

Висновки до розділу 3

У третьому розділі створено і перевірено систему оповіщення для надзвичайних ситуацій. Спочатку було продумано, процес реагування на небезпеку, підібрано потрібні деталі, написано програму на C++ для Arduino і протестувано все в Tinkercad. Система навчилася читати дані з датчиків температури, газу, світла і руху, реагувати на небезпеку, відкривати двері і показувати повідомлення на моніторі, усе це запрацювало, як було заплановано раніше.

Також важливо відмітити, що через технічні обмеження Tinkercad не вийшло протестувати у віртуальному макеті системи GSM-модуль, бо програма цього не дозволяє. Але вона зроблена так, що в реальному пристрої, наприклад, із модулем SIM800L, все легко інтегрується в реальний пристрій на базі Arduino. У підсумку, усе, що було заплановано, вдалося зробити: система автоматично реагує на небезпеку, легка у використанні для користувача, її без проблем можна покращити, і вона готова до використання в житті.

ВИСНОВКИ

У дипломній роботі було досліджено і перевірено систему сповіщення про небезпеку, яка працює на Arduino UNO та сумісна з GSM модулем. Система автоматично без втручання людей, помічає небезпечні ситуації, наприклад виток газу, перевищення показників температури, зміна освітленості та рух, після цього одразу реагує, вмикає світлові сигнали, відкриває двері та готує текстові повідомлення для користувача.

Об’єктом дослідження був процес попередження людей про техногенні аварії чи інші небезпеки. Метою було розробити GSM систему оповіщення про небезпечні ситуації на базі Arduino. Поставлену мету вдалося виконати.

Усі завдання які були поставленні на початку роботи були виконанні:

1) розібрано як сучасні системи сповіщення, повідомляють про небезпеку. Досліджено сам модуль GSM SIM800L, та порівняно з його аналогами, такими як SIM900A. Виявилося, що для майбутнього впровадження у систему, підійде краще модуль SIM800L;

2) вивчено готові рішення, як WEA, NHK Alert, Cell Broadcast, eCall і систему ДСНС, щоб зрозуміти, якою має бути сучасна система;

3) спроєктувано систему на Arduino UNO. Вибрано цей контролер, бо він простий у розробці, популярний і працює в Tinkercad;

4) підібрано деталі: датчик газу MQ-2, температури TMP36, фоторезистор для світла, ІЧ-датчик для руху, сервопривід для дверей і світлодіоди Neopixel, для кращої візуалізації тривоги;

5) написано програму на C++ для Arduino, щоб система реагувала за заданими правилами. Додано нічний режим, повідомлення на світлодіодах, автоматичне відкривання дверей і анімацію «FIRE»;

6) налаштувано Serial монітор, щоб показувати температуру, газ, світло і рух у реальному часі;

7) система була повністю протестована у середовищі Tinkercad, що дозволило перевірити логіку роботи, реакцію на загрози та індикацію.

Незважаючи на обмеження Tinkercad (відсутність підтримки GSM), усі інші модулі були перевірені та успішно функціонували;

8) написано інструкцію для користувача і намалювано схему, яка показує, як система реагує на датчики;

9) запропонувано, як покращити систему у майбутньому: додати модуль SIM800L для SMS, Wi-Fi, Bluetooth, мобільний додаток, запасну батарею та зібрати справжній пристрій.

Ще звернуто увагу на те, як розвивається мобільний зв’язок. Мережу 5G поетапно починають використовувати, але GSM і 2G/3G досі актуальні в Україні та інших країнах, тому вибір модуля GSM був правильним. Підсумок: система працює та сповіщає про небезпеку, як було задумано. Вона добре показала себе в тестах, її можна використовувати в реальному житті та впроваджувати нові модифікації, які були запропоновані у підрозділі 3.5. З такими доопрацюваннями вона зможе повноцінно функціонувати і для житлових будинків, і для заводів чи шкіл.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Азаров С. І., Єременко С. А., Сидоренко В. Л., Білошицький М. В., Кузнєцов Є. А. Техногенно-екологічні наслідки Чорнобильської катастрофи : монографія. *Київ :* НМЦ ЦЗ та БЖД, 2019. 324 с. URL: https://nmc.dsns.gov.ua/upload/1/3/9/6/6/line-biblioteka-texnogenno-ekologicni-naslidki-cornobilia.pdf (дата звернення: 03.05.2025).

2. Азарова О. В., Гудзь О. В., Блонський О. В. Аналіз методів захисту GSM та шляхи його підвищення. 2020. С. 1. URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/29043/Азарова%2С%20Гудзь%2С%20Блонський%20(АНАЛІЗ%20МЕТОДІВ%20ЗАХИСТУ%20GSM%20МЕРЕЖ%20).pdf – (дата звернення: 24.04.2025).

3. Андрусенко В. П., Коваленко О. Г. Сучасні технології моніторингу довкілля. *Київ:* Видавничий центр «Екологія», 2022. 254 с**.** URL: https://dea.edu.ua/img/source/Biblioteka/додано%202022/Сучасні%20технології%20моніторингу%20довкілля%20на%20прикладі%20Київської%20агломерації.pdf – (дата звернення: 02.05.2025).

4. Белюченко Д. Ю., Стрілець В. М. Багатофакторна оцінка ефективності оперативного розгортання пожежних автомобілів в умовах виникнення надзвичайних ситуацій техногенного характеру. *Вісник Національного університету цивільного захисту України.* 2020. Вип. № 3(156). С. 204-211. DOI: 10.33042/2522-1809-2020-3-156-204-211 – (дата звернення: 02.05.2025).

5. Верещак Д. С., Калашніков О. О. Використання мобільних систем зв'язку для оповіщення. Інформаційні технології: наука, техніка, технологія, наук. вид. : тези доп. 25-ї міжнар. наук.-практ. конф. MicroCAD–2019, 17-19 травня 2019 Харків: НТУ "ХПІ", 2019. С. 318. URL: http://repository.kpi.kharkov.ua/handle/KhPI-Press/44590 – (дата звернення: 24.04.2025).

6. Дивень В. І., Ковалишин В., Хлевной О., Доценко О. Дослідження евакуації людей різних груп мобільності з торговельно-розважального центру. Проблеми надзвичайних ситуацій. *Науковий Вісник: Цивільний захист та*

пожежна безпека. 2022. Вип № 2(14). С. 1 - 152.URL:http://repositsc.nuczu.edu.ua/handle/123456789/21956 – (дата звернення: 16.05.2025).

7. Закон України «Про мобільний зв'язок та телекомунікації в Україні» від 18.11.2003 № 1280 IV. Редакція від 16.12.2021. URL: https://zakon.rada.gov.ua/laws/show/1280-15 – (дата звернення: 24.04.2025).

8. Закора О. В., Фещенко А. Б., Борисова Л. В. Визначення стану електромагнітної сумісності РЕЗ району надзвичайної ситуації. Теорія та практика гасіння пожеж і ліквідації надзвичайних ситуацій : *матеріали XX Міжнар. наук.-практ. конф., м. Харків*, 19 трав. 2022 р. Харків : НУЦЗУ, 2022. С. 129. URL: http://repositsc.nuczu.edu.ua/bitstream/123456789/18528/1/Teoriya_ta_praktyka.pdf – (дата звернення: 16.05.2025).

9. І. В. Мельник. Інформаціні технології проєктування. *Збірник тез доповідей студентської наукової конференції* Черкаси: Черкаський державний технологічний університет, 15 - 17 квітня ЧДТУ. 2019. С. 92 – 99 URL: https://er.chdtu.edu.ua/bitstream/ChSTU/1055/1/ДСН-2019.pdf – (дата звернення: 04.05.2025).

10. Коваленко Р. І. Математичний опис процесу виникнення пожеж під час воєнного стану. Теорія та практика гасіння пожеж і ліквідації надзвичайних ситуацій : *матеріали XX Міжнар. наук.-практ. конф., м. Харків*, 19 трав. 2022 р. Харків: НУЦЗУ, 2022. С. 101. URL: http://repositsc.nuczu.edu.ua/bitstream/123456789/18528/1/Teoriya_ta_praktyka.pdf – (дата звернення: 16.05.2025).

11. Кустов М. В., Федоряка О. І., Сошинський О. І., Савченко О. В. Геоінформаційна система управління пожежними підрозділами. Харків: НУЦЗУ, 2021. С.122 - 133. DOI: https://doi.org/10.52363/2524-0226-2021-34-9 – (дата звернення: 16.05.2025).

12. Неклонський І. М., Рагімов С. Ю., Новожилова М. В. Аналіз оперативних дій рятувальних формувань за допомогою методу мережевого планування. Проблеми надзвичайних ситуацій. 2021. Вип. № 2(34). С. 1-14.

URL: http://repositsc.nuczu.edu.ua/handle/123456789/14734 – (дата звернення: 16.05.2025).

13. Фещенко А. Б., Закора О. В., Борисова Л. В. Розробка імовірнісної моделі типового фрагмента відомчої цифрової телекомунікаційної мережі ДСНС. Харків: НУЦЗУ, 2021. Вип. № 1(33), С. 222 - 233. DOI: https://doi.org/10.52363/2524-0226-2021-33-17 – (дата звернення: 26.04.2025).

14. Amat, R., Mallick, S., & Suna, P.Accident Detection Using GSM and GPS Modules. *International Journal of Recent Technology and Engineering*. 2022. P. 4, URL: https://www.ijrte.org/wp- content/uploads/papers/v11i6/F75060311623.pdf – (Last Accessed: 29.04.2025).

15. Bautista, J. M., Tapic, L., Base, G., & Cabrera, E.Real-Time Vehicle Accident Alert System Based on Arduino with SMS Notification. *Southeast Asian Journal of Science and Technology*, 2019. P. 98. URL: https://sajst.org/online/index.php/sajst/article/view/101 – (Last Accessed: 27.04.2025).

16. Borysova, L. V., Zakora, O. V., Feshchenko, A. B., Rozrobka im-ovirnisnoyi modeli elementarnoho frahmenta vidomchoyi informatsiyno-telekomunikatsiynoyi merezhi, Problems of Emergency Situations. 1(31), 2020. P. 34–43. DOI: 10.5281/zenodo.3901945 – (Last Accessed: 29.04.2025).

17. Chouhan, P. Emergency Alerting System with Location over GSM for Women. *International Research Journal of Modernization in Engineering Technology and Science*. 2020. P. 3 URL: https://www.irjmets.com/uploadedfiles/paper/volume2/issue_9_september_2020/3605/1628083144.pdf – (Last Accessed: 24.04.2025).

18. Gowthami S. Emergency Alert System with GPS and GSM Integration for Real-Time Vehicle Accidents, *International Journal of Progressive Research in Engineering Management and Science.* 2024. P. 11 URL: https://www.ijprems.com/paperdetail.php?paperId=9098ba15497b217cb9be1ea5a0cef4ff&title=EMERGENCY+ALERT+SYSTEM+WITH+GPS+AND+GSM+INTEGRATION+FOR+REAL-TIME+VEHICLE+ACCIDENTS&authpr=Gowthami.+S. – (Last Accessed: 04.05.2025).

19. Kustov, M. V., Sobol, O. M., Fedoryaka, O. I. Territorial location of fire departments of different functional capacity. Problems of emergencies, 2021. Vol 1(33), P. 181-192. DOI: 10.52363/2524-0226-2021-33-14 – (Last Accessed: 24.04.2025).

20. The GSMA's Ministerial Programme at MWC, *(Spain, Barcelona).* 2022. P. 1-43. URL: https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/wp-content/uploads/2022/02/280222-The-Mobile-Economy-2022.pdf – (Last Accessed: 24.04.2025).

21. Hapon Yu., Kustov M., Chyrkina M., Romanova O. Multistage Corrosion of Fuel Element Materials in Nuclear Reactors. Solid State Phenomena. 2022. Vol.334. P.63-69. DOI: https://doi.org/10.4028/p-0s9zyu – (Last Accessed: 24.04.2025).

22. Khartad S., Thorat P., Gunjal S. Emergency Alert & Tracking Device for Women Safety, *International Journal of Research Publication and Reviews*. 2024. T. 5, № 5. P. 3 URL: https://ijrpr.com/uploads/V5ISSUE5/IJRPR28889.pdf – (Last Accessed: 29.04.2025).

23. Mahmudova S., Application of Ant Algorithm for Software Optimization. *Review of Information Engineering and Applications*. 2020. 7. P. 6-17. DOI:10.18488/journal.79.2020.71.6.17 – (Last Accessed: 24.04.2025).

24. Omoruyi, U. S., Olasunkanmi, Y. T., Gandonu, T. F., John, O. B., Adebayo, O. O., Nurein, S. A., Oke, B. O. Impact of Global System for Mobile Communication GSM on the Development of Small-Scale Enterprises in Nigeria. Path of Science, 11(1), 2025. P. 1-11 DOI: https://doi.org/10.22178/pos.113-12. – (Last Accessed: 29.04.2025).

25. Repanovici, R. M., Nedelcu, S., Tarbă, L. A. and Busuioceanu, S. Improvement of Emergency Situation Management through an Integrated System Using Mobile Alerts. 2022. P. 14-24. DOI: 10.3390/su142416424 – (Last Accessed: 24.04.2025).

ДОДАТОК А

Код програми

```
#define photoSensor A1
#define IKDatchik 12
#define doorPin 2
#define gasPin A3
#define tempPin A5

#define doorPixelsPin 7
#define doorPixelsCount 4

#define tempFire 30
#define gasFire 500

#include <Adafruit_NeoPixel.h>
#include <Servo.h>
#include <SoftwareSerial.h>
Servo ServoDoor;

// GSM модуль для майбутніх впроваджень у систему
SoftwareSerial gsm(10, 11);

#define lusterPin 3
#define lusterCount 12
Adafruit_NeoPixel luster = Adafruit_NeoPixel(lusterCount, lusterPin, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel wayPixels = Adafruit_NeoPixel(doorPixelsCount, doorPixelsPin, NEO_GRB
+ NEO_KHZ800);

#define pixels1Pin 6
#define pixels2Pin 5
#define pixels3Pin 4
#define pixels4Pin 8
#define pixels5Pin 9
#define pixelsCount 4
#define sizeLines 5

Adafruit_NeoPixel pixels1 = Adafruit_NeoPixel(pixelsCount, pixels1Pin, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel pixels2 = Adafruit_NeoPixel(pixelsCount, pixels2Pin, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel pixels3 = Adafruit_NeoPixel(pixelsCount, pixels3Pin, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel pixels4 = Adafruit_NeoPixel(pixelsCount, pixels4Pin, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel pixels5 = Adafruit_NeoPixel(pixelsCount, pixels5Pin, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel lines[5] = {pixels1, pixels2, pixels3, pixels4, pixels5};

int TEXT_FIRE[5][27] = {
  {0,1,1,1,0,0,1,1,1,0,0,1,0,1,0,1,0,0,0,1,0,0,0,1,1,1,0},
  {0,1,0,1,0,0,1,0,1,0,0,0,1,1,1,0,0,0,1,0,1,0,0,1,0,1,0},
  {0,1,0,1,0,0,1,0,1,0,0,0,0,1,0,0,0,0,1,1,1,0,0,1,1,1,0},
  {0,1,0,1,0,0,1,0,1,0,0,0,1,1,1,0,0,0,1,0,1,0,0,1,0,0,0},
  {0,1,0,1,0,0,1,1,1,0,0,1,0,1,0,1,0,0,1,0,1,0,0,1,0,0,0},
};
```

```
int TEXT_EXIT[5][27] = {
  {1,1,1,0,1,0,0,0,1,0,1,0,1,1,1,0},
  {1,0,0,0,0,1,0,1,0,0,0,0,0,1,0,0},
  {1,1,1,0,0,0,1,0,0,0,1,0,0,1,0,0},
  {1,0,0,0,0,1,0,1,0,0,1,0,0,1,0,0},
  {1,1,1,0,1,0,0,0,1,0,1,0,0,1,0,0},
};

void setup() {
  Serial.begin(115200);
  gsm.begin(9600);
  delay(1000);
  gsm.println("AT");
  delay(1000);
  gsm.println("AT+CMGF=1");
  delay(1000);

  pinMode(IKDatchik, INPUT);
  pinMode(tempPin, INPUT);
  pinMode(gasPin, INPUT);

  pinMode(pixels1Pin, OUTPUT); pinMode(pixels2Pin, OUTPUT);
  pinMode(pixels3Pin, OUTPUT); pinMode(pixels4Pin, OUTPUT);
  pinMode(pixels5Pin, OUTPUT); pinMode(doorPixelsPin, OUTPUT);

  wayPixels.begin();
  for (int i = 0; i < sizeLines; i++) {
    lines[i].begin();
  }

  luster.begin();
  ServoDoor.attach(doorPin);
  closeDoor();
}

void sendAlertSMS(String message) {
  gsm.println("AT+CMGF=1");
  delay(1000);
  gsm.println("AT+CMGS=\"+380XXXXXXXXX\"");
  delay(1000);
  gsm.print(message);
  delay(500);
  gsm.write(26);
  delay(5000);
}

int getTemperature() {
  int reading = analogRead(tempPin);
  float voltage = reading * (5.0 / 1023.0);
  float temperatureC = (voltage – 0.5) * 100.0;
  return (int)temperatureC;
}

int getPhoto() {
  return map(analogRead(photoSensor), 1017, 348, 255, 1);
}

int getIK() {
  return digitalRead(IKDatchik);
}
```

```
int getGas() {
  return analogRead(gasPin);
}

void openDoor() {
  ServoDoor.write(90);
}

void closeDoor() {
  ServoDoor.write(0);
}

void textOFF() {
  for (int i = 0; i < sizeLines; i++) {
    for (int j = 0; j < pixelsCount; j++) {
      lines[i].setPixelColor(j, lines[i].Color(0, 0, 0));
    }
    lines[i].show();
  }
}

void textExit() {
  int photo = getPhoto();
  for (int l = 0; l < sizeLines; l++) {
    lines[l].setBrightness(photo);
    for (int j = 0; j < pixelsCount; j++) {
      lines[l].setPixelColor(j, lines[l].Color(255 * TEXT_EXIT[l][j], 0, 0));
    }
    lines[l].show();
  }
  for (int i = 0; i < doorPixelsCount; i++) {
    wayPixels.setPixelColor(i, wayPixels.Color(0, 0, 0));
  }
  wayPixels.show();
}

void turnOnWayFull() {
  for (int i = 0; i < lusterCount; i++) {
    luster.setPixelColor(i, luster.Color(233, 250, 47));
  }
  luster.show();

  while (getGas() > gasFire || getTemperature() > tempFire) {
    int turn = -1;
    for (int i = 0; i < 27 – pixelsCount; i++) {
      for (int l = 0; l < sizeLines; l++) {
        for (int j = 0; j < pixelsCount; j++) {
          lines[l].setPixelColor(j, lines[l].Color(255 * TEXT_FIRE[l][i + j], 0, 0));
        }
        turn = (turn + 1) % doorPixelsCount;
        wayPixels.setPixelColor(turn, wayPixels.Color(255, 0, 0));
        lines[l].show();
        wayPixels.show();
        delay(100);
        wayPixels.setPixelColor(turn, wayPixels.Color(0, 0, 0));
        wayPixels.show();
      }
    }
  }
```

```
  closeDoor();
}

void goodBye() {
  while (getIK()) {
    openDoor();
    int photo = getPhoto();
    wayPixels.setBrightness(photo);
    for (int i = 0; i < sizeLines; i++) {
      lines[i].setBrightness(photo);
      lines[i].show();
    }
    for (int i = 0; i < doorPixelsCount; i++) {
      wayPixels.setPixelColor(i, wayPixels.Color(255, 0, 0));
      wayPixels.show();
    }
    delay(100);
  }
  wayPixels.setBrightness(255);
  for (int i = 0; i < doorPixelsCount; i++) {
    wayPixels.setPixelColor(i, wayPixels.Color(0, 0, 0));
  }
  wayPixels.show();
  closeDoor();
}

void adjustNightBrightness() {
  int photo = getPhoto();
  if (photo > 220) {
    luster.setBrightness(30);
  } else if (photo < 30) {
    luster.setBrightness(50);
  } else {
    luster.setBrightness(photo);
  }
}

void warningServoPulse() {
  for (int i = 0; i < 3; i++) {
    ServoDoor.write(30);
    delay(100);
    ServoDoor.write(0);
    delay(100);
  }
}

void blinkFireText(int times) {
  for (int t = 0; t < times; t++) {
    textOFF();
    delay(300);
    for (int l = 0; l < sizeLines; l++) {
      for (int j = 0; j < pixelsCount; j++) {
        lines[l].setPixelColor(j, lines[l].Color(255 * TEXT_FIRE[l][j], 0, 0));
      }
      lines[l].show();
    }
    delay(300);
  }
}
```

```
void gasDangerLevel(int gasVal) {
  int level = map(gasVal, 200, 900, 0, doorPixelsCount);
  for (int i = 0; i < doorPixelsCount; i++) {
    if (i < level)
      wayPixels.setPixelColor(i, wayPixels.Color(255, 0, 0));
    else
      wayPixels.setPixelColor(i, wayPixels.Color(0, 0, 0));
  }
  wayPixels.show();
}

void loop() {
  int temp = getTemperature();
  int photo = getPhoto();
  int gas = getGas();
  int IK = getIK();

  Serial.print("Temp: "); Serial.print(temp);
  Serial.print(" | Light: "); Serial.print(photo);
  Serial.print(" | Gas: "); Serial.print(gas);
  Serial.print(" | IR: "); Serial.println(IK);

  adjustNightBrightness();

  for (int i = 0; i < lusterCount; i++) {
    luster.setPixelColor(i, luster.Color(233, 250, 47));
  }
  luster.show();

  gasDangerLevel(gas);

  if (temp > tempFire || gas > gasFire) {
    openDoor();
    textOFF();
    for (int i = 0; i < sizeLines; i++) {
      lines[i].setBrightness(255);
    }
    luster.setBrightness(255);

    if (gas > gasFire) {
      Serial.println("FIRE! Reason: smoke (gas)");
      sendAlertSMS("Увага! Виявлено дим у приміщенні.");
    } else if (temp > tempFire) {
      Serial.println("FIRE! Reason: high temperature");
      sendAlertSMS("Увага! Температура перевищила безпечний рівень.");
    }

    warningServoPulse();
    blinkFireText(5);
    turnOnWayFull();
  } else if (IK == 1) {
    Serial.println("Motion detected opening door.");
    delay(500);
    goodBye();
  } else {
    Serial.println("Normal mode displaying EXIT.");
    textExit();
  }
  delay(400);
}
```