

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

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

_____ І. М. Журавська

«__» _____ 2023 р.

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

**Апаратно-програмний комплекс розпізнавання
пожежі на базі Arduino та сповіщення користувача
через мобільний застосунок**

Спеціальність «Комп'ютерна інженерія»

123 – КМР.1 – 605.21710511

Студент

_____ В. І. Кукуруза
підпис

«__» _____ 2023 р.

Керівник канд. техн. наук, доцент

_____ В. Ю. Савінов
підпис

«__» _____ 2023 р.

Миколаїв – 2023

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	5
1 АНАЛІТИЧНА ЧАСТИНА. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ СИСТЕМИ, ЩО РОЗРОБЛЯЄТЬСЯ. ФОРМУВАННЯ ВИМОГ ДО АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	8
1.1 Інновації у сфері систем пожежної безпеки	9
1.1.1 Відео виявлення диму/полум'я	11
1.1.2 Інтегрована голосова евакуація	12
1.1.3 Технологія бездротової пожежної сигналізації	13
1.1.4 Нові технології звітності станції спостереження	15
1.1.5 Нові технології гучномовців	16
1.2 Сучасна базова система протипожежної безпеки	18
1.2.1 П'ять основних частин системи пожежної сигналізації	18
1.3 Недоліки існуючих методів вирішення поставлених задач.	20
1.4 Вимоги до апаратно-програмного забезпечення	23
Висновки до розділу 1	24
2 МАТЕМАТИЧНІ МЕТОДИ ТА ПРОЄКТУВАННЯ СИСТЕМИ ЩО РОЗРОБЛЯЄТЬСЯ	25
2.1 Принцип роботи датчику вогню	25
2.1.1 Види датчиків полум'я	25
2.1.2 Характеристики датчику полум'я	27
2.1.3 Переваги та недоліки датчиків полум'я	29
2.2 Принцип роботи GSM модулю	30
2.2.1 Переваги та недоліки GSM модулю	34
2.3 Метод виявлення та розпізнавання пожежі	35
Висновки до розділу 2	41
3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	43

3.1 Детальний опис апаратної платформи та її компонентів	43
3.1.1 Arduino Nano Зв'язок	45
3.1.2 Програмування Arduino Nano	46
3.1.3 Модуль інфрачервоного датчику полум'я	46
3.1.4 Модуль GSM SIM800L	48
3.1.5 Живлення	49
3.3 Побудова макетної схеми пристрою	50
3.4 З'єднання модуля датчику полум'я з апаратною платформою	51
3.5 Розробка програмної частини	51
3.5.1 Вибір мови програмування	53
3.5.2 Вибір середовища розробки	54
3.5.3 Бібліотека SoftwareSerial	55
3.5.4 Бібліотека String.h	55
3.5.5 Використанні змінні та функції	56
Висновки до розділу 3	59
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ. АНАЛІЗ РЕЗУЛЬТАТІВ ОБЧИСЛЕНЬ	60
4.1 Створення акаунту у ThingSpeak та налаштування системи	60
4.2 Налаштування мобільного додатку для отримання сповіщення від пристрою.	62
4.3 Перевірка працездатності	67
4.4 Порівняння прототипу з існуючими рішеннями	69
Висновки до розділу 4	72
ВИСНОВКИ	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	76
Додаток А. Код ПРОГРАМИ	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- BIM – Buidling Information Modeling
- DAC – Digital Alarm Communicator Transmitter
- FACP – Fire Alarm Control Panel
- GCP – Google Cloud Platform
- GSM – Global System for Mobile Communications
- SCT – Smarty Connected Things
- SQL – Structured Query Language
- IoT – Internet of Things

ВСТУП

Пожежі завжди відбувалися через недотримання правил пожежної безпеки, просто випадковості та навіть навмисних цілей із за яких і виникали пожежі що завдавали людству лише лиха, великі збитки та найгірший фактор - численні людські втрати. На даний момент через кількість пожеж в країні через нинішні обставини з агресивними обстрілами та проблемами з електроенергією дуже актуальною є тема запобігання та швидкого оперативного виявлення та ліквідації надзвичайних ситуацій заради спасіння людей та домівок від значних пошкоджень та руйнувань.

На даний момент існує достатньо рішень щодо вирішення проблеми пожежної безпеки і вони здебільшого справляються з поставленою задачею але майже всі також володіють недоліками які знижують ефективність або оперативність задачі, не завжди є економічно вигідними тому достатньо багато людей або жк нехтують правилами пожежної безпеки. Тому для створення апаратно-програмного комплексу було обрано саме сферу IoT через високу ефективність виконання задач, економічну вигідність та безпековий фактор, проект буде оснований на базі Arduino з додаванням необхідних компонентів задля реалізації мети та всіх цілей. Це саме та технологія, яка буде використанна для побудови комплексу.

Актуальність магістерської роботи полягає в тому, що розроблений комплекс стане дуже цінним в сфері пожежної безпеки різноманітних приміщень. Це пов'язано з тим, що після виявлення пожежі він швидко зможе повідомити користувача або екстренні служби, і зменшить ризик затримки пожежної та рятувальної служби в такий час коли пожежі трапляються достатньо часто.

Мета магістерської роботи: реалізувати апаратно-програмний комплекс розпізнавання пожежі на базі Arduino та сповіщення користувача/екстренні служби через мобільний застосунок для швидкого

вирішення надзвичайної ситуації.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз рішень, що існують для розпізнавання пожежі та протипожежної безпеки;
- за результатами аналітичного огляду літератури та патентної інформації визначити загальні вимоги до комплексу та її компонентів;
- розробити алгоритм керування комплексом, навести його блок-схему та дати опис;
- розробити програмне забезпечення системи у вигляді коду керування комплексом;
- побудувати прототип комплексу та здійснити його випробування.

Об'єкт магістерської роботи: алгоритми керування при виявленні та розпізнаванні пожеж у сфері IoT.

Предмет магістерської роботи: Методи та засоби виявлення пожежі та найскоріше оповіщення користувача через мобільний додаток.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- розкриття об'єкту і предмету дослідження;
- опис і аналіз структурних і функціональних особливостей об'єкта дослідження;
- огляд і аналіз сучасного стану інформаційних технологій у даній предметній області;
- огляд і аналіз принципів роботи Arduino з IoT системою.
- виконати розробку апаратної частини та макетної схеми.
- виконати розробку програмної частини та блок-схеми.
- створити прототип системи та тестувати його працездатність.

Наукова новизна полягає в реалізації програмно-апаратного комплексу для виявлення пожежі, який відрізняється простотою програмної

частини, мінімального використання модулів, що зменшить затрати на побудову та собівартість комплексу.

Практичне значення отриманих результатів:

Після аналізу та підбиття підсумків щодо проаналізованого матеріалу, можна зрозуміти методи та принципи виявлення пожежі та передачі сигналу для інформування користувача, та саме як їх використати при розробці апаратно-програмного комплексу. Даний комплекс може бути використаний для покращення пожежної безпеки в приватних будинках, офісах, робочих та інших приміщеннях що потребують покращення пожежної безпеки та швидкого реагування якщо станеться пожежа.

Апробація. Кукуруза В. І., Савінов В. Ю. Апаратно-програмний комплекс розпізнавання пожежі на базі Arduino та сповіщення користувача через мобільний застосунок. Інформаційні технології та інженерія : тези доп. Всеукр. наук.-практ. конф. молодих вчених, аспірантів і студентів. Миколаїв, 7–10 лютого 2023 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2023.

Публікації. За результатами кваліфікаційної роботи створено публікацію у збірнику матеріалів Всеукраїнської конференції [1].

1 АНАЛІТИЧНА ЧАСТИНА. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ СИСТЕМИ, ЩО РОЗРОБЛЯЄТЬСЯ. ФОРМУВАННЯ ВИМОГ ДО АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Пожежна безпека є важливою та необхідною на робочому місці, щоб запобігти та захистити від руйнувань, викликаних пожежею. Пожежна безпека знижує ризик травм і пошкодження будівель, які можуть спричинити пожежі. Розробка та впровадження протоколів пожежної безпеки на робочому місці не тільки вимагається законом, але й має вирішальне значення для безпеки кожного, хто може перебувати в будівлі під час пожежі.

Не варто забувати про важливість пожежної безпеки на робочому місці. Через кількість робочих місць, оточених інгредієнтами та матеріалами, які можуть швидко спалахнути вогонь, необхідно обговорити пожежну безпеку.

Дискусії щодо пожежної безпеки сприяють розумінню небезпеки пожежі та трьох необхідних інгредієнтів для розпалювання вогню – тепла, палива та кисню. Після ознайомлення працівників з цими компонентами та небезпеками, небезпеки стають легше помітними на робочому місці та з більшою ймовірністю поводитися з ними, зменшуючи загальний ризик пожежі.

Навчання працівників є важливим для досягнення загальної мети пожежної безпеки на робочому місці, але ще одним важливим заходом є розробка плану запобігання пожежам. Цей план запобігання містить конкретний опис обов'язків кожного працівника щодо виявлення горючих матеріалів, існуючої пожежної небезпеки та обладнання, що виробляє тепло.

Підготовка до пожежної безпеки, освіта та профілактика – це невелика ціна порівняно з альтернативними збитками, які приносить пожежа. Ризик для бізнесу є високим, коли під загрозою знаходяться як працівники, так і клієнти.

Визначення потенційних небезпек і тригерів

Першим кроком у запобіганні пожежі є проведення оцінки ризику для виявлення потенційних небезпек і тригерів, які можуть призвести до надзвичайної ситуації. Це вимога HSE перед тим, як будь-кому дозволено працювати в приміщенні.

Небезпеки можуть включати:

- Несправні електронні пристрої, проводка та кабелі
- Перевантажені або несправні розетки
- Прилади для приготування їжі
- Теплові прилади
- Горючі матеріали та легкозаймисті рідини, розміщені поблизу джерел тепла
- Сигарети, запальнички та відкритий вогонь

1.1 Інновації у сфері систем пожежної безпеки

Лише минулого червня на конференції NFPA було багато розмов про Smart Connected Things (SCoT) у системах протипожежного захисту. Одну сесію проводив Террі Віктор, Johnson Controls, Grinnell Fire Protection Solutions, а також Крістіна Френсіс, Procter & Gamble Co. Обидва дослідники розглядали ці системи як способи, які дозволяють як власникам, так і постачальникам послуг визначати стан системи та працювати деякі функції перевірки та тестування віддалено. Підключені технології були частиною нашого світу протягом багатьох років, і зараз вони активно присутні в індустрії протипожежного захисту [2].

Наприклад, використання інтелектуальних технологій може забезпечити більш точні та ефективні перевірки та тестування систем. Тільки це може врятувати життя. У статті, в якій взяли інтерв'ю у пари ведучих, вони описують вигаданий сценарій пожежі на складі. Якщо склад оснащено інтелектуальною технікою для моніторингу тиску води та швидкості потоку

спринклерної системи будівлі, вони знають, скільки води тече за хвилину. Це означає, що вони повинні мати контроль над вогнем і можуть передавати цю інформацію команді з питань інциденту. Можливість зовнішнього моніторингу потоку води в будівлі з окроплювачем є лише одним із способів, за допомогою яких розумні технології можуть буквально врятувати життя.

Інтелектуальну технологію також можна використовувати для сповіщення власників будівель про несправність спринклерних систем, замерзлі труби тощо.

Розширене виявлення диму для домовласників

Kidde випустив перший в історії детектор диму (Див. Рис 1.1), який відповідає новим стандартам безпеки UL 268 на 2020 рік. Нові стандарти, щоб усі сигналізації та детектори диму відповідали двом важливим критеріям: підвищена чутливість, щоб відповідати двом новим випробуванням пінополіуретану та здатність розрізняти димові аерозолі від джерел вогню та димові аерозолі від джерел приготування їжі. Більше інформації про ці тести можна знайти на веб-сайті стандартів безпеки UL.



Рисунок 1.1 – Детектор диму та його виявлення

Коротше кажучи, детектори диму повинні вміти виявляти різницю в матеріалах залежно від типу диму, який вони виділяють під час горіння. Детектори також повинні розпізнавати різницю між димом від приготування їжі (або «неприємною» пожежею) і справжньою пожежею.

Нова димова сигналізація Kidde оснащена технологією TruSense, яка може розрізнити пожежі, що швидко спалахують/тліють, і деякі звичайні

побутові сигналізації. Ці зміни були внесені в надії, що домовласники не просто знімуть детектори диму, а сприймуть їх більш серйозно. Інші компанії, що займаються димовою сигналізацією/детекторами, розробляють комерційні програми, оскільки цього року стандартні вимоги в розпалі.

1.1.1 Відео виявлення диму/полум'я

Технологія виявлення диму за допомогою відеозображення існує вже деякий час (Див. Рис. 1.2), але виявлення на основі відео використовується для розширення діапазону застосувань [3].

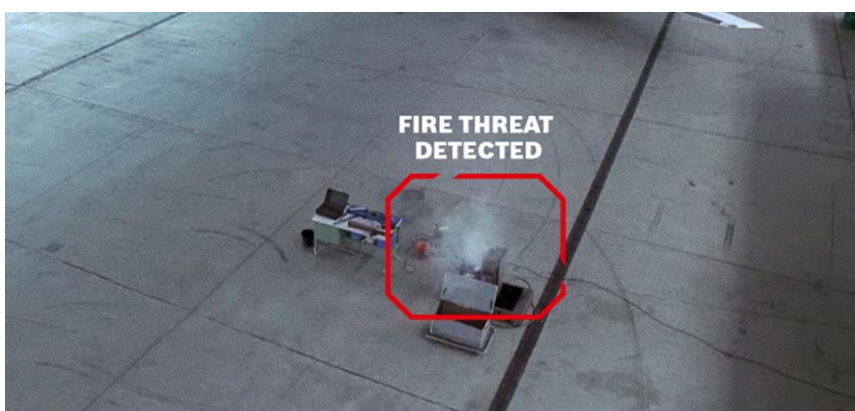


Рисунок 1.2 – Відео детекція диму

Система виявлення пожежі «складається з аналітичних алгоритмів на основі відео, які інтегрують камери в передові системи виявлення полум'я та диму». Це відеозображення потім обробляється програмним забезпеченням, яке визначає, чи можна ідентифікувати дим чи полум'я від пожежі. Алгоритми, що використовуються для виявлення диму/полум'я, можуть базуватися на кількох критеріях, таких як зміни яскравості, контрастності, руху тощо. Залежно від типу системи, ці засоби виявлення можуть навіть забезпечувати виявлення руху та інші функції спостереження на додаток до виявлення/ Місцезнаходження.

Ідея технології VISD полягає в тому, що ви фактично можете побачити ознаки диму та полум'я, окреслені на певній ділянці. Наприклад, на електростанціях, стадіонах, паперових фабриках і у великих складських/розподільних центрах, де пожежі може бути важко локалізувати

традиційними методами. Системи Flame VID відстежують вогонь до місця його виникнення, щоб забезпечити швидше й ефективніше гасіння.

1.1.2 Інтегрована голосова евакуація

Голосова евакуація та можливість налаштовувати директиви евакуації є критично важливими інструментами порятунку. У 2019 році виробники Notifier і Edwards випустили продукти, які виводять інтегровану голосову евакуацію на новий рівень.

Системні сенсорні колонки серії L від Notifier передають повідомлення, необхідні для відповідності коду, з різноманітними опціями, які відповідають вашому застосуванню. Їх 8-дюймові динаміки з подвійною напругою забезпечують високу точність і гучність звуку. Серія також має варіанти кріплення на стіну та стелю. Ці вдосконалені динаміки ідеально підходять для великих складів, густонаселених або надзвичайно гучних приміщень.

Edwards випустив свою платформу EST4 у 2019 році як «преміальну систему зв'язку в екстрених ситуаціях». Таке налаштування системи дає розробникам більше гнучкості для планування реалізації своїх планів на випадок надзвичайних ситуацій, оскільки забезпечує розширену мережеву платформу пожежної сигналізації та аварійного зв'язку.

Основною проблемою для індустрії протипожежного захисту, яка просувається вперед, є стійкість. Системи придушення водяного туману, такі як HI-FOG® охолоджують вогонь та оточення, блокує випромінюване тепло та усуває кисень із осередку вогню. Ці типи систем (Див. Рис. 1.3) найчастіше використовуються в населених пунктах або там, де загроза пошкодження водою може бути згубною.



Рисунок 1.3 – Система придушення пожежі водою

Окрім спринклерних систем, галузь гасить пожежу ще одним способом – це газові системи гасіння. Система гасіння газу була створена як заміна системам галонів кілька десятиліть тому. Тепер системи водяного туману можуть бути заміною придушення газу. Придушення газу є безпечнішим для нашої атмосфери, але було встановлено, що воно сприяє глобальному потеплінню. У результаті системи водяного туману набувають популярності. Ці системи на водній основі є безпечнішими та водночас такими ж ефективними, як інші типи систем.

1.1.3 Технологія бездротової пожежної сигналізації

Сучасний світ повний усіх видів бездротових технологій [4]. Ринок пожежної сигналізації повільно впроваджував бездротову технологію через суворий характер нормативних вимог до обладнання безпеки життя. Однак за останні 10 років завдяки прогресу в бездротовому обладнанні пожежної сигналізації на ринок з'явилися нові продукти, включаючи повний спектр передавачів, ініціальних пристроїв і навіть пристроїв сповіщення. Використання довговічних літій-іонних акумуляторів є одним із ключових факторів, які допоможуть зробити бездротові пристрої та прилади пожежної сигналізації життєздатною комерційною альтернативою дротовим пристроям.

На перший погляд здається, що повністю або переважно бездротова установка пожежної сигналізації дасть власнику значну економію коштів. І з

точки зору вартості встановлення часто це так. Однак характер існуючого архітектурного середовища та сучасні будівельні технології, сертифіковані LEED, можуть створити деякі проблеми для широкого використання технології бездротової пожежної сигналізації (Див. Рис. 1.4). Вогнестійкі бетонні стіни, підлоги та стелі, а також скляні та сталеві конструкції можуть блокувати або послаблювати потужність бездротового сигналу, що призводить до проблем із підтримкою необхідних протоколів зв'язку з обладнанням безпеки життя.

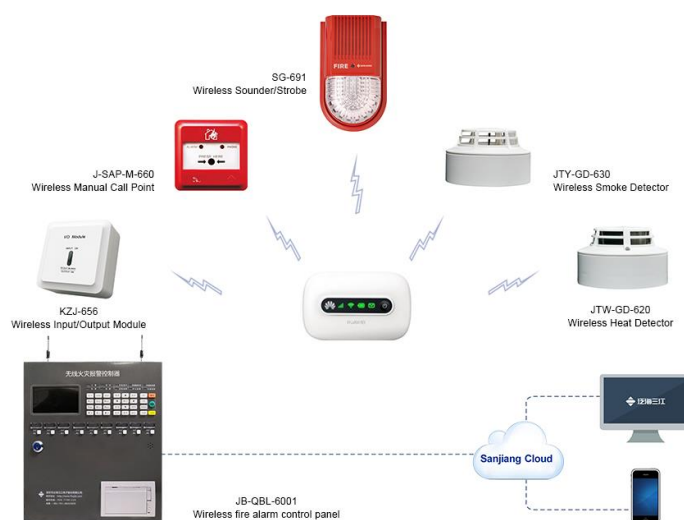


Рисунок 1.4 - Бездротова пожежна сигналізація

Як приклад, вентилі протипожежної спринклерної зони часто встановлюються на сходових клітинах виходу. Як димонепроникні кожухи вони часто виготовляються зі сталі та бетону, тому бездротові сигнали можуть бути ослаблені або навіть перешкоджати постійному зв'язку з бездротовим обладнанням моніторингу пожежної сигналізації, розташованим у них. Незважаючи на те, що такі проблеми можна подолати за допомогою надійного, компетентного дизайну, часто проблеми виявляються лише під час або навіть після введення системи в експлуатацію [4]. На цьому етапі інсталяційні зміни коштують набагато дорожче.

Технологію бездротової пожежної сигналізації слід використовувати хірургічним шляхом у дуже специфічних ситуаціях. Часто це може бути

обмежено історично чи архітектурно чутливими районами або там, де бездротова технологія є єдиним варіантом для конкретної ситуації. У таких випадках переваги бездротової технології можна підвищити за допомогою правильного проектування розташування передавачів і ретрансляторів і перевірки відповідності вимогам виробників щодо встановлення. Надійна програма тестування, перевірки та технічного обслуговування також дуже важлива, оскільки батареї потрібно замінити, а потужність сигналу має підтримуватися протягом усього терміну служби пристрою.

1.1.4 Нові технології звітності станції спостереження

Можливо, ви чули, що звичайна стара телефонна лінія не працює? І ви маєте рацію. Тисячі й тисячі існуючих систем пожежної сигналізації використовують аналогові «стаціонарні» телефонні лінії для зв'язку з віддаленими станціями спостереження або центральними станціями через передавачі цифрової сигналізації (DACT). Хоча така технологія добре працювала протягом багатьох років, вона безперечно змінилася. Протягом кількох останніх циклів NFPA 72, Національного кодексу протипожежної сигналізації та сигналізації, вимоги до станцій спостереження дозволяли та навіть віддавали перевагу використанню бездротового мережевого радіо, стільникового зв'язку та технологій на основі Інтернет-протоколу (IP). Ці технології можуть значно підвищити продуктивність і вартість [5].

Підтримка типових стаціонарних телефонів (зазвичай двох) становить значні витрати на об'єкт щомісяця. Все частіше оператори телефонного зв'язку намагаються ліквідувати стаціонарні телефонні зв'язки, що змушує вас шукати інше рішення. Ці рішення можна надати з невеликими початковими витратами, але в більшості випадків із щомісячною економією протягом усього життя. Перенесення звітності наглядової станції на нову технологію може бути дуже економічно ефективним і вигідним з точки зору продуктивності.

Наприклад, один заклад використовував стандартне з'єднання DAST через дві телефонні лінії для звітування на станцію спостереження. Після спрацьовування тривоги на диспетчері пожежної сигналізації сигнал передавався на станцію спостереження та, зрештою, до пожежної частини для реагування. Цей процес займав у середньому 90 секунд. Однак установа припинила використання своїх стаціонарних телефонів і перейшла на використання бездротової сітчастої радіосистеми. Це скоротило час, необхідний для отримання сигналу тривоги безпосередньо до пожежної частини, приблизно до 8 секунд. Хоча це може бути не типовим, це означає значну перевагу в продуктивності, яку може надати така система. Крім того, установі більше не потрібно щомісяця оплачувати вартість телефонних ліній.

Оновлення технології звітування вашої станції спостереження вимагає належного дизайну. Слід провести дослідження з вашим поточним та іншими постачальниками моніторингу, щоб дізнатися, які технології можуть найкраще працювати у вашій місцевості та яке прийнятне обладнання доступне на станції спостереження. Якщо планується використовувати технології стільникового зв'язку чи радіозв'язку, потрібно узгодити потужність сигналу та розташування антен. І, звичайно, будь-яка зміна в технології повинна бути ретельно перевірена для підтвердження всіх типів передачі та прийому сигналу.

1.1.5 Нові технології гучномовців

Системи голосової пожежної сигналізації активно використовуються вже більше 40 років. Деякі продовжують працювати досить добре, тоді як багато інших були замінені або потребують заміни чи оновлення. Протягом останніх п'яти років або близько того спостерігався приплив голосових систем від різних виробників, щоб задовольнити потребу у все більшій кількості кодових вимог для голосових систем. Однак, окрім переходу від аналогового обладнання до цифрової обробки, значна частина головного

обладнання обробки та підсилювачів не змінювалася протягом багатьох років [6].

Певним чином це стосується технології гучномовців. Однак удосконалення технології або, можливо, краще сказати, розширення технології з комерційного звукового та комунікаційного світу на світ пожежної сигналізації збільшило доступні варіанти розбірливості.

Між гучномовцями пожежної сигналізації майже кожного виробника є деякі спільні риси. Використання 4-дюймового паперового конуса динаміка є економічно ефективним і поміщає динамік у набагато меншу упаковку. Гучномовець зазвичай може бути встановлений на 4-дюймовий або 4 11/16-дюймовий задній ящик, з або без подовжувального кільця для зручності встановлення. Зазвичай доступні дві різні напруги (25 VRMS або 70,7 VRMS), щоб забезпечити певну універсальність із вибраним підсилювачем і довжиною дроту. Частота гучномовця майже завжди становить від 400 до 4000 Гц. Потужність зазвичай забезпечується 4 кранами від 1/4 Вт до 2 Вт, але іноді збільшується від 1/8 Вт до 4 Вт у вісім ступенів потужності. Опції 25 VRMS можуть відрізнятися від опцій 70 VRMS на тому самому гучномовці.

Крім того, усі гучномовці протипожежної сигналізації зі списку UL побудовані з розумінням того, що контрольна напруга контролюватиме цілісність ланцюга ланцюга гучномовця, коли ланцюг не використовується. Зазвичай це робиться за допомогою блокуючого діода та вказує на значну відмінність від комерційних аудіодинаміків, тому вони зазвичай не є взаємозамінними [6].

Інші розробки змінюють статус-кво гучномовців пожежної сигналізації. По-перше, відбувається перехід від звичайних до високоякісних динаміків. Це дуже важливо. Перехід до високоякісного звуку передбачає збільшення частоти динаміка, розширення від 400 до 4000 Гц до 300–8000 Гц, а в деяких випадках навіть від 200 до 10 000 Гц. Майте на увазі, що

велика частина (приблизно більше 30 відсотків) зони людського мовлення знаходиться вище діапазону 4000 Гц. Перехід до високої якості відбувається при продовженні використання 4-дюймового конуса динаміка (таким чином зберігаючи необхідну невелику нерухомість) і зберігаючи всі інші параметри потужності динаміка та напруги.

У цьому відношенні слід бути певною обережністю, оскільки «висока точність» часто використовується як маркетинговий термін і може відображати так звану високу точність лише на певній частоті, наприклад 520 Гц, і може не вказувати на збільшення фактичного повного частотний діапазон гучномовця для людської мови.

1.2 Сучасна базова система протипожежної безпеки

Система комерційної пожежної сигналізації є важливою частиною повної безпеки життя для комерційних і промислових об'єктів [7]. Системи пожежної сигналізації (Див. Рис. 1.5) комерційних будівель можуть бути досить складними, як-от багатоповерхові багатоповерхові будинки або розгалужені кампуси з багатьма будівлями та об'єктами, які необхідно підключити для масового сповіщення. Постачальники комерційних систем пожежної сигналізації повинні йти в ногу з останніми досягненнями в технологіях безпеки життя та гарантувати, що всі компоненти системи пожежної сигналізації працюють разом.



Рисунок 1.5 – Система протипожежної безпеки

1.2.1 П'ять основних частин системи пожежної сигналізації

Комерційна та промислова пожежна сигналізація та системи безпеки життя складаються з п'яти основних частин, які складають повну систему пожежної сигналізації та включають наступне:

- Пульт пожежної сигналізації
- Ініціальні пристрої
- Пристрої сповіщення
- Первинне джерело живлення
- Резервне живлення

Пульт керування пожежною сигналізацією (FACP)

Панель керування пожежною сигналізацією — це мозок усієї операції, системний центр, який контролює цілісність системи, вводить, виводить і передає всю інформацію. Коли спрацьовує пристрій ініціювання, він посилає сигнал на прилад пожежної сигналізації, який потім запускає пристрої сповіщення або тривогу. Звичайні контрольні панелі пожежної сигналізації

можуть містити стандартну розгалужену проводку зі змінними платами, по одній для кожної визначеної зони. Досконаліші системи можуть містити цифрові схеми для передачі даних на всі пристрої, які можна запрограмувати для зон. Усі FACP відобразять стан вашої системи з кодами усунення несправностей, які можуть дозволити вам вручну налаштувати, як-от вимкнути будильник або скинути систему після проблеми. Завжди звертайтеся до свого постачальника системи пожежної сигналізації з будь-якими запитаннями щодо сповіщень про статус.

Пристрої ініціювання активують систему в аварійній ситуації і можуть бути як ручними, так і автоматичними.

Вручну – станція ручної тяги є прикладом ручного ініціального пристрою, який вимагає, щоб хтось потягнув за важіль, натиснув кнопку або розбив скло, щоб вручну ініціювати сигналізацію.

Автоматичний – пристрої автоматичного ініціювання включають пристрої виявлення диму, газу, полум'я та тепла. Пристрої ініціювання можуть бути як адресними, так і неадресними, з адресними пристроями, які сигналізують про точне місце надзвичайної ситуації. Існують різні типи детекторів диму, як-от аспіраційні сповіщувачі або променеві сповіщувачі, які є різними технологіями для виявлення диму, але надсилають сигнал на панель керування, щоб сигналізувати про тривогу.

Ваш постачальник систем пожежної сигналізації може допомогти вам визначити, який тип пристроїв ініціювання найкраще підходить для вашої будівлі та де саме вони повинні бути розташовані для максимальної безпеки та відповідності нормам.

Пристрої сповіщення сповіщають усіх мешканців будівлі про пожежу чи надзвичайну ситуацію. Вони включають в себе стробоскопи, гудки, сирени, дзвінки або комбінацію стробоскопів і гудків. Вони також можуть включати звукові сповіщення, як-от конкретні інструкції щодо безпечної евакуації з будівлі. Системи масового сповіщення можна запрограмувати на

доставку різних повідомлень про різні події або спеціальних звукових сповіщень.

1.3 Недоліки існуючих методів вирішення поставлених задач.

На певному рівні переважна більшість пожеж спричинена людською помилкою, чи то через падіння сірника, чи через несправність обладнання. Тому не варто дивуватися тому, що системи протипожежного захисту все більше автоматизуються, намагаючись усунути цей людський компонент. Теоретично мережеві системи можуть допомогти виявляти та своєчасно гасити пожежі, рятуючи майно та життя [8].

Проте, як і багато нових технологій, існує небезпека того, що підприємства не повністю усвідомлюють пов'язані з цим ризики. Хоча мережеві протипожежні системи пропонують багато переваг, вони також схильні до ризиків, які не вплинули б на традиційну систему. Оскільки ці системи стають все більш поширеними, дуже важливо, щоб ми розглянули необхідність їх захисту та притягували до відповідальності виробників і розробників програмного забезпечення.

Для активного протипожежного захисту зазвичай потрібен якийсь вхідний сигнал, наприклад особа, яка запускає пожежну сигналізацію. Там, де вони є автоматичними, наприклад зі спринклерами чи сигналізаціями, які виявляють дим і вогонь, ці системи, як правило, локалізовані та працюють лише в межах певної близькості. Щоб вимкнути їх, зазвичай потрібно отримати доступ до клапана або панелі керування та внести зміни вручну.

Мережеві протипожежні системи мають низку переваг перед традиційними. Якщо, наприклад, спрацьовує сигнал тривоги, мережева система може повідомити вам, де саме він знаходиться та коли він був активований, надаючи вам більше інформації, яку ви можете використати для вжиття заходів. Він також може дозволити вам з'єднати різні частини будівлі чи ділянки без прокладання кабелів між ними, і може дозволити вам керувати ним дистанційно, щоб вимкнути його або для тестування.

Завдяки можливості бездротового підключення так званих датчиків Інтернету речей (IoT) мережеві системи протипожежного захисту все частіше розглядаються як недороге та високотехнологічне рішення. Завдяки використанню дедалі досконалішого програмного забезпечення управління інформацією про будівлі (BIM) цими системами часто можна керувати через єдиний інтерфейс, а дані використовуються для планування технічного обслуговування та прийняття рішень [9].

Переваги мережевих систем активного протипожежного захисту незаперечні, і цілком ймовірно, що попереду буде ще більше покращень. Неможливо уявити собі час, коли комп'ютери розумно оцінюватимуть ризики, самі передаватимуть інформацію пожежній бригаді, а також плануватимуть і бронюватимуть технічне обслуговування. Однак системи, які ми маємо на даний момент, є технологіями, що розвиваються, і є відкритими для недоліків, головним з яких є програмне забезпечення, яке вони використовують.

Щоб працювати дистанційно та збирати дані, кожен аспект мережевої системи (наприклад, сигналізація або детектори диму) має містити крихітний комп'ютер, на якому працює запатентована форма програмного забезпечення. Оскільки наразі існує кілька визнаних стандартів, це програмне забезпечення часто розробляється компаніями, які виробляють апаратне забезпечення, багато з яких розташовані в Китаї. Це програмне забезпечення часто не оновлюється регулярно, і воно може бути не дуже безпечним.

Для операторів, які не звикли до мережевого протипожежного захисту або не знайомі з конкретним апаратним забезпеченням, це може створити ряд дірок у безпеці вашої мережі. Незважаючи на те, що ваші протипожежні системи та системи BIM можна відокремити від Інтернету, багато з цих датчиків і пристроїв є готовими рішеннями, що підключаються та працюють. Якщо ви встановите їх, не налаштувавши їх певним чином, вони часто

використовуватимуть налаштування за замовчуванням, паролі та мікропрограму, що може легко стати причиною їх зламу хакерам.

Це має численні потенційні наслідки. Отримавши віддалений доступ до окремого IoT-пристрою, наприклад детектора диму, хакери можуть використовувати пристрій як «бекдор» в мережу та отримати доступ до файлів на інших комп'ютерах. Простіше кажучи, вони також можуть втручатися в систему протипожежного захисту або інші аспекти керування будівлею – дистанційно запускаючи сигналізацію або вимикаючи системи, піддаючи вас ризику в надзвичайній ситуації.

На щастя, існує кілька способів вирішити цю проблему. Найбільш очевидним є дотримання традиційних систем активного протипожежного захисту, які не сприйнятливі до цих проблем. Однак ви можете зіткнутися зі зростаючим запитом на оновлення, і ви втратите інформацію, яка допоможе вам ефективніше реагувати на пожежі [9].

Безпека також має займати центральне місце під час купівлі та встановлення нових систем, зосереджуючись на пошуку апаратного та програмного забезпечення від перевірених продавців, які можуть підтвердити свій досвід у сфері кібербезпеки. Багато проблем з Інтернетом речей та іншими мережевими пристроями пішли від придбання дешевих пристроїв, які часто мають погано розроблене програмне забезпечення або неангломовні інтерфейси, що ускладнює керування ними.

Якщо ви турбуєтеся про підводні камені мережевої системи активного протипожежного захисту, одним із варіантів може бути інвестування більших коштів у пасивний протипожежний захист. Хоча життєво важливо, щоб ви доклали всіх зусиль для безпеки та захисту своїх сигналізацій та інших пристроїв, пасивний протипожежний захист може підвищити безпеку та забезпечити додатковий спокій. Такі системи, як протипожежні стіни, бар'єри та двері, можуть допомогти захистити структуру вашої будівлі,

захищаючи критичну інфраструктуру та забезпечуючи цілісність шляхів евакуації.

Працюючи над запровадженням високих стандартів і вибираючи правильних постачальників, ми можемо спільно забезпечити мінімізацію ризику, пов'язаного з якістю апаратного та програмного забезпечення. Окрім цього, окремі підприємства будуть відповідальними за те, щоб їхній персонал був навчений використовувати нове обладнання, підтримувати його в актуальному стані та захищати від ризику онлайн-втручання.

Як результат постає мета покращити та зменшити кількість недоліків IoT систем протипожежної безпеки до мінімуму та покращити виконання мети до максимуму з якомога швидкішим оповіщенням екстрених служб/користувача про початок пожежі та як наслідок її скоріше гасіння та спасіння життів [10].

1.4 Вимоги до апаратно-програмного забезпечення

Після аналізу доступної інформації, було сформовані певні вимоги до майбутнього апаратно-програмного забезпечення:

- Невеликий розмір для зручності у приміщеннях;
- простота використання;
- велика точність виявлення пожежі;
- достатній діаметр виявлення пожежі;
- розмір програмної частини не повинен перевищувати ліміт платформи;
- низька ціна у порівнянні з аналогами.

Висновки до розділу 1

Під час оформлення розділу 1, було опановано та проаналізовано матеріал щодо різних систем, технологій виявлення пожежі та пожежної безпеки.

Було описано датчки які зазвичай використовуються для виявлення пожежі та описано їх принцип роботи на надано характеристики.

Щодо проблем, було описано та проаналізовано недоліки існуючих IoT систем пожежної безпеки та розпізнавання пожежі та запроновано варіанти вирішення наявних проблем різними способами.

Були проаналізовані аналоги, та сформовані вимоги до майбутнього апаратно-програмного забезпечення. Апаратно-програмний комплекс повинен бути: невеликого розміру, простим у використанні, точним при розрахунках для виявлення пожежі, розмір програмної частини не повинен перевищувати ліміт платформи, та дешевшим у порівнянні з аналогами.

2 МАТЕМАТИЧНІ МЕТОДИ ТА ПРОЄКТУВАННЯ СИСТЕМИ ЩО РОЗРОБЛЯЄТЬСЯ

2.1 Принцип роботи датчику вогню

Датчик полум'я - це тип детектора, який використовується для виявлення, а також реагування на пожежу або полум'я. Датчик полум'я часто реагує швидше і точніше в порівнянні з датчиком тепла або диму через механізми, які він використовує для виявлення полум'я. Датчики полум'я зазвичай використовують для перевірки правильності роботи печей.

Датчики полум'я використовують УФ (ультрафіолетове) або ІЧ (інфрачервоне) або УФ-ІК технологію для виявлення полум'я за частки секунди. Ці датчики реагують на виявлене полум'я в залежності від установки, хоча вони включають подачу сигналу тривоги, відключення паливопроводу та активацію системи керування вогнем [11].

УФ-датчик полум'я працює, просто виявляючи УФ-випромінювання. Як правило, всі пожежі генерують УФ-випромінювання в точці займання, тому у разі пожежі датчик буде попереджений і генеруватиме серію імпульсів, які змінюються електронікою датчика та видають сигнал тривоги.

Датчик полум'я з ІЧ-технологією працює, перевіряючи смугу ІЧ-спектру на наявність особливого орнаменту, що випромінює гарячі газу. Але для цього типу пристроїв необхідно мерехтливий рух полум'я. Інфрачервоне випромінювання зазвичай генерується не тільки полум'ям, а й лампами, печами і т.д. Тому високий ризик хибної тривоги.

Датчик УФ-ІК здатний виявляти як УФ-, так і ІЧ-випромінювання, тому він має обидва датчики. Комбінований датчик має найкращі можливості придушення хибних тривог порівняно з окремим УФ/ІЧ-детектором.

2.1.1 Види датчиків полум'я

УФ-датчики полум'я

Датчики з УФ рамкою або ультрафіолетовою рамкою використовуються для виявлення ультрафіолетового випромінювання, тому що більшість пожеж випромінюють ультрафіолетове випромінювання. Ці датчики особливо корисні для служб, які схильні до особливих небезпек, таких як загоряння галогенів, металів і вуглеводнів. Таким чином, ці типи датчиків забезпечують високу чутливість на невеликих відстанях від 0 до 50 футів, але їхня продуктивність буде знижуватися на будь-якій відстані. Ці датчики реагують на електричні розряди, такі як блискавка.

Інфрачервоні датчики полум'я

Інфрачервоні датчики полум'я контролюють ІЧ-випромінювання. Ці датчики також виявляють і аналізують ІЧ-спектральний діапазон, щоб виявити певні попередньо визначені шаблони, випромінювані гарячими газами. Ці прототипи виявляються за допомогою термографічних або термографічних камер. Технологія, яка використовується цими датчиками, — це технологія розпізнавання полум'я, яка виявляє випромінювання ближнього інфрачервоного діапазону за допомогою пристрою із зарядовим зв'язком (CCD). На датчики ІЧ-випромінювання вплив водяної пари значно впливає, оскільки вода просто поглинає більшу частину отриманого інфрачервоного випромінювання. Через цю основну причину ці датчики не можуть забезпечити точні результати на відкритому повітрі.

Ультрафіолетові/інфрачервоні сповіщувачі полум'я

Ультрафіолетові/інфрачервоні сповіщувачі полум'я додають датчики як для УФ-, так і для ІЧ-випромінювання. Ці два датчики просто працюють окремо. Однак деякі включали схеми та проводку для допомоги в процедурі детектора та для оцінки УФ- та ІЧ-сигналів. Таким чином, це допомагає розвіяти будь-які помилкові тривоги, які можуть бути викликані тим чи іншим сигналом.

Ця підвищена стійкість до помилкової тривоги дозволяє використовувати ультрафіолетові або інфрачервоні сповіщувачі полум'я як

усередині, так і зовні. Однак він обмежує виявлення пожежі включенням пожеж, які просто створюють УФ- та ІЧ-випромінювання.

Багатоспектральні інфрачервоні датчики полум'я

Датчики MSIR використовують ІЧ-випромінювання багатьох довжин, щоб відрізнити випромінювання, що викликає полум'я, від джерел випромінювання, які не викликають полум'я. Датчики MSIR дуже швидко реагують на пожежі на відстані до 200 футів як у приміщенні, так і на вулиці. Ці датчики здатні виявляти полум'я навіть у найдимніших пожежах, а також не викликають помилкових тривог через сонячне світло, освітлення чи інші гарячі об'єкти в оточенні [12].

Датчики полум'я IR3

Датчики полум'я IR3 порівнюють картини випромінювання між трьома різними інфрачервоними спектральними діапазонами та співвідношенням діапазонів випромінювання. Як правило, ці типи датчиків полум'я запрограмовані на виявлення однієї смуги випромінювання в діапазоні 4,4 мкм, а решта двох смуг – у діапазоні нижче та вище спектру 4,4 мкм. Отже, цей датчик здатний відокремлювати фактичне полум'я та неполум'яне випромінювання, яке впливає на результати. Тому ці датчики забезпечують чудові результати виявлення полум'я, просто уникаючи фонового випромінювання.

2.1.2 Характеристики датчику полум'я

Діапазон виявлення датчика полум'я залежить від місця його встановлення, доступності детектора та атмосфери, що оточує сповіщувач. Діапазон виявлення полум'я показаний на наступному зображенні (Див. Рис. 2.1).

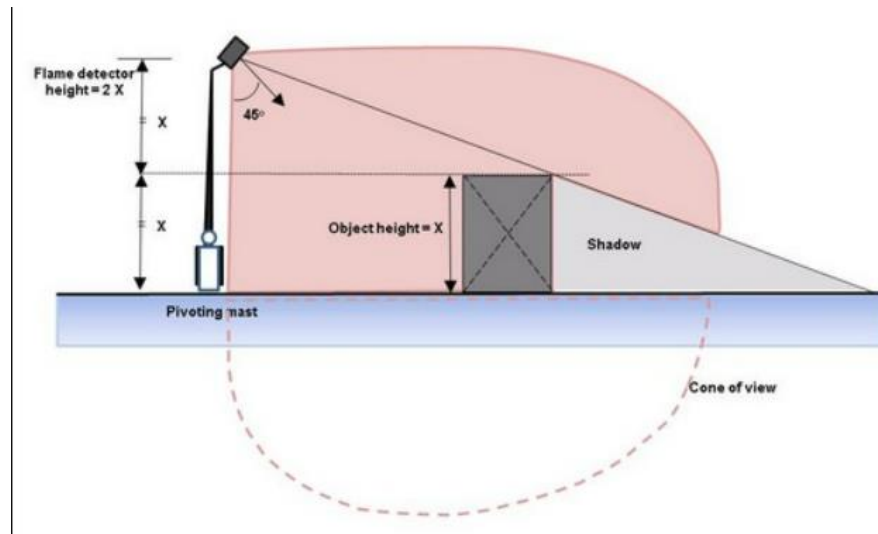


Рис. 2.1 – Діапазон виявлення вогню

Датчик полум'я розміщується на висоті, що вдвічі перевищує об'єкт. Кут нахилу фіксується на рівні 45 градусів для максимального покриття дальності [13].

Триконтактна схема контактів датчика полум'я показана нижче (Див. рис. 2.2).



Рис. 2.2. - Схема контактів датчика

Pin1 (VCC): Це контакт джерела живлення 5 В.

Pin2 (GND): це контакт GND.

Pin3 (OUT): це цифровий вихідний контакт.

Технічні характеристики датчика полум'я включають наступне:

- Діапазон робочої напруги від 3,3 В до 5 В.
- Робочий струм 15 мА.

- Використовується мікросхема компаратора LM393.
- Тип датчика – фототранзистор YG1006.
- Чутливість регулюється за допомогою потенціометра.
- Тип виходу: цифровий о/р або цифровий і аналоговий вихід.
- Червоний світлодіод для живлення, а зелений для виходу.
- Діапазон спектру від 760 нм до 1100 нм
- Кут виявлення від 0 до 60 градусів.
- Діапазон робочих температур від -25 °С до 85 °С
- Розмір друкованої плати 3 см X 1,6 см.

Ця схема модуля датчика може бути розроблена з різними компонентами, такими як LM393, фототранзистор YG1006, два резистори 1К, такі як R1 і R3, резистор R2 -10К, потенціометр VR1 – 10К, два керамічні конденсатори C1, C2 0,1 мкФ і червоні світлодіоди D1, D2. (Див. Рис. 2.3).

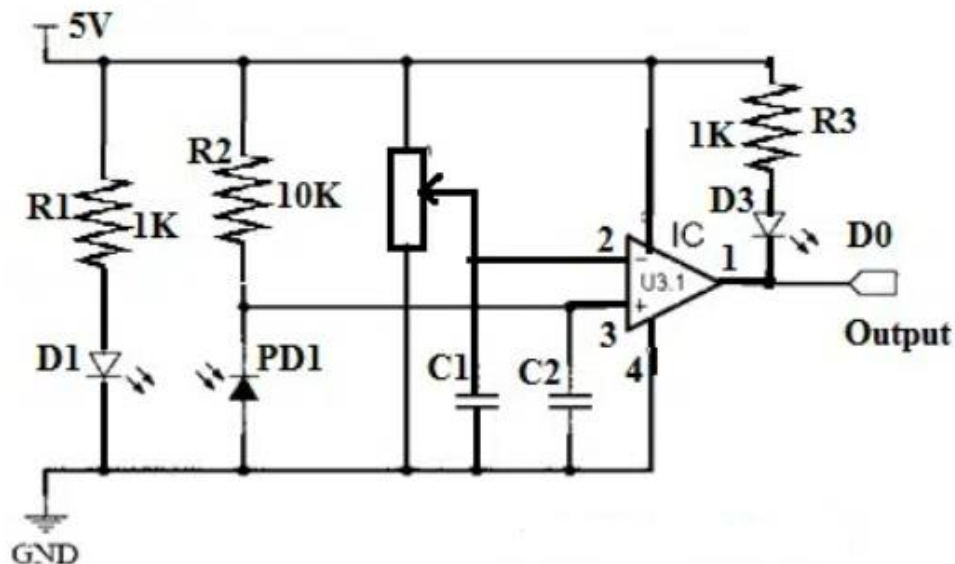


Рис. 2.3 - Принципова схема датчика полум'я

2.1.3 Переваги та недоліки датчиків полум'я

До переваг датчиків полум'я можна віднести наступне.

Ці датчики мають високу швидкість реагування.

Ці датчики стійкі до фальшивих тривог.

Цей датчик часто й точніше реагує швидше порівняно з датчиком тепла чи диму завдяки механізмам, які він використовує для виявлення полум'я.

Він має велику відстань виявлення, хорошу адаптивність до навколишнього середовища та високу надійність.

Цей датчик служить приблизно 5 років, а середній термін служби опалювальної системи становить 15-30 років.

До недоліків датчика полум'я можна віднести наступне.

УФ- чи ІЧ-детектори полум'я не можуть бути застосовані для неуглецевих пожеж і здатні виявляти пожежі, які створюють УФ- та ІЧ-випромінювання окремо.

Ціна оптичного датчика полум'я висока.

2.2 Принцип роботи GSM модулю

GSM або Глобальна система мобільного зв'язку є найпопулярнішим методом бездротового стільникового зв'язку, який використовується для публічного спілкування. Стандарт GSM був розроблений для встановлення протоколів для цифрових стільникових мереж другого покоління (2G).

Спочатку це було мережа з комутацією каналів, але пізніше була також реалізована комутація пакетів після інтеграції технології GPRS (General Packet Radio Service). Широко використовувані діапазони частот GSM – 900 МГц і 1800 МГц.

Архітектура GSM поділяється на радіопідсистему, підсистему мережі та комутації та операційну підсистему. Підсистема радіозв'язку складається з мобільної станції та підсистеми базової станції [14].

Мобільна станція – це, як правило, мобільний телефон, що складається з приймача, дисплея та процесора. Кожна кишенькова або портативна мобільна станція складається з унікального ідентифікатора, який зберігається в модулі, відомому як SIM (чип ідентифікації абонента). Це маленький

мікročіп, який вставляється в мобільний телефон і містить базу даних мобільної станції. (Див. Рис. 2.4).

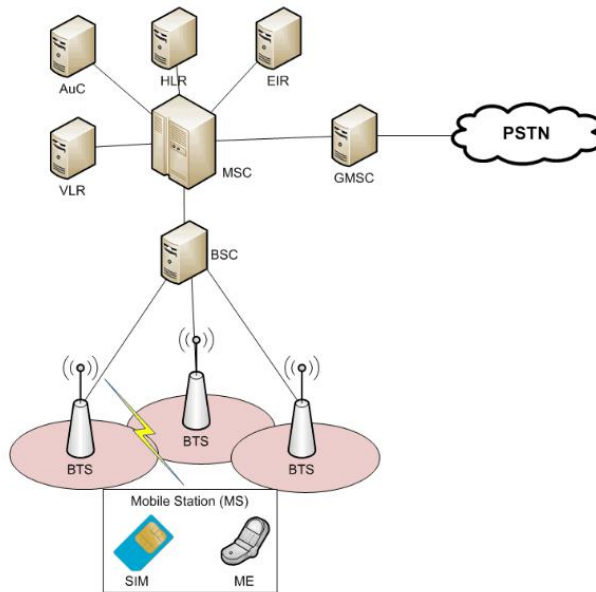


Рис. 2.4 – Архітектура мережі GSM

Він з'єднує мобільну станцію з підсистемою мережі через радіоінтерфейс.

Він складається з наступних заданих елементів:

Базова приймально-передавальна станція: одна або кілька базових приймально-передавальних станцій забезпечують фізичне підключення мобільної станції до мережі у формі радіоінтерфейсу. Залежно від навантаження, поведінки абонента та структури трансформації він може мати різні конфігурації: стандартну конфігурацію (кожній BTS призначається інший ідентифікатор комірки (CI), а кілька BTS утворюють зону розташування).

Конфігурація парасолькової комірки (один BTS з високою потужністю передачі, встановлений на більшій висоті, діючи як парасолька для базових станцій передавача з меншою потужністю передачі), суміжні конфігурації (кілька BTS, розміщених на одному місці, але антени охоплюють лише одну область 120 або 180 градусів). Це мережа сусідніх радіосот, які забезпечують повне покриття зони обслуговування.

Контролер базової станції: керує роботою ще однієї базової приймально-передавальної станції, в основному передачею або керуванням живленням. BSC підключається до BTS через інтерфейс Abis. Він складається з бази даних, що містить інформацію про стан технічного обслуговування BTS, якість радіо та наземних ресурсів, а також операційне програмне забезпечення BTS).

Блок транскодування та швидкості адаптації: розташований між контролером базової станції та центром комутації мобільних пристроїв. Стискає або розпаковує мовлення мобільної станції. Однак він не використовується для підключення до даних.

Підсистема комутації мережі: забезпечує повний набір функцій керування та бази даних, необхідних для встановлення виклику за допомогою функцій шифрування, автентифікації та роумінгу. Він в основному забезпечує мережеве підключення до мобільної станції. Він складається з наступних заданих елементів

Мобільний комутаційний центр: це головний елемент глобальної мережі GSM. Це як станція комутованої телефонної мережі загального користування (PSTN) або станція цифрової мережі інтегрованих послуг (ISDN). Окрім звичайної офіційної, він підтримує додаткові функції, такі як реєстрація, аутентифікація, визначення місця виклику та маршрутизація виклику до абонента.

Він забезпечує інтерфейси до телефонної мережі загального користування (PSTN) для підключення до фіксованої лінії або інтерфейс до іншого центру комутації мобільних пристроїв (MSC) для підключення до іншого мобільного телефону.

Реєстр домашніх розташувань: це сховище, у якому зберігаються дані, що належать великій кількості передплатників. По суті, це велика база даних, яка керує даними кожного абонента. З метою безпеки він зберігає

специфічний для абонента параметр, наприклад параметр Kі, відомий лише HLR та SIM.

Віртуальний реєстр розташування: схожий на реєстр домашнього розташування (HLR), але відрізняється тим, що він зберігає динамічну інформацію про дані передплатників. Він діє в разі роумінгу, коли абонент переїжджає з одного місця в інше. Інформація зберігається в реєстрі ідентифікації обладнання, який веде облік усіх мобільних станцій, кожен з яких ідентифіковано міжнародним ідентифікаційним номером мобільного обладнання (IMEI).

Глобальна система мобільного зв'язку (GSM) використовує комбінацію множинного доступу з тимчасовим поділом (TDMA) і множинного доступу з частотним поділом (FDMA).

Множинний доступ із розділенням частот: передбачає поділ смуги частот на кілька смуг таким чином, що кожна смуга частот, що розділяється, призначається одному абоненту. FDMA у GSM ділить смугу пропускання 25 МГц на 124 несучі частоти, кожен з яких розділено 200 кГц. Кожній базовій станції призначається одна або кілька несучих частот.

Множинний доступ із тимчасовим поділом: передбачає призначення того самого частотного каналу різним абонентам шляхом поділу частотного діапазону на кілька часових інтервалів. Кожен користувач отримує власний часовий проміжок, що дозволяє кільком станціям спільно використовувати той самий простір для трансляції.

Для GSM кожна розділена несуча частота ділиться на різні часові інтервали за допомогою технології TDMA. Кожен кадр TDMA має довжину 4,164 мілісекунди (мс) і містить 8 часових інтервалів. Кожен часовий інтервал або фізичний канал у цьому кадрі триває 577 мікросекунд, і дані передаються у часовому інтервалі пакетами (Див. Рис. 2.5).

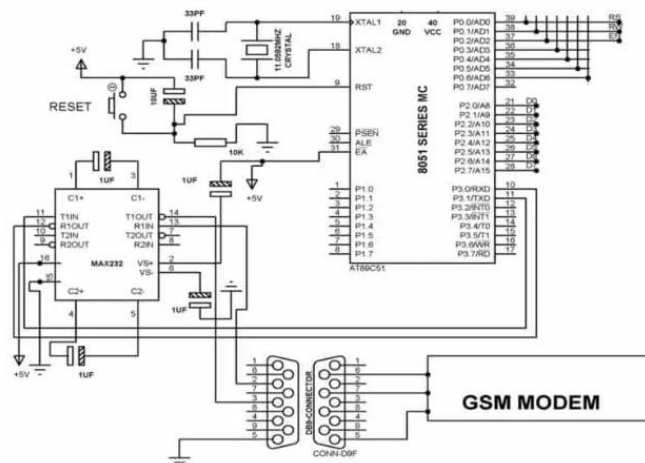


Рис. 2.5 - Схема модему GMS – робота модуля GSM

2.2.1 Переваги та недоліки GSM модулю

Переваги GSM:

- Найбільш підходяща мережа з надійними характеристиками.
- Без плати за роумінг на міжнародні дзвінки.
- Підключення по всьому світу та широке покриття.
- Методи SAIC і DAIC, що використовуються в GSM, забезпечують дуже високу якість передачі.
- Телефон заснований на SIM-карті, тому користувачам легко перемикається між різними типами телефонів.
- Сигнал GSM не має жодних погіршень.
- Легко інтегрувати GSM з іншими бездротовими технологіями, такими як CDMA та LTE.
- Має можливість використовувати повторювачі.
- Через імпульсний характер передачі тривалість розмови зазвичай велика.

Недоліки GSM:

- Затримка пропускну здатності через те, що кілька користувачів використовують однакову пропускну здатність, щоб потік міг знайти інтерфейс.

- Може створювати перешкоди певним електронним пристроям, наприклад слуховим апаратам, завдяки технології передачі імпульсів. Як результат, у багатьох місцях, наприклад у лікарнях, аеропортах і заправних станціях, потрібно вимкнути мобільні телефони.

- Для збільшення покриття необхідна установка репітерів.

- Він забезпечує обмежену швидкість передачі даних, тому для високої швидкості передачі даних використовуються вдосконалені версії пристроїв GSM.

- Велика частина технології GSM є власністю Qualcomm, тому необхідно отримати ліцензію від них.

- Виробники не випускають прилади IC-95 через відсутність великого ринку, тому IC-95 зазвичай встановлюють у малій вежі.

- GSM має максимальний фіксований діапазон дзвінків до 35 км, що дуже обмежено.

- Немає наскрізного шифрування даних користувача.

- Різні несумісності зі стандартами GSM.

- Електромагнітне випромінювання більше при використанні GSM.

- Макростільники, на які впливає втрата багатопроменевого сигналу.

2.3 Метод виявлення та розпізнавання пожежі

Метод виявлення пожежі буде розглянуто на прикладі набору даних а саме зображень.

База даних підготовки містить 9200 зображень денних і нічних лам, як показано в таблиці 1. Ми зібрали їх із відкритих баз даних і зображень Google у різних умовах (розмір, колір і форма). У будь-якому випадку, з цими наборами даних неможливо досягти нормального результату, і ми можемо не досягти очікуваної точності розпізнавання вогню в середовищах реального часу. Щоб вирішити цю проблему, ми прагнемо використовувати метод трансформації зображення, щоб збільшити кількість зображень сцен пожежі в наборі даних. У наступному підрозділі вичерпно описано створення набору даних [15].

Обертання всіх зображень вручну та їх повторне маркування вимагає значного часу обробки. Тому ми створили спеціальне програмне забезпечення для цього, використовуючи теорію матриць трансформації у рамках OpenCV. Перетворення зображення може бути виражене у формі множення матриці за допомогою афінного перетворення. Афінне перетворення широко використовується для виконання операцій обертання, і його представляють наступні матриці 2×3 :

$$M = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix}$$

Представлення двовимірного (2D) вектора, який ми хочемо перетворити, виглядає наступним чином.

$$X = \begin{bmatrix} x \\ y \end{bmatrix}$$

Ми можемо отримати перетворений вектор наступним чином.

$$T = M * [(x, y, 1)]^T$$

$$M = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix}$$

У наведених вище матрицях T вказує на вектор перетворення, а M вказує на матрицю обертання.

Перетворення матриці можна виразити через кут повороту та координати центру.

$$\begin{bmatrix} \alpha & \beta & (1 - \alpha) * centre.x - \beta * centre.y \\ -\beta & \alpha & \beta * centre.x + (1 - \alpha) * centre.y \end{bmatrix}$$

У цьому перетворенні $\alpha = scale * \cos\theta$

, $\beta = масштаб * \sin\theta$

, і θ

є кутом повороту. Ми обчислюємо масштаб = 1. Зазвичай розмір зображення та опорна точка змінюються після повороту, і матрицю перетворення потрібно змінити (Див. Рис. 2.6).

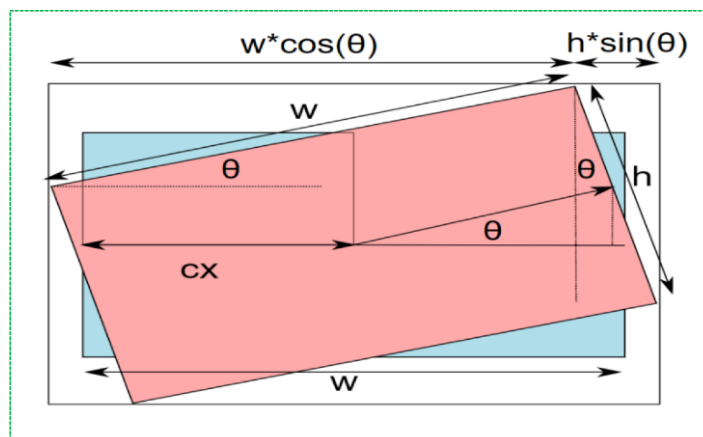


Рис. 2.6 - Схема повороту зображення.

Згодом ширина та висота нового розміру зображення були розраховані за допомогою наступного рівняння.

$$\text{new.width} = h * \sin(\theta) + w * \cos(\theta)$$

$$\text{new.height} = h * \cos(\theta) + w * \sin(\theta)$$

Нарешті, коли зображення змінюється, координати точки повороту (центру зображення) також змінюються. Щойно отримані координати центру зображення обчислюються за допомогою наступної матриці перетворення:

$$\begin{bmatrix} \alpha & \beta & (1 - \alpha) * \text{centre. x} - \beta * \text{centre. y} + \left(\text{new. } \frac{\text{width}}{2} - \text{centre. x} \right) \\ -\beta & \alpha & \beta * \text{centre. x} + (1 - \alpha) * \text{centre. y} + (\text{new. height} / 2 - \text{centre. y}) \end{bmatrix}$$

де $\alpha = \cos \theta$, $\beta = \sin \theta$ і θ – кут повороту. Під час обертання зображення ми виявляємо ефект згладжування та дірки на вихідному зображенні. Цю проблему можна легко вирішити за допомогою операції інтерполяції. На відміну від наших попередніх досліджень, ми повернули всі зображення вогню на 90° , 180° і 270° . (Див. Рис. 2.7).

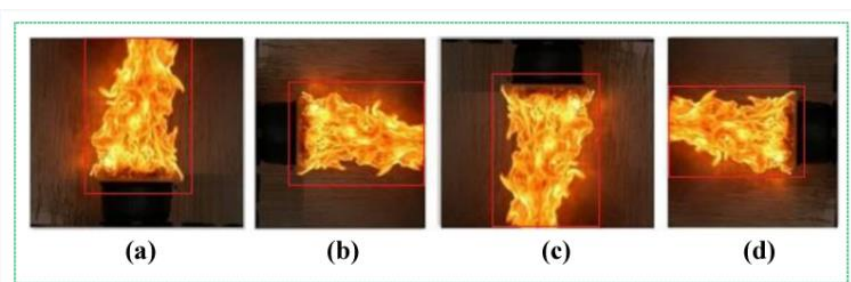


Рис. 2.7 - (a) Оригінальне зображення; (b) поворот на 90° (a); (c) поворот на 180° (a) і (d) поворот на 270° (a).

Після застосування цього методу ми збільшуємо зображення в наборі даних у три рази більше, ніж кількість вихідних збільшених кадрів. Як згадувалося вище, наші дані склалися з 9200 оригінальних зображень. Загальний розмір наборів даних збільшився до 27 600 після ефективного методу масштабування, використаного на цьому етапі. Крім того, для запобігання хибнопозитивних результатів додано понад 10 000 зображень, схожих на вогонь.

Щоб оцінити продуктивність методу виявлення пожежі, модель була протестована в різних сценаріях пожежі в приміщенні та на вулиці. У попередньому розділі обговорюються різні експерименти, які були проведені та реалізовані за допомогою систем класифікаторів YOLO. Тут ми зосереджуємося на аналізі ефективності звичайних методів виявлення пожежі та запропонованому підході з точки зору якісних і кількісних результатів. Нижче показано приклади видимих експериментів у денному та нічному середовищі з використанням екстрактора функцій CSPDarknet-53

(Див. Рис. 2.8). CSPDarknet-53 — це CNN і магістраль для виявлення об'єктів за допомогою DarkNet-53. Він використовує стратегію CSPNet, щоб розділити карту функцій базового рівня на дві частини, а потім об'єднати їх через ієрархію між етапами. Використання стратегії розділення та злиття забезпечує більш градієнтний потік через мережу (Див. Рис. 2.9).



Рис. 2.8 - Видимі результати в денних сценах пожежі

Де:

(а) послідовності вхідних зображень;

(б) послідовності вихідних зображень із виявленими областями пожежі.



Рис. 2.9 - Видимі результати в нічних сценах пожежі

Де:

(а) вхідні послідовності зображень;

(б) вихідні послідовності зображень з виявленими областями пожежі.

Експерименти показали, що наш вдосконалений пожежний сповіщувач може зменшити занепокоєння людей і забезпечити раннє придушення та

швидке реагування незалежно від часу доби, форми чи розміру пожежі. Важко запобігти пожежі на ранніх стадіях через хибні тривоги від традиційних пожежних сповіщувачів, таких як ті, що спрацьовують від рухомих об'єктів, які демонструють значення інтенсивності пікселів і кольори, подібні до тих областей пожежі на фоні.

Щоб оцінити та проаналізувати ефективність випадків розпізнавання пожежі, ми порівняли запропонований підхід з нещодавно опублікованими методами виявлення пожежі. Для виконання цього завдання ми використовуємо широко використовувані метрики оцінки для виявлення об'єктів (статичні або динамічні), як і в наших попередніх публікаціях [16,56]. По-перше, ми обчислюємо точність і показники відкликання, щоб виміряти продуктивність детектора об'єктів у визначенні дійсності виявлення. Точність — здатність класифікатора ідентифікувати лише релевантні об'єкти; це частка виявлених справжніх позитивних результатів. Відновлення вимірює здатність моделі ідентифікувати всі відповідні випадки; це частка істинних позитивів, виявлених серед усіх основних істин. Хороша модель — це та, яка може ідентифікувати більшість реальних об'єктів (демонструє високу запам'ятовуваність), водночас ідентифікуючи лише відповідні об'єкти (демонструє високу точність). Ідеальною моделлю є модель із хибнонегативним значенням = 0 (пригадування = 1) і хибнопозитивним значенням = 0 (точність = 1). Ми використовуємо наступні рівняння для оцінки середньої точності та показників відкликання систем пожежної розвідки:

$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{Recall} = \frac{TP}{TP + FN}$$

де TP означає справжні спрацьовування, що позначають точно знайдені пікселі полум'я, FP означає хибні спрацьовування, а FN вказує на кількість хибнонегативних.

Оцінка моделі виявлення об'єктів за допомогою точності та запам'ятовування може дати цінну інформацію про продуктивність моделі при різних значеннях довіри. Оцінка F-міри (FM) особливо корисна для визначення оптимального достовірного балансування точності та значень запам'ятовування для методів тестування та може бути оцінена за допомогою рівняння, яке враховує як точність, так і запам'ятовування.

$$FM = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Загалом, для аналізу продуктивності моделей рекомендується використовувати перевірочний набір (набір даних, який використовується для підгонки гіперпараметрів) і тестовий набір (набір даних, який використовується для оцінки продуктивності повністю навченої моделі). Функція класифікації лише оцінює ймовірність того, що об'єкт класу з'явиться на зображенні, що є простим завданням для класифікатора, щоб визначити правильні прогнози з неправильних прогнозів. Однак завдання виявлення об'єкта далі визначає місцезнаходження об'єкта через пов'язану обмежувальну рамку з відповідним показником достовірності, щоб повідомити, наскільки точно визначено обмежувальну рамку класу об'єктів.

Висновки до розділу 2

Під час оформлення розділу 2, було опановано та проаналізовано матеріал щодо методів та принципів роботи датчиків полум'я та GSM модулів а також їх різновидностей датчиків полум'я та принципів їх роботи.

Датчик полум'я - це тип детектора, який використовується для виявлення, а також реагування на пожежу або полум'я. Датчик полум'я часто реагує швидше і точніше в порівнянні з датчиком тепла або диму через механізми, які він використовує для виявлення полум'я.

До переваг використання датчику полум'я можна віднести: низьку вартість, невеликий розмір, має високу швидкість реагування, широкий

діапазон дії виявлення, високу надійність та адаптивність до навколишнього середовища. А до недоліків: може бути застосовані для неуглецевих пожеж і здатні виявляти пожежі, які створюють УФ- та ІЧ-випромінювання окремо, ціна саме оптичного датчику полум'я достатньо висока.

GSM або Глобальна система мобільного зв'язку є найпопулярнішим методом бездротового стільникового зв'язку, який використовується для публічного спілкування. Стандарт GSM був розроблений для встановлення протоколів для цифрових стільникових мереж другого покоління (2G).

До переваг використання GSM модулю можна віднести: невеликий розмір і низьку вартість, високу точність, низьке енергоспоживання, високу надійність, підключення по всьому світу та широке покриття, методи SAIC і DAIC, що використовуються в GSM, забезпечують дуже високу якість передачі. Недоліки включають: Для збільшення покриття необхідна установка репітерів, забезпечує обмежену швидкість передачі даних, тому для високої швидкості передачі даних використовуються вдосконалені версії пристроїв GSM.

Наприкінці було проаналізовано метод та математичні рівняння для виявлення пожежі за допомогою датчику полум'я. Поєднання даних з датчику полум'я та набору даних згідно цього метода може значно покращити виявлення пожежі та з допомогою GSM модулю тим самим інформування користувача.

3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Детальний опис апаратної платформи та її компонентів

Arduino Nano — це тип плати мікроконтролера, розроблений Arduino.cc. Його можна створити за допомогою мікроконтролера, такого як Atmega328. Цей мікроконтролер також використовується в Arduino UNO. Це плата невеликого розміру, але також гнучка з широким спектром застосувань. Інші плати Arduino в основному включають Arduino Mega, Arduino Pro Mini, Arduino UNO, Arduino YUN, Arduino Lilypad, Arduino Leonardo та Arduino Due. Інші плати розробки: плата розробки AVR, плата розробки PIC, Raspberry Pi, Intel Edison, MSP430 Launchpad і плата ESP32 (Див. Рис. 3.1).

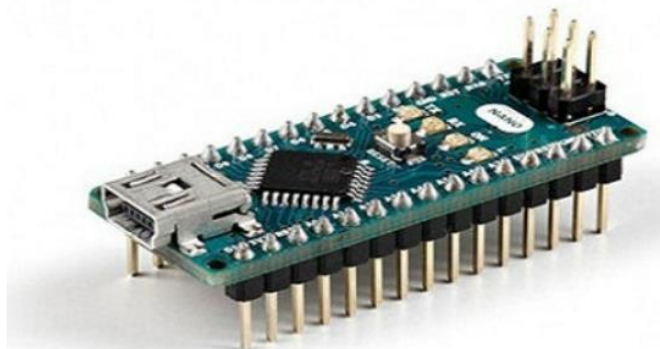


Рис. 3.1 – Плата Arduino Nano

Ця плата має багато функцій і особливостей, як плата Arduino Duemilanove. Однак ця нанодошка має іншу упаковку. Він не має роз'єму постійного струму, тому джерело живлення може забезпечуватися невеликим USB-портом, інакше він підключається безпосередньо до контактів, таких як VCC і GND. Цю плату можна живити від 6 до 20 вольт за допомогою міні-порту USB на платі [16].

Особливості Arduino nano в основному включають наступне.

Мікроконтроллер ATmega328P відноситься до сімейства 8-розрядних AVR

Робоча напруга 5В

Вхідна напруга (V_{in}) становить від 7 В до 12 В

Вхідні/вихідні контакти 22

Аналогові і/р контакти 6 від А0 до А5

Цифрових контактів 14

Споживана потужність 19 мА

Контакти введення/виведення Постійний струм становить 40 мА

Флеш-пам'ять - 32 Кб

SRAM становить 2 Кб

EEPROM становить 1 Кб

Частота CLK становить 16 МГц

Вага-7г

Розмір друкованої плати 18 x 45 мм

Підтримує три комунікації, як-от SPI, ІІС та USART

Arduino nano конфігурація контактів показана нижче (Див. Рис. 3.2), і функції кожного контакту обговорюються нижче.

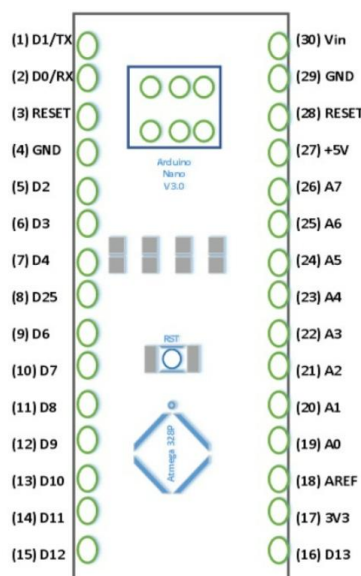


Рис. 3.2 – Конфігурація пінів Ардуїно Нано

V_{in} - це вхідна напруга плати, і воно використовується, коли використовується зовнішнє джерело живлення від 7 В до 12 В.

5 В — це регульована напруга джерела живлення наноплати, і вона використовується для живлення плати, а також компонентів.

3,3 В - це мінімальна напруга, яка генерується регулятором напруги на платі.

GND — контакт заземлення плати

Вивід RST (Reset): Цей контакт використовується для скидання мікроконтролера

Аналогові контакти (A0-A7): Ці контакти використовуються для розрахунку аналогової напруги плати в діапазоні від 0 В до 5 В

Контакти введення/виведення (цифрові контакти від D0 до D13): ці контакти використовуються як контакти і/р, інакше як контакти о/р. 0 В і 5 В

Послідовні контакти (Tx, Rx): ці контакти використовуються для передачі та отримання послідовних даних TTL.

Зовнішні переривання (2, 3): ці контакти використовуються для активації переривання.

ШІМ (3, 5, 6, 9, 11): Ці контакти використовуються для забезпечення 8-бітного виходу ШІМ.

SPI (10, 11, 12 і 13): ці контакти використовуються для підтримки зв'язку SPI.

Вбудований світлодіод (13): цей контакт використовується для активації світлодіода.

ІС (A4, A5): ці контакти використовуються для підтримки зв'язку TWI.

AREF: Цей висновок використовується для надання опорної напруги вхідній напрузі.

3.1.1 Arduino Nano Зв'язок

Зв'язок між платою Arduino Nano можна здійснювати за допомогою різних джерел, наприклад, за допомогою додаткової плати Arduino, комп'ютера чи іншим чином за допомогою мікроконтролерів. Мікроконтролер, який використовується на платі Nano (ATmega328), забезпечує послідовний зв'язок (UART TTL). Це може бути доступно на

цифрових контактах, таких як TX і RX. Програмне забезпечення Arduino складається з послідовного монітора, який дозволяє легко передавати та отримувати текстову інформацію з плати [17].

Світлодіоди TX і RX на платі Nano блиматимуть щоразу, коли дані надсилаються через канали FTDI та USB у напрямку комп'ютера. SoftwareSerial, подібно до бібліотеки, забезпечує послідовний зв'язок на будь-якому з цифрових контактів на платі. Мікроконтролер також підтримує зв'язок SPI і I2C (TWI).

3.1.2 Програмування Arduino Nano

Програмувати Arduino nano можна за допомогою програмного забезпечення Arduino. Треба натиснути на опцію Інструменти та вибрати наноплату. Мікроконтролер ATmega328 на платі Nano поставляється з попередньо запрограмованим завантажувачем. Цей завантажувач дозволяє завантажувати новий код без використання зовнішнього апаратного програматора. Передача цього може здійснюватися за допомогою протоколу STK500. Тут також можна обійти завантажувач, а програму мікроконтролера можна виконати за допомогою In-Circuit Serial Programming або заголовка ICSP з Arduino ISP.

3.1.3 Модуль інфрачервоного датчику полум'я

Для визначення та розпізнавання пожежі було обрано інфрачервоний датчик полум'я.

Детектор полум'я - це датчик, призначений для виявлення та реагування на наявність полум'я або пожежі. Реакція на виявлене полум'я залежить від установки, але може включати звуковий сигнал, відключення паливної лінії (наприклад, лінії пропану або природного газу) та активацію системи пожежогасіння (Див. Рис. 3.3).

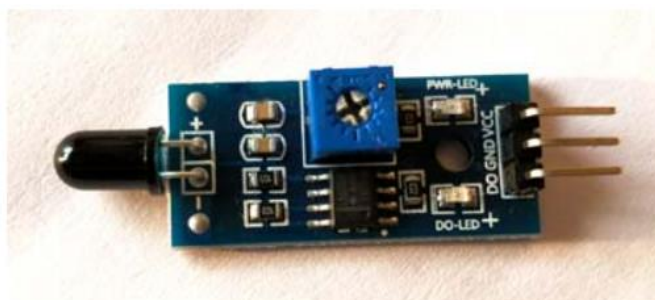


Рисунок 3.3 – Модуль інфрачервоного датчику полум'я

У табл. 3.1 наведено технічні характеристики модуля датчику полум'я.

Таблиця 3.1 – Технічні характеристики модуля датчику полум'я

Робоча напруга	3.3V – 5V
Робочий струм	15 mA
Мікросхема компаратора	LM393
Тип датчика	Фототранзистор YG1006
Діапазон спектру	760 нм ~ 1100 нм
Кут виявлення	0-60 градусів
Робоча температура	-25°C ~ 85°C
Розмір датчику	3 см X 1,6 см

Коли вогонь горить, він випромінює невелику кількість інфрачервоного світла, це світло буде прийнято фотодіодом (ІЧ-приймачем) у модулі датчика. Потім ми використовуємо операційний підсилювач, щоб перевірити зміну напруги на ІЧ-приймачі, щоб у разі виявлення пожежі вихідний (DO) штифт давав 0 В (LOW), а якщо пожежі немає, вихід (DO)) висновок дасть 0 В (Низький), і якщо пожежі немає, вихідний висновок буде 5 В (Високий).

Він заснований на датчику YG1006, який є високошвидкісним і високочутливим кремнієвим фототранзистором NPN. Він може виявляти інфрачервоне світло з довжиною хвилі від 700 нм до 1000 нм, а його кут виявлення становить близько 60°. Модуль датчика полум'я складається з фотодіода (інфрачервоного приймача), резистора, конденсатора,

потенціометра та компаратора LM393 на інтегральній схемі. Чутливість можна регулювати за допомогою вбудованого потенціометра. Робоча напруга від 3,3 В до 5 В постійного струму, з цифровим виходом. Логіка високого рівня на виході вказує на наявність полум'я або пожежі. Логічний низький вихід вказує на відсутність полум'я або пожежі.

3.1.4 Модуль GSM SIM800L

SIM800L — це мініатюрний стільниковий модуль, який дозволяє передавати GPRS, надсилати й отримувати SMS, здійснювати й отримувати голосові дзвінки. Низька вартість, невеликий розмір і підтримка чотирьох діапазонів частот роблять цей модуль ідеальним рішенням для будь-якого проекту, який потребує підключення на великій відстані. Після підключення модуля живлення він завантажується, шукає стільникову мережу та автоматично авторизується. Вбудований світлодіод показує стан підключення (немає покриття мережі, швидко блимає, увійшли в систему, повільно блимає) (Див. Рис. 3.4).



Див. Рис. 3.4 – Модуль GSM

Таблиця 3.2 – Технічні характеристики модуля датчику GSM

Напруга живлення	3.8V - 4.2V
Рекомендована напруга живлення	4V
Споживання енергії	сплячий режим < 2,0 мА режим очікування < 7,0 мА
Передача GSM (середнє)	350 мА

Передача GSM (пік)	2000 мА
Розмір модуля	25 x 23 мм
Інтерфейс	UART (макс. 2,8 В) і AT-команди
Підтримувані частоти	Quad Band (850/950/1800/1900 МГц)
Діапазон робочих температур	від -40 до + 85 °С

3.1.5 Живлення

В якості живлення було обрано Літій-іонне батарейку LP18650A. Елементи живлення електрохімічної схеми Li-SOCL2 гарантують довгий термін використання при температурах від -55° до 85° [18].

На рис. 3.5 продемонстровано зовнішній вигляд обраного джерела живлення.



Рисунок 3.5 – Літій-іонна батарейка для живлення плати

У табл. 3.4 наведено характеристики літій-тіоніл-хлоридного джерела живлення.

Таблиця 3.4 – Характеристики Li-SOCL2 джерела живлення

Ємність	2400 мА/год
Найвища робоча температура	85° С
Найнижча робоча температура	-55° С
Напруга	3,6 В

3.3 Побудова макетної схеми пристрою

Перед початком будівництва прототипу схему і принципові діаграми були розроблені в середовищі проектування Fritzing.

Fritzing — це широко використовувана платформа програмного забезпечення з відкритим кодом, розроблена спеціально для електронного прототипування та проектування схем. Платформа пропонує повний та інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам створювати та тестувати електронні схеми за допомогою візуального представлення.

Програмне забезпечення надає велику бібліотеку електронних компонентів, включаючи мікроконтролери, датчики, виконавчі механізми та різні інші компоненти, які можна легко розмістити на віртуальній платі. З'єднання між компонентами можна встановити та перевірити, а поведінку схеми можна змодельювати для цілей тестування.

Після успішного встановлення всіх необхідних для комплексу компонентів, було спроектовано макетну та принципові схеми прототипу (Див. рис. 3.6).

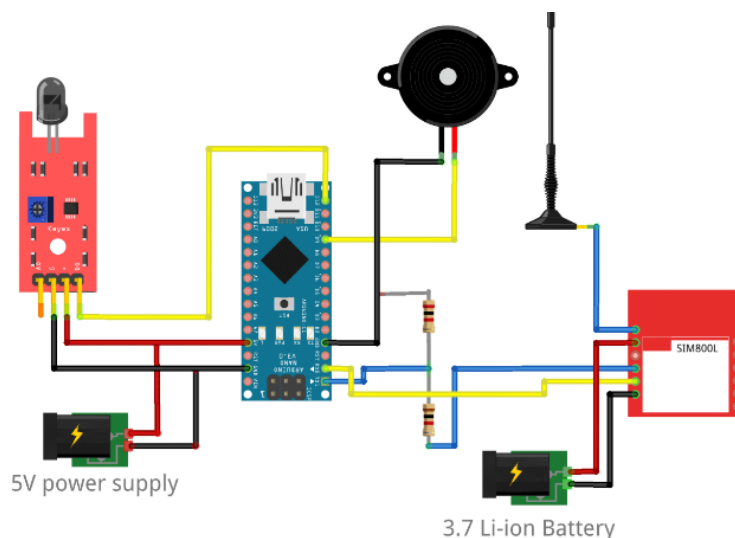


Рисунок 3.6 – Макетна схема прототипу

3.4 З'єднання модуля датчику полум'я з апаратною платформою

Після того як була побудована макетна схема, було розпочато конструювання прототипу. До плати Arduino Nano було під'єднано модуль інфрачервоного датчику полум'я:

Датчик полум'я..... Arduino Nano

VCC 5 В

GND GND

A0..... Аналоговий пін.

Результат продемонстровано на рис. 3.7

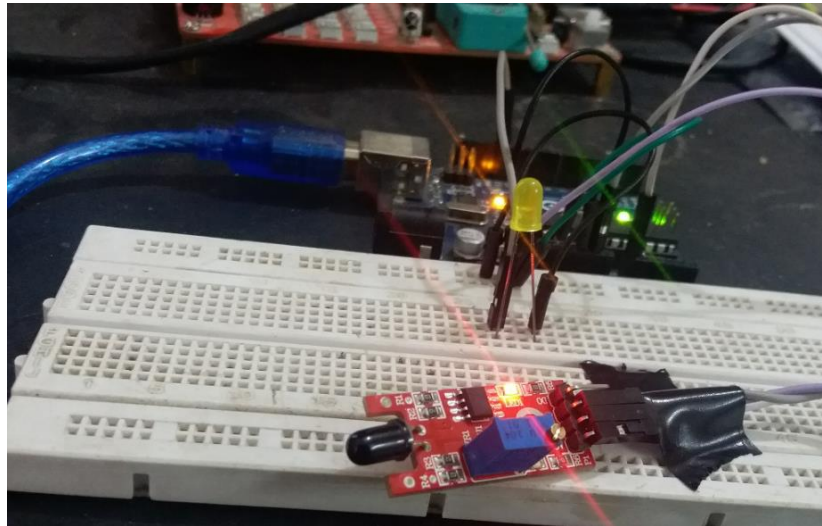


Рисунок 3.7 – Результат під'єднання модуля до плати Arduino Nano

3.5 Розробка програмної частини

Перед початком розробки програмної частини комплексу, у онлайн середовищі draw.io було розроблено алгоритм за яким потрібно щоб програмна частина працювала. На рис. 3.8 зображено алгоритм роботи програмної частини.

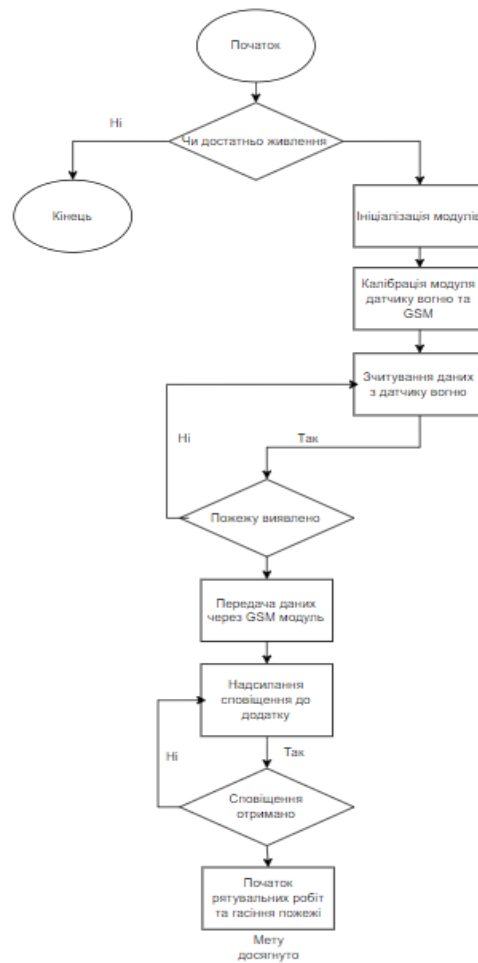


Рис. 3.8 – Алгоритм роботи програми

На даному рисунку можна побачити що для початку роботи комплексу іде перевірка на те чи достатньо подається живлення, потім іде ініціалізація модулів а саме інфрачервоного датчику вогню та GSM модулю і наступним кроком відбувається калібрування даних модулів. Потім відбувається зчитування даних з датчику вогню, якщо вогонь не виявлено алгоритм повертається до минулого кроку, якщо вогонь успішно виявлено то відбувається передача даних про пожежу через GSM модуль та йду надсилання даних до додатку, якщо воно отримано то пожежники та користувач проінформовані про пожежу та слідом починаються рятувальні роботи та гасіння пожежі, роботу програми можна рахувати успішною в такому алгоритму подій.

3.5.1 Вибір мови програмування

Для розробки програмної частини комплексу була обрана мова C. Використання мови програмування C з Arduino Nano має декілька переваг:

Переносна мова

C є портативною мовою. Він був розроблений для компіляції за допомогою різних компіляторів і може працювати на більшості апаратних платформ, що означає, що не потрібно знати все про розташування пам'яті машини, щоб мати можливість писати для неї програми.

Структурована мова програмування

C є структурованою мовою програмування, що означає, що вона має відкриваючу та закриваючу дужки для кожного блоку коду. Програміст може використовувати додатковий простір, наданий цими фігурними дужками, щоб зробити відступ у своєму коді в кожній функції, що полегшить читання.

Підтримка сумісності

C є широко використовуваною мовою, яка підтримується великою кількістю інструментів розробки та бібліотек. Це дозволяє легко інтегрувати код, написаний на C, з іншими компонентами, такими як датчики, виконавчі механізми та інші мікроконтролери;

Динамічний розподіл пам'яті

C підтримує динамічний розподіл пам'яті, що означає, що програміст може створювати та знищувати розділи пам'яті за потреби. Це важливий аспект сучасних мов програмування, оскільки неможливо заздалегідь знати, скільки місця знадобиться.

Швидкодія

C – це скомпільована мова, яка відома своєю швидкістю виконання. Програми, написані на C, можуть працювати швидше і ефективніше, ніж програми, написані на мовах вищого рівня, що робить її добре придатною для середовищ з обмеженими ресурсами, таких як Arduino Nano.

3.5.2 Вибір середовища розробки

В якості середовища розробки для написання програмної частини було використано середовище Arduino IDE.

Arduino IDE — це програмне забезпечення з відкритим кодом, яке використовується для написання та завантаження коду на плати Arduino. Програма IDE підходить для різних операційних систем, таких як Windows, Mac OS X і Linux. Підтримує мови програмування C і C++. Тут IDE означає інтегроване середовище розробки [19].

Програму або код, написаний в Arduino IDE, часто називають ескізом. Нам потрібно підключити плату Genuino та Arduino до IDE, щоб завантажити написаний ескіз у програмне забезпечення Arduino IDE. Ескіз зберігається з розширенням «.ino».

Дозволяє керувати ескізами з кількома файлами (кожен з яких відображається на окремій вкладці). Це можуть бути звичайні файли коду Arduino (без видимого розширення), файли C (розширення .c), файли C++ (.cpp) або файли заголовків (.h).

Перед компіляцією ескізу всі звичайні файли коду Arduino в ескізі (.ino, .pde) об'єднуються в один файл у порядку відображення вкладок. Інші типи файлів залишаються як є (Див. Рис. 3.9).

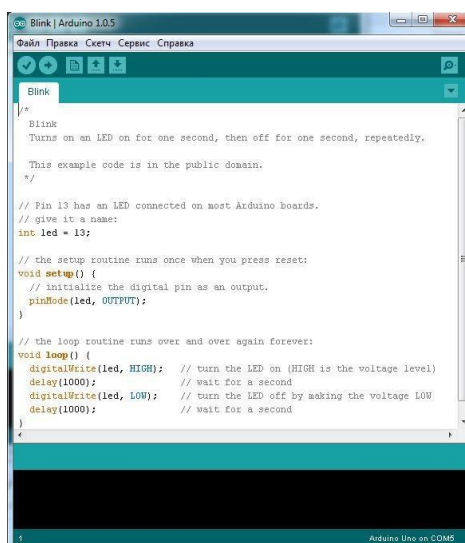


Рис. 3.9 – Середовище розробки Arduino IDE

Бібліотеки надають додаткові функції для використання в ескізах, напр. працювати з обладнанням або маніпулювати даними. Щоб використати бібліотеку в ескізі, виберіть її в меню «Ескіз» > «Імпортувати бібліотеку». Це вставить один або кілька операторів `#include` у верхній частині ескізу та компілює бібліотеку з вашим ескізом. Оскільки бібліотеки завантажуються на дошку разом із вашим ескізом, вони збільшують обсяг місця, який він займає. Якщо макет більше не потребує бібліотеки, просто видаліть його оператори `#include` у верхній частині коду.

3.5.3 Бібліотека SoftwareSerial

Arduino реалізує апаратну підтримку для послідовного інтерфейсу даних через контакти 0 і 1 (які також використовуються для зв'язку з комп'ютером через USB). Апаратна робота з послідовним інтерфейсом здійснюється за допомогою вбудованого в мікроконтролер спеціального пристрою, який називається трансивер UART. Це дозволяє мікроконтролеру Atmega обробляти вхідні дані, навіть коли ви працюєте над іншими завданнями.

Бібліотека SoftwareSerial дозволяє реалізувати послідовний інтерфейс на будь-якому цифровому контакті Arduino за допомогою програмних засобів, які дублюють функціональні можливості UART (звідси назва «SoftwareSerial»). Бібліотека дозволяє програмно створювати декілька послідовних портів, які працюють на швидкості до 115200 бод. Для пристроїв, які працюють з інвертованим сигналом, бібліотека надає відповідний параметр, який включає інверсію.

3.5.4 Бібліотека String.h

String - це масив символів. Мова C надає нам бібліотеку `string.h` для маніпулювання рядками та роботи з ними. Крім того, існують різні типи даних і макроси для роботи з рядками. Ця бібліотека `string.h` широко

використовується для виконання таких операцій, як порівняння, конкатенація та копіювання.

Знайомство зі `string.h` на C дуже важливо для програмування на мові C, оскільки він широко використовується для маніпулювання рядками та роботи з ними.

У бібліотеці `string.h` визначено так багато функцій, за допомогою яких ми можемо виконувати такі операції, як порівняння двох рядків, об'єднання рядків, копіювання одного рядка в інший і виконання різноманітних операцій із маніпулюванням рядками.

Функції `string.h` у C використовуються не лише для обробки рядків або маніпулювання ними, але також використовуються для різних операцій обробки пам'яті.

3.5.5 Використанні змінні та функції

У `setup()` виконуються всі основні ініціалізації, такі як послідовні ініціалізації, ініціалізації модулів SIM800L і оголошення пінів GPIO.

```
void setup()
{
  pinMode(12, OUTPUT);
  pinMode(9, INPUT);
  gprsSerial.begin(9600);           // швидкість передачі даних GPRS
  Serial.begin(9600);             // швидкість передачі даних GPRS
  Module_Init();
}
```

У функції ініціалізації модуля SIM800L викликаються різні AT-команди, щоб ініціалізувати модуль і дізнатися статус модуля. Функціональні можливості окремих AT-команд можна знайти в таблиці AT-команд SIM800L.

```
void Module_Init()
{
  gprsSerial.println("AT");
  delay(1000);
  gprsSerial.println("AT+CPIN?");
}
```

```
delay(1000);
gprsSerial.println("AT+CREG?");
delay(1000);
gprsSerial.println("AT+CGATT?");
delay(1000);
gprsSerial.println("AT+CIPSHUT");
delay(1000);
gprsSerial.println("AT+CIPSTATUS");
delay(2000);
gprsSerial.println("AT+CIPMUX=0");
delay(2000);
ShowSerialData();
gprsSerial.println("AT+CSTT=\"www\"");
delay(1000);
ShowSerialData();
gprsSerial.println("AT+CIICR");
delay(3000);
ShowSerialData();
gprsSerial.println("AT+CIFSR");
delay(2000);
ShowSerialData();
gprsSerial.println("AT+CIPSPRT=0");
delay(3000);
ShowSerialData();
}
```

Усередині loop() цифрові значення виводу 12 зчитуються та зберігаються в змінній.

```
int fire = digitalRead(12);
```

Потім, коли пожежу виявлено, використовується цикл if-else для виявлення тригера SIM800L для виконання необхідних дій. Як показано нижче, AT+CIPSTART використовується для підключення до сервера Thingspeak і запуску з'єднання. AT+CIPSEND використовується для надсилання даних на сервер.

```
gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"); //
запустити з'єднання
delay(6000);
ShowSerialData();
```

```

gprsSerial.println("AT+CIPSEND");//begin send data to remote server
delay(4000);
ShowSerialData();
String          str          =          "GET
https://api.thingspeak.com/update?api_key=ER43PWXXXXQF0I&field1=" + String(1);
Serial.println(str);
gprsSerial.println(str);//почати посилати дані на віддалений сервер

```

Після завершення передачі даних AT+CIPSHUT використовується для закриття з'єднання.

```

gprsSerial.println("AT+CIPSHUT");//закрити з'єднання
delay(100);

```

3.6 Діаграми для пояснення роботи комплексу

На рисунку 3.10 зображено діаграму use-case роботи даного комплексу [20].

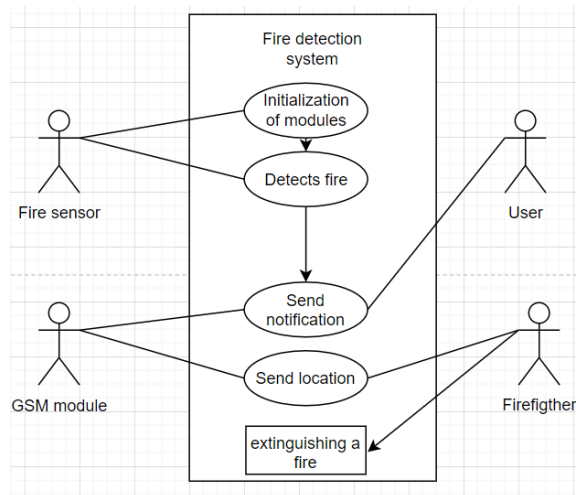


Рис. 3.10 – Діаграма use-case

На рисунку 3.11 зображено діаграму послідовностей роботи апаратно-програмного комплексу.

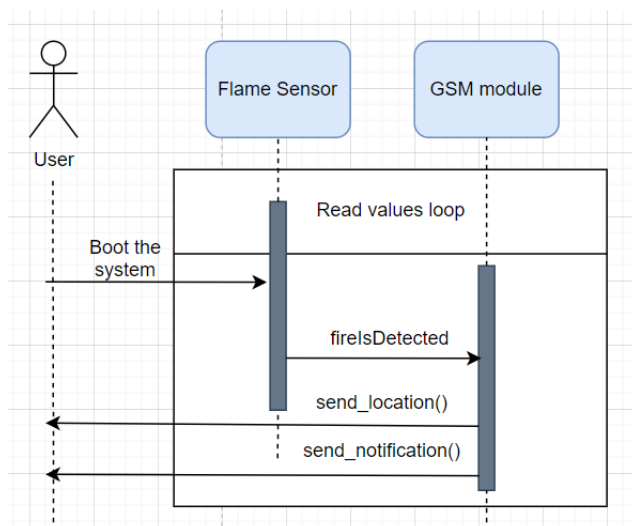


Рис. 3.11 – Діаграма послідовностей

Висновки до розділу 3

На початку третього розділу було наведено характеристику для обраної апаратної платформи а саме мікроконтролеру Arduino Nano. Далі було детально описано характеристики обраної платформи та модулів які були використані під час проектування та конструювання прототипу. Програмний додаток було розроблено на мові програмування C у середовищі розробки Arduino Nano використовуючи бібліотеки для керування датчиком полум'я та GSM модулем.

Згідно з розробленого алгоритму, програма повинна проводити ініціалізацію модулів комплексу та потім зчитувати дані з модуля інфрачервоного датчику полум'я виконуючи надсилання даних через GSM модуль сповіщення про пожежу з подальшим інформуванням користувача через додаток.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ. АНАЛІЗ РЕЗУЛЬТАТІВ ОБЧИСЛЕНЬ

4.1 Створення акаунту у ThingSpeak та налаштування системи

Для створення прототипу комплексу та тестування системи виявлення пожежі та надсилання сповіщення було обрано онлайн платформу ThingSpeak [21]. ThingSpeak™ — це служба аналітичної платформи IoT, яка дозволяє агрегувати, візуалізувати та аналізувати живі потоки даних у хмарі. ThingSpeak забезпечує миттєвий перегляд даних, опублікованих вашими пристроями в ThingSpeak. Завдяки можливості запуску коду MATLAB® на ThingSpeak ви можете виконувати онлайн-аналіз та обробку даних, щойно вони надходять. ThingSpeak часто використовується для створення прототипів і підтвердження концепції систем IoT, які потребують аналізу (Див. рис. 4.1).

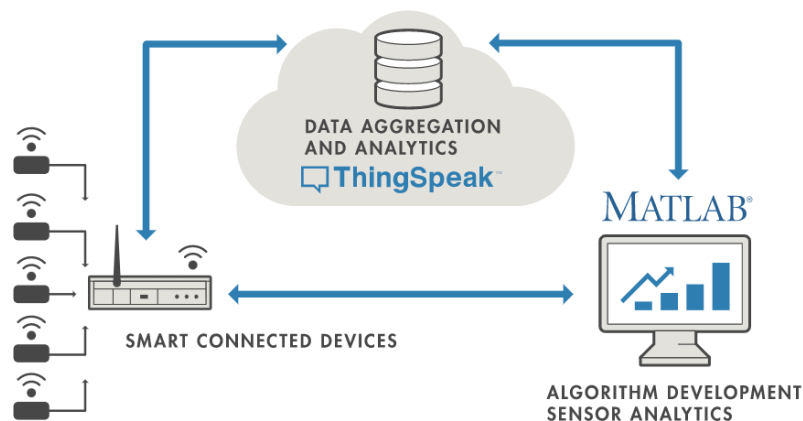


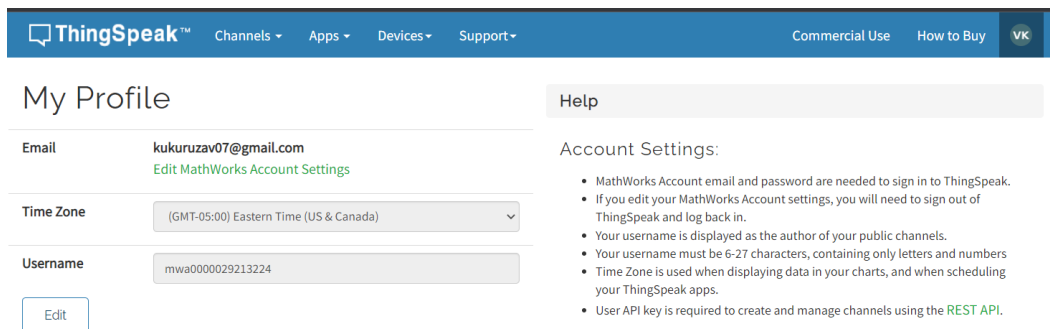
Рис. 4.1 – Середовище Thingspeak

ThingSpeak дозволяє агрегувати, візуалізувати та аналізувати живі потоки даних у хмарі. Деякі з ключових можливостей ThingSpeak включають можливість:

- Легко налаштовувати пристрої для надсилання даних на ThingSpeak за допомогою популярних протоколів IoT.
- Переглядати дані датчика в реальному часі.
- Додавати дані за запитом зі сторонніх джерел.

- Використовувати можливості MATLAB, щоб зрозуміти дані IoT.
- Виконувати свою аналітику IoT автоматично на основі розкладів або подій.
- Прототипувати та створювати системи IoT без налаштування серверів чи розробки веб-програмного забезпечення.
- Автоматично діяти на основі даних і спілкуватися за допомогою сторонніх служб, таких як Twilio® або Twitter®.

Спочатку на сайті <https://thingspeak.com/> було створено новий аккаунт Mathworks (Див. рис. 4.2).



Наступним кроком увійшовши у ThingSpeak за допомогою своїх облікових даних і натиснуто «Новий канал». Тепер було заповнено такі деталі проекту, як ім'я, назви полів. Потім натиснуто «Зберегти канал» (Див. Рис. 4.3).

The screenshot shows the 'New Channel' page on the ThingSpeak website. The header includes the ThingSpeak logo and navigation links for Channels, Apps, Devices, and Support. The main content area is divided into two columns: 'New Channel' and 'Help'.

New Channel Form:

- Name:** Arduino Fire Detection System With The Notification O
- Description:** The system was created to detect the fire and to notify the user and firefighters to start putting out the fire.
- Fields:** Eight fields are listed, each with a label and a checkbox. Field 1 is labeled 'Field Label 1' and is checked. Fields 2 through 8 are currently empty and unchecked.
- Metadata:** A text input field for additional information.

Help Section:

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings:

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
 - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.

Рис. 4.3 – Створення проекту в акаунті

В результаті було створено сторінку проекту апаратно-програмного комплексу для виявлення пожежі для прототипування та тестування роботи системи (Див. рис. 4.4).

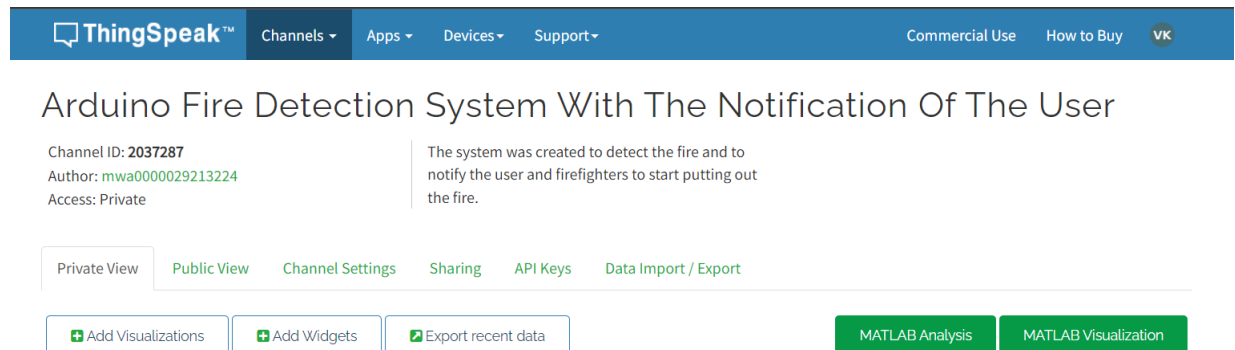


Рис. 4.4 – Сторінка системи виявлення пожежі

Наступним кроком було створено віджет для перевірки працездатності комплексу, коли вогонь буде виявлено датчиком полум'я та передано дані на сервіс то лампочка загориться зеленим кольором (Див. рис. 4.5).

Configure widget parameters ? x

Name

Condition If

v

v

turn Lamp ON

Update Interval second(s)

Color

Див. рис. 4.5 - Налаштування віджету для тесту працездатності

4.2 Налаштування мобільного додатку для отримання сповіщення від пристрою.

Для отримання сповіщень користувачем про початок пожежі було обрано додаток Pushsafer. Pushsafer — це служба для надсилання та отримання миттєвих push-сповіщень на телефоні, планшеті чи робочому столі (iOS, Android, Windows 10, Chrome, Firefox) із різних джерел а саме

проектів створених в Arduino. Через шлюз електронної пошти або через API Pushsafer можна інтегрувати практично в будь-яку програму [22].

Для тестування отримання сповіщень було обрано мобільний додаток для оперування на смартфоні Iphone 11 (Див. рис. 4.6).

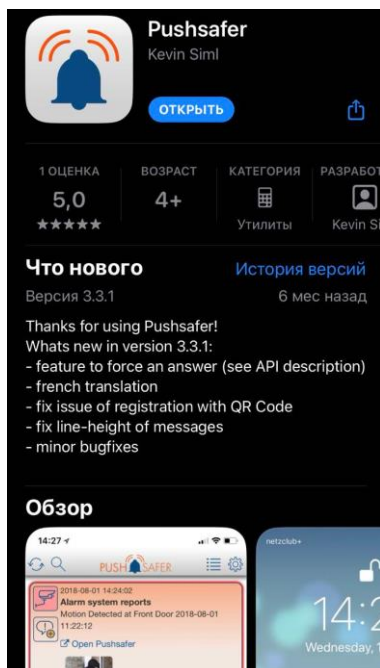


Рис. 4.6 – Мобільний додаток Pushsafer для IOS

Наступним кроком у даному додатку було створено аккаунт для подальшого отримання сповіщення з розробленого комплексу (Див. рис. 4.7).

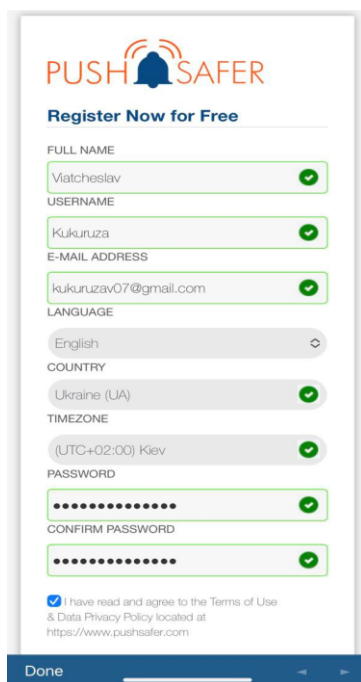


Рис. 4.7 – Регістрація у додатку Pushsafer

В результаті було успішно створено аккаунт у додатку та додано розроблений комплекс до списку девайсів мобільного додатку та проінформовано що пристрій здатний отримувати сповіщення про початок пожежі через додаток (Див. рис. 4.8). ID доданого девайсу – 63376.



1/1

Рис. 4.8 – Доданий апаратно програмний комплекс

З'єднання апаратного-програмного комплексу з мобільним додатком Pushsafer відбувається наступним чином [23].

```
#include <SPI.h>
#include <Ethernet.h>

byte mac[] = {0xDE,0xAC,0xBF,0xEF,0xFE,0xAA};

String privatekey = "Fire Detection system";

int length;

EthernetClient client;

void setup()
{
  Serial.begin(9600);
  Serial.print(F("Starting ethernet..."));
  if(!Ethernet.begin(win)) Serial.println("failed");
```

```
else Serial.println(Ethernet.localIP());

delay(5000);
Serial.println("Ready");
}

void loop()
{
pushsafer("Fire was detected!", "Fire detection
system", "1", "1", "1", "#FF0000", "https://www.pushsafer.com/dashboard", "wPP6gykYx9ZxzQYc
utKS ", "0", "2", "60", "6000", "Successful detection", "63376");
delay(60000);
}

byte pushsafer(char *pushsafermessage, char *pushsafertitle, char *pssound, char
*psvibration, char *psicon, char *pscolor, char *psurl, char *psurlltitle, char
*ptime2live, char *pspriority, char *psretry, char *psexpire, char *psanswer, char
*psdevice)
{
String title = pushsafertitle;
String message = pushsafermessage;
String device = psdevice;
String sound = pssound;
String vibration = psvibration;
String icon = psicon;
String color = pscolor;
String url = psurl;
String urltitle = psurlltitle;
String time2live = ptime2live;
String priority = pspriority;
String retry = psretry;
String expire = psexpire;
String answer = psanswer;

length = 45 + message.length() + title.length() + sound.length() +
vibration.length() + icon.length() + color.length() + url.length() +
urltitle.length() + time2live.length() + device.length() + privatekey.length();

if(client.connect("pushsafer.com",80))
{
```

```
client.println("POST /api HTTP/1.1");
client.println("Host: https://www.pushsafer.com");
client.println("Connection:      close\r\nContent-Type:      application/x-www-form-
urlencoded");
client.print("Content-Length: ");
client.print(length);
client.println("\r\n");
client.print("k=");
client.print(privatekey);
client.print("&m=");
client.print(message);
client.print("&t=");
client.print(title);
client.print("&s=");
client.print(sound);
client.print("&v=");
client.print(vibration);
client.print("&i=");
client.print(icon);
client.print("&c=");
client.print(color);
client.print("&u=");
client.print(url);
client.print("&ut=");
client.print(urlltitle);
client.print("&l=");
client.print(time2live);
client.print("&pr=");
client.print(priority);
client.print("&re=");
client.print(retry);
client.print("&ex=");
client.print(expire);
client.print("&a=");
client.print(answer);
client.print("&d=");
client.print(device);
while(client.connected())
{
    while(client.available())
```

```
{
  char ch = client.read();
  Serial.write(ch);
}
}
client.stop();
}
}
```

4.3 Перевірка працездатності

Було повністю зібрано та з'єднано усі компонуючі апаратно-програмного комплексу (Див. Рис. 4.9). Після компіляції коду у програмній частині та завантаженні оновленого коду до апаратної платформи, було перевірено працездатність всього комплексу. Була проведена імітацію пожежі, після чого датчик вогню розпізнав її.

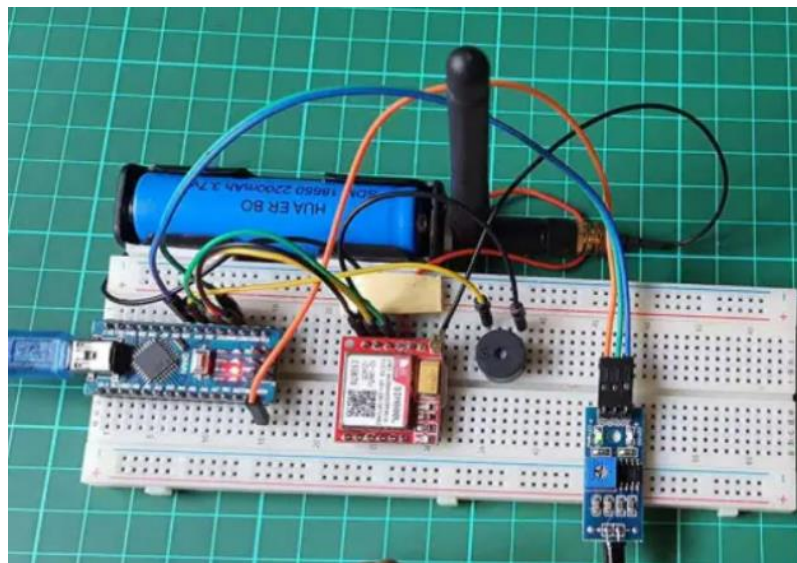


Рис. 4.9 – Апаратно-програмний комплекс

Після того як програму запущено та система розпізнала вогонь або так звану імітовану пожежу то в онлайн середовищі ThingSpeak у віджеті який відповідає за детекцію пожежі загорілася зелена лампа, як і планувалося тому роботу пристрою можна рахувати успішною (Рис. 4.10).

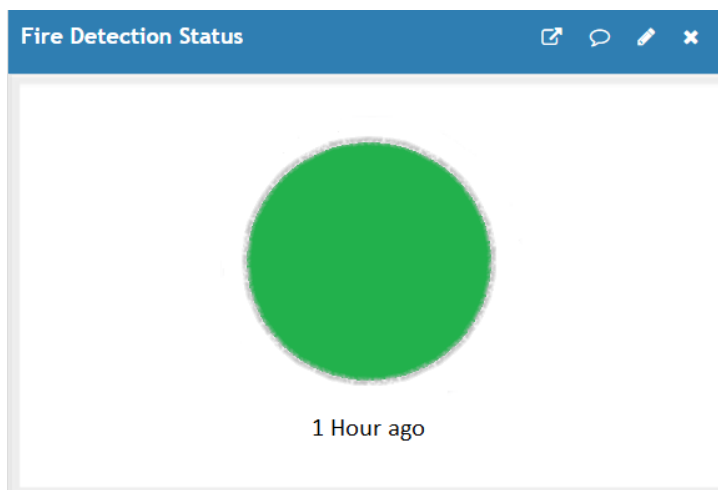


Рис. 4.10 – Тестування системи у ThingSpeak

Відразу після того як було виявлено пожежу в мобільний додаток Pushsafer на Iphone 11 прийшло сповіщення про виявлену пожежу. На рисунку зображено скріншот екрану з отриманим сповіщенням що 18 лютого 2023 року в будинку було виявлено пожежу, як і очукувалося (Див. Рис. 4.11).

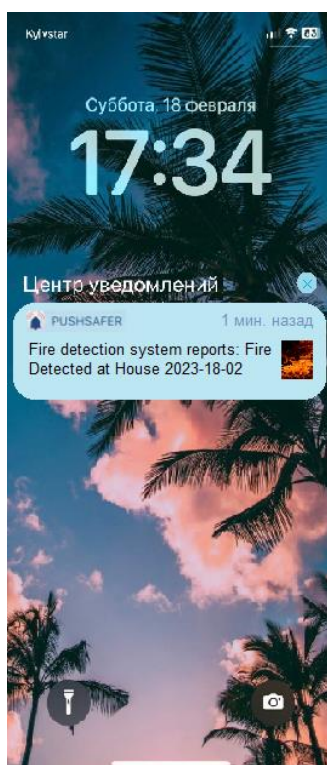


Рис. 4.11 – Сповіщення на мобільний через Pushsafer

Заходячи у мобільний додаток надалі є повідомлення що пожежу було виявлено двічі на день – у 17:33 під час першого тесту та у 18:58 під час

повторного тестування та обидва сповіщення було успішно отримані (Див. рис. 4.12).

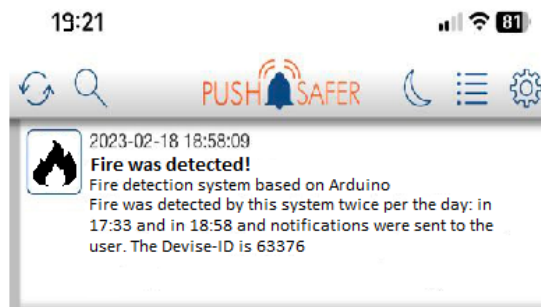


Рис. 4.12 – Повідомлення про детектування пожежі

Далі було проведено експерименти з різними розташуваннями джерела вогню в зоні діапазону дії інфрачервоного датчику вогню і проведено загалом 10 таких тестувань. Результати можна побачити на графіку нижче (Див. рис. 4.13). В проміжку виявлення мається на увазі час як розпізнавання пожежі так відсилання сповіщення через мобільний додаток, тобто загалом [24].



Рис. 4.13 – Тестування системи

4.4 Порівняння прототипу з існуючими рішеннями

Точність датчика полум'я в приміщенні залежить від різних факторів, таких як тип і якість датчика, розміщення та конфігурація датчика, а також умови навколишнього середовища в приміщенні.

Датчики полум'я зазвичай працюють, виявляючи інфрачервоне випромінювання, яке випромінює полум'я, і можуть викликати тривогу, коли виявлено полум'я. Вони зазвичай використовуються в таких приміщеннях, як підвали, де пожежі можуть залишатися непоміченими протягом тривалого періоду часу, і можуть забезпечити раннє попередження про пожежу, перш ніж вона стане занадто великою або пошириться на інші частини будівлі [25].

Щоб забезпечити оптимальну точність, датчик полум'я слід розташувати в місці, де він зможе швидко та надійно виявляти полум'я. Датчик має бути встановлено таким чином, щоб мінімізувати ризик створення перешкод або перешкод, а чутливість датчика має бути налаштована для мінімізації помилкових тривог, забезпечуючи при цьому надійне виявлення полум'я.

Також важливо враховувати умови навколишнього середовища в підвалі, такі як температура, вологість і наявність пилу чи інших частинок, які можуть вплинути на точність датчика. Регулярне технічне обслуговування та перевірка датчика полум'я може допомогти переконатися, що він функціонує належним чином і забезпечує точне виявлення пожежі.

Існує багато типів датчиків які використовуються для виявлення пожежі та у таблиці 4.1 нижче наведено їх характеристики та порівняння.

Таблиця 4.1 Характеристики та порівняння датчиків для виявлення пожежі

Тип датчика	Спосіб виявлення	Переваги	Недоліки
Датчик диму	Оптичний або іонізаційний	Раннє виявлення повільно горючих пожеж	Може бути схильний до помилкових тривог від приготування їжі або пари
Датчик тепла	Підвищення температури	Надійне виявлення швидкогорючих пожеж	Повільно для виявлення повільногорючих пожеж
Датчик вогню	Інфрачервоне або	Швидке та надійне	Обмежений діапазон

	ультрафіолетове випромінювання	виявлення полум'я	виявлення
Датчик газу	Виявлення специфічних газів, що утворюються при пожежі	Може виявляти пожежі в місцях з низьким рівнем кисню	Може бути схильний до помилкових тривог від інших джерел газу
Датчик чадного газу	Виявлення чадного газу, що утворюється під час пожеж	Може виявляти пожежі в місцях з низьким рівнем кисню	Може бути схильний до помилкових тривог від інших джерел чадного газу

З порівняння можна зробити висновок що саме датчик вогню для розпізнавання пожежі був обраний правильно тому що це найефективніший метод, недоліки інших датчиків є значними, а недолік датчика вогню може бути вирішений встановленням другої системи виявлення пожежі для розширення діапазону дії детекції полум'я або встановленням датчику у менших за площею приміщеннях або місцях де пожежна безпека має бути на найвищому рівні.

При виборі відповідного типу датчика важливо враховувати конкретне застосування та середовище, в якому буде використовуватися датчик. У деяких випадках для оптимального виявлення та запобігання пожежі може використовуватися комбінація датчиків. Регулярне технічне обслуговування та перевірка датчиків також може допомогти забезпечити їхнє належне функціонування та надійне виявлення пожежі при різних ступінях її розгоряння (Див. рис. 4.14).

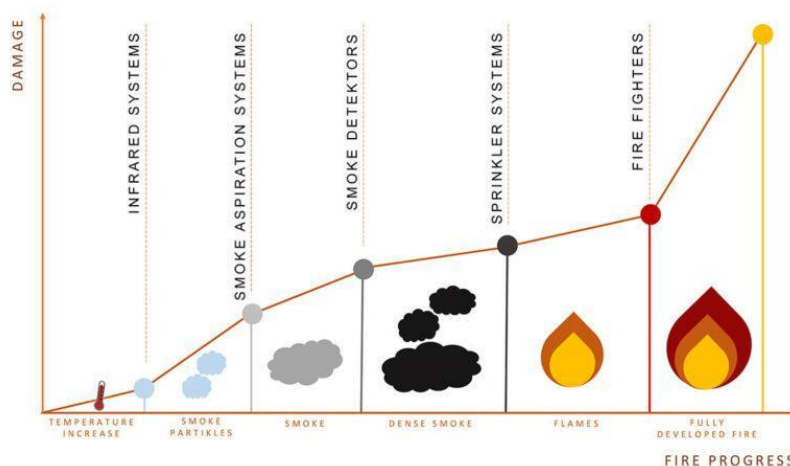


Рис. 4.14 – Прогрес розгоряння пожежі

Як результат створений апаратно-програмний комплекс має змогу достатньо швидко та ефективно виявляти пожежу та повідомляти про це користувача або екстренні служби на встановленому приміщенні, зони яка має бути пожежно безпечною. Недоліки даного комплексу можна вирішити правильним розміщенням саме датчику вогню для точного розпізнавання, додаванням ще одного аналогічного комплексу у іншу частину будівлі якщо це велике приміщення або навіть вдосконалювати пристрій поєднуючи його з іншими типами датчиків які можуть бути використанні для виявлення пожежі. Головне для підбору даних датчиків правильно їх розмістити зважаючи на особливості приміщення, та саме конкретне середовище де будуть використовуватись датчики розроблюємого комплексу.

Висновки до розділу 4

На початку четвертого розділу була виконана реєстрація та налаштування у онлайн середовищі ThingSpeak та налаштовано однойменний сервіс IoT. ThingSpeak™ — це служба аналітичної платформи IoT, яка дозволяє агрегувати, візуалізувати та аналізувати живі потоки даних у хмарі. На цій платформі було створено новий канал та додано проект апаратно-програмного комплексу для виявлення пожежі та створено віджет необхідний для тестування системи який покаже що комплекс працює правильно так виконує поставлену мету.

Наступним кроком було обрано та встановлено мобільний додаток PushSafer для швидкого надсилання сповіщення на телефон. Pushsafer — це служба для надсилання та отримання миттєвих push-сповіщень на телефоні, планшеті чи робочому столі (iOS, Android, Windows 10, Chrome, Firefox) із різних джерел а саме проектів створених в Arduino. Було створено акаунт та додано розроблений проект до нього. Потім програмну частину було інтегровано з проектом в мобільному додатку для отримання сповіщення.

Зібраний апаратно-програмний комплекс було протестовано при допомозі імітації пожежі а саме розташуванням джерела вогню біля датчику

полум'я і як результат віджет який мав показати правильну роботу системи загорівся зеленим кольором коли пожежу було виявлено, як і задумано. Одразу ж на телефон де було встановлено мобільний додаток Pushsafer прийшло сповіщення про початок пожежі і пізніше після повторного тестування комплексу в додатку було вказано що комплекс виявив пожежу двічі, показано коли було виявлено детектування та отримані сповіщення для надання інформації про пожежу. Було проведено експерименти з різними розташуваннями джерела полум'я в діапазоні дії датчика полум'я та надано графік результатів роботи системи з часом виявлення пожежі та надсиланням сповіщення через мобільний додаток.

Також було порівняно прототип системи зі схожими рішеннями та наведено їх характеристики, обґрунтовано саме вибір таких компонентів для виконання поставленою мети.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи магістра було проаналізовано матеріал щодо можливих технологій та методів виявлення пожежі, оглянуто існуючі рішення для виявлення пожежі та проаналізовано можливі апаратні платформи для реалізації прототипу комплексу. Було опановано та проаналізовано матеріал щодо методів та принципів роботи датчиків полум'я та GSM модулів а також їх різновидностей датчиків полум'я та принципів їх роботи.

Далі було наведено характеристику для обраної апаратної платформи а саме мікроконтролеру Arduino Nano. Далі було детально описано характеристики обраної платформи та модулів які були використані під час проектування та конструювання прототипу. Програмний додаток було розроблено на мові програмування C у середовищі розробки Arduino Nano використовуючи бібліотеки для керування датчиком полум'я та GSM модулем.

Згідно з розробленого алгоритму, програма повинна проводити ініціалізацію модулів комплексу та потім зчитувати дані з модуля інфрачервоного датчику полум'я виконуючи надсилання даних через GSM модуль сповіщення про пожежу з подальшим інформуванням користувача через додаток. Також було спроектовано макетну схему апаратної частини та створено декілька діаграм для кращого пояснення принципу роботи комплексу.

Було створено програмну частину та обрано IoT сервіс для тестування правильної роботи системи та обрано мобільний додаток завдяки якому було отримано сповіщення на телефон про початок пожежі. Апаратно-програмний комплекс було успішно протестовано і перевірено правильність детектування пожежі у платформі ThingSpeak та через створений аккаунт у додатку Pushsafer для IOS успішно отримано сповіщення про початок пожежі, так само і при повторних тестуваннях системи, результат позитивний.

Було проведено експерименти які показали трохи різний час виявлення пожежі та надсилання сповіщення через розміщення джерела імітації пожежі. Також було порівняно розроблений прототип зі схожими існуючими рішеннями та надано їх характеристики і обґрунтовано перевагу розробленого комплексу щодо схожих рішень.

Як результат створений апаратно-програмний комплекс має змогу достатньо швидко та ефективно виявляти пожежу та повідомляти про це користувача або екстренні служби на встановленому приміщенні, зони яка

має бути пожежно безпечною. Недоліки даного комплексу можна вирішити правильним розміщенням саме датчику вогню для точного розпізнавання, додаванням ще одного аналогічного комплексу у іншу частину будівлі якщо це велике приміщення або навіть вдосконалювати пристрій поєднуючи його з іншими типами датчиків які можуть бути використанні для виявлення пожежі.

Результати виконання кваліфікаційної роботи магістра можуть бути рекомендовані для використання у сфері пожежної безпеки, а саме наприклад для невеликих приміщень, офісів, гаражів, частних маєтків або інших різноманітних приміщеннях для швидкого надання інформації про початок пожежі через додаток користувачеві.

В процесі виконання також було виконано частину з охорони праці. Були проаналізовані умови в приміщенні підприємства в якому проводились дослідження. Було запропоновано доцільні заходи щодо його покращення описані у розділі охорони праці.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кукуруза В. І., Савінов В. Ю. Апаратно-програмний комплекс розпізнавання пожежі на базі Arduino та сповіщення користувача через мобільний застосунок. Інформаційні технології та інженерія : тези доп. Всеукр. наук.-практ. конф. Миколаїв, 07–10 лют. 2023 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2023. С. 76–78.
2. Fire Safety Systems. URL: <https://www.csuohio.edu/access-security/fire-safety-systems> (дата звернення: 14.12.2022).
3. Плата NANO Arduino. URL: https://uk.wikipedia.org/wiki/Arduino_Nano. (дата звернення: 14.12.2022).
4. Fire Alarm System. URL: https://en.wikipedia.org/wiki/Fire_alarm_system (дата звернення: 15.12.2022).
5. Пожежна безпека об'єктів будівництва <https://wiki.legalaid.gov.ua/> (дата звернення: 15.12.2022).
6. Fire Detection control panel; URL: <https://www.ddfire.gov.uk/fire-detection-and-warning-system>. (дата звернення: 15.12.2022).
7. Arduino IoT Cloud; URL: <https://docs.arduino.cc/arduino-cloud/> (Дата звернення: 14.01.2023).
8. Waldemar Rebizant , Janusz Szafran , Andrzej Wiszniewski. Digital Signal Processing in Fire System Protection and Control. Springer, Berlin. 2011. №3:622.
9. МОНК С. Arduino programming. Professional work with sketches. СПб. 2017.
10. Thampi, S.M., Krishnan, S., Corchado, J.M., Das, S., Wozniak, M., Al-Jumeily, D. Advances in Signal Processing and Intelligent Recognition Systems. SIRS. 2017.
11. Getting Started with Arduino products: <https://www.arduino.cc/en/Guide> (дата звернення: 16.01.2023).

12. Arduino Integrated Development Environment:
<https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics> (дата звернення: 19.01.2023).

13. SIM800L GSM module characteristics:
<https://nettigo.eu/products/sim800l-gsm-grps-module> (дата звернення: 24.01.2023).

14. Fire detection: <https://www.sciencedirect.com/topics/engineering/fire-detection> (дата звернення: 27.01.2023).

15. Fire safety through mathematical modelling:
https://www.researchgate.net/publication/228763095_Fire_safety_through_mathematical_modelling (дата звернення: 01.02.2023).

16. Flame Sensor: Principle of work:
<https://www.watelectronics.com/flame-sensor/> (дата звернення: 02.02.2023).

17. Infrared Flame Sensor Module, how it works:
<https://www.electroduino.com/ir-infrared-flame-sensor-module/> (дата звернення: 02.02.2023).

18. Push notifications using Pushsafer:
<https://projecthub.arduino.cc/pushsafer/acd03914-b45f-40a3-93b6-73c0d7ba5ae0> (дата звернення - 05.02.2023).

19. Connection of Arduino project with Pushsafer app
<https://www.pushsafer.com/en/arduino> (дата звернення - 07.02.2023).

20. What is GSM means:
<https://www.techtarget.com/searchmobilecomputing/definition/GSM> (дата звернення – 09.02.2023).

21. Learn More About ThingSpeak:
https://thingspeak.com/pages/learn_more (дата звернення: 11.02.2023).

22. Li-Ion battery 3.6V - 3.7V with 3200mAh capacity:
<https://www.akkuteile.de/en/lithium-ionen-battery/size-26500/26650a-li-ion->

battery-3-6v-3-7v-with-4500mah-capacity-and-15a-discharge-current_100733_3027 (дата звернення - 13.02.2023).

23. Different Types of Fire Detector sensors: <https://firerisk.co.uk/different-types-of-fire-detector-heads/> (дата звернення – 15.02.2023).

24. Arduino Programming in C: <https://www.codementor.io/@dewetvanthomas/tutorial-arduino-programming-in-c-12b7czyui> (Дата звернення - 16.02.2023).

25. UML Diagrams: Types: <https://tallyfy.com/uml-diagram/> (дата звернення: 17.02.2023).

ДОДАТОК А. КОД ПРОГРАМИ

Лістинг коду програми:

```
#include <SoftwareSerial.h>

SoftwareSerial gprsSerial(10, 11);

#include <String.h>

#include <SPI.h>
#include <Ethernet.h>

byte mac[] = {0xDE,0xAC,0xBF,0xEF,0xFE,0xAA};

String privatekey = "Fire Detection system";

int length;

EthernetClient client;

int flag = 0;

void setup()

{

    pinMode(9, OUTPUT);

    pinMode(12, INPUT);

    gprsSerial.begin(9600);           // the GPRS baud rate

    Serial.begin(9600);   // the GPRS baud rate

    Module_Init();

}

void loop()
```

```
{

    if (gprsSerial.available())

        Serial.write(gprsSerial.read());

    int fire = digitalRead(12);

    if (fire == 0)

    {

        digitalWrite(9, HIGH);

    }

    void setup()
    {
        Serial.begin(9600);
        Serial.print(F("Starting ethernet..."));
        if(!Ethernet.begin(win)) Serial.println("failed");
        else Serial.println(Ethernet.localIP());

        delay(5000);
        Serial.println("Ready");
    }

    void loop()
    {
        pushsafer("Fire was detected!","Fire detection
system","1","1","1","#FF0000","https://www.pushsafer.com/dashboard","wPP6gykYx9ZxzQYc
utKS ","0","2","60","6000","Successful detection","63376");
        delay(60000);
    }

    byte pushsafer(char *pushsafermessage, char *pushsafertitle, char *pssound,
char *psvibration, char *psicon, char *pscolor, char *psurl, char *psurltitle, char
*psptime2live, char *pspriority, char *psretry, char *psexpire, char *psanswer, char
*psdevice)
    {
        String title = pushsafertitle;
        String message = pushsafermessage;
        String device = psdevice;
```

```
String sound = pssound;
String vibration = psvibration;
String icon = psicon;
String color = pscolor;
String url = psurl;
String urltitle = psurltitle;
String time2live = pstime2live;
String priority = pspriority;
String retry = psretry;
String expire = psexpire;
String answer = psanswer;

length = 45 + message.length() + title.length() + sound.length() +
vibration.length() + icon.length() + color.length() + url.length() +
urltitle.length() + time2live.length() + device.length() + privatekey.length();

if(client.connect("pushsafer.com",80))
{
    client.println("POST /api HTTP/1.1");
    client.println("Host: https://www.pushsafer.com");
    client.println("Connection: close\r\nContent-Type: application/x-www-form-
urlencoded");
    client.print("Content-Length: ");
    client.print(length);
    client.println("\r\n");;
    client.print("k=");
    client.print(privatekey);
    client.print("&m=");
    client.print(message);
    client.print("&t=");
    client.print(title);
    client.print("&s=");
    client.print(sound);
    client.print("&v=");
    client.print(vibration);
    client.print("&i=");
    client.print(icon);
    client.print("&c=");
    client.print(color);
    client.print("&u=");
    client.print(url);
```

```

client.print("&ut=");
client.print(urltitle);
client.print("&l=");
client.print(time2live);
client.print("&pr=");
client.print(priority);
client.print("&re=");
client.print(retry);
client.print("&ex=");
client.print(expire);
client.print("&a=");
client.print(answer);
client.print("&d=");
client.print(device);
while(client.connected())
{
  while(client.available())
  {
    char ch = client.read();
    Serial.write(ch);
  }
}
client.stop();
}
}

```

```

gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"); //start up
the connection

```

```

delay(6000);

```

```

ShowSerialData();

```

```

gprsSerial.println("AT+CIPSEND");//begin send data to remote server

```

```

delay(4000);

```

```

ShowSerialData();

```

```

String str = "GET
https://api.thingspeak.com/update?api_key=ER43PXXXXXHQF0I&field1=" + String(1);

```

```
Serial.println(str);

gprsSerial.println(str);//begin send data to remote server

delay(4000);

ShowSerialData();

digitalWrite(9, LOW);

gprsSerial.println((char)26);//sending

delay(5000);//waitting for reply, important! the time is base on the
condition of internet

gprsSerial.println();

ShowSerialData();

gprsSerial.println("AT+CIPSHUT");//close the connection

delay(100);

ShowSerialData();

flag = 0;

}

else

{

digitalWrite(9, LOW);

if (flag == 0)

{

flag = 1;
```

```
gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"); //start up
the connection

    delay(6000);

    ShowSerialData();

    gprsSerial.println("AT+CIPSEND"); //begin send data to remote server

    delay(4000);

    ShowSerialData();

    String str = "GET
https://api.thingspeak.com/update?api_key=ER43PWT91CGHQF0I&field1=" + String(0);

    Serial.println(str);

    gprsSerial.println(str); //begin send data to remote server

    delay(4000);

    ShowSerialData();

    digitalWrite(9, LOW);

    gprsSerial.println((char)26); //sending

    delay(5000); //waitting for reply, important! the time is base on the
condition of internet

    gprsSerial.println();

    ShowSerialData();

    gprsSerial.println("AT+CIPSHUT"); //close the connection

    delay(100);
```

```
        ShowSerialData();

    }

}

void ShowSerialData()

{

    while (gprsSerial.available() != 0)

        Serial.write(gprsSerial.read());

    delay(5000);

}

void Module_Init()

{

    gprsSerial.println("AT");

    delay(1000);

    gprsSerial.println("AT+CPIN?");

    delay(1000);

    gprsSerial.println("AT+CREG?");

    delay(1000);

    gprsSerial.println("AT+CGATT?");

    delay(1000);

    gprsSerial.println("AT+CIPSHUT");
```



```
    delay(1000);

    gprsSerial.println("AT+CIPSTATUS");

    delay(2000);

    gprsSerial.println("AT+CIPMUX=0");

    delay(2000);

    ShowSerialData();

    gprsSerial.println("AT+CSTT=\"www\"");

    delay(1000);

    ShowSerialData();

    gprsSerial.println("AT+CIICR");

    delay(3000);

    ShowSerialData();

    gprsSerial.println("AT+CIFSR");

    delay(2000);

    ShowSerialData();

    gprsSerial.println("AT+CIPSPRT=0");

    delay(3000);

    ShowSerialData();

}
```