

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ПОЖЕЖНОЇ БЕЗПЕКИ
ДЛЯ СИСТЕМИ РОЗУМНИЙ БУДИНОК
SOFTWARE IMPLEMENTATION OF THE FIRE SAFETY SUBSYSTEM FOR
THE SMART HOUSE SYSTEM

Виконала: студентка 4 курсу, групи КНТ-210

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ЧУБКО Н. О.

(ПРИЗВИЩЕ та ініціали)

Керівник ПАРХОМЕНКО А.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І. Я.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет комп'ютерних наук та технологій
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЧУБКО Наталі Олександрівни

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація підсистеми пожежної безпеки для системи розумний будинок. Software Implementation of the Fire Safety Subsystem for the Smart House System.

керівник проєкту (роботи) к.т.н., доцент, ПАРХОМЕНКО Анжеліка Володимирівна

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “24” квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування програмного забезпечення. 3. Конструювання підсистеми протипожежної безпеки. 4. Інструкція щодо використання підсистеми протипожежної безпеки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ПАРХОМЕНКО А.В., доцент		
Нормоконтроль	БЄЛОВА А.В., асистент		

7. Дата видачі завдання « 15 » квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Наталя ЧУБКО
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ Анжеліка ПАРХОМЕНКО
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 63 с., 12 табл., 39 рис., 2 дод., 36 джерел.

РОЗУМНИЙ БУДИНОК, ПРОТИПОЖЕЖНА ПІДСИСТЕМА, ДАТЧИКИ, МОНІТОРИНГ, ARDUINO, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – моніторинг і раннє виявлення пожеж в системі "розумний" будинок.

Предмет дослідження – програмно-апаратні засоби моніторингу і раннього виявлення пожеж.

Мета роботи – розробка програмно-апаратного забезпечення підсистеми пожежної безпеки для моніторингу і раннього виявлення пожеж в системі "розумний" будинок.

Матеріали, методи та технічні засоби: Arduino UNO, датчик газу MQ2, датчик температури LM35, LCD дисплей, датчик полум'я, GSM модем, зумер, мова програмування C++, середовище програмування Arduino, персональний комп'ютер з процесором Intel Core 2 Duo під управлінням операційної системи Microsoft Windows 10, Proteus 8 Professional.

Результати. Створено прототип підсистеми пожежної безпеки, що включає датчики диму, температури та газу, зв'язані з центральним модулем на базі Arduino, а також програмне забезпечення моніторингу та раннього виявлення пожеж.

Висновки. Використання програмно-апаратного комплексу дозволяє створити доступну та просту у використанні підсистему пожежної безпеки, яка може бути реалізована навіть з обмеженим бюджетом.

Галузь використання – моніторинг та раннє виявлення пожеж в різних приміщеннях.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 63 pages, 12 tables, 39 figures, 2 appendixes, 36 sources.

SMART HOME, FIRE SUBSYSTEM, SENSORS, MONITORING, ARDUINO, SOFTWARE.

Object of study is monitoring and early detection of fires in the Smart House system.

Subject of study is software/hardware tools for monitoring and early detection of fires.

Purpose of work is development of software and hardware of the fire safety subsystem for monitoring and early detection of fires in the Smart House system.

Materials, methods and tools: Arduino UNO, MQ2 gas sensor, LM35 temperature sensor, LCD display, flame sensor, GSM modem, buzzer, C++ programming language, Arduino programming environment, personal computer with Intel Core 2 Duo processor running Microsoft Windows 10 operating system, Proteus 8 Professional.

Results. A prototype of a fire safety subsystem has been created, including smoke, temperature and gas sensors connected to a central module based on Arduino, as well as software for monitoring and early detection of fires.

Conclusions. The usage of the hardware/software complex allows creating an affordable and easy-to-use fire safety subsystem that can be implemented even with a limited budget.

The field of application is monitoring and early detection of fires in various premises.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Аналіз предметної області.....	9
1.1 Задачі моніторингу пожежної безпеки.....	9
1.2 Огляд існуючих апаратних рішень.....	10
1.3 Огляд існуючих програмно-апаратних рішень	14
1.4 Постановка завдань дипломної роботи.....	17
2 Проєктування програмного забезпечення	18
2.1 Аналіз вимог	18
2.2 Розробка архітектури системи	18
2.3 Вибір мови та середовища програмування	20
2.4 Вибір апаратних рішень	22
2.5 Висновки до розділу 2	29
3 Конструювання підсистеми протипожежної безпеки	30
3.1 Алгоритм роботи підсистеми.....	30
3.2 Опис апаратної реалізації підсистеми.....	32
3.3 Опис програмної реалізації підсистеми.....	33
3.4 Висновки до розділу 3	38
4 Інструкція щодо використання підсистеми протипожежної безпеки	39
Висновки	46
Перелік джерел посилань	47
Додаток А Текст програми.....	51
Додаток Б Слайди презентації	56

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

BIS	– Building Integration System;
IDE	– Integrated Development Environment;
LCD	– Liquid-Crystal Display;
ПБ	– пожежна безпека;
ППБ	– підсистема пожежної безпеки;
СРБ	– система “розумний” будинок;
СДА	– система домашньої автоматизації.

ВСТУП

За останні десятиліття впровадження технологій змінило і розширило наше повсякденне життя, що відкрило нові можливості для забезпечення комфорту та безпеки. У сфері будівництва та житлової інфраструктури концепція “розумного” будинку стала однією з основних тенденцій. Сучасні системи “розумний” будинок (СРБ) мають багато функцій, спрямованих на те, щоб допомогти автоматизувати та оптимізувати різні аспекти життя.

Тим не менш, із зростанням складності таких систем зростає потреба в забезпеченні безпеки, особливо в разі пожежі, найбільшої загрози для будь-якого будинку. З цієї причини дуже важливо створити підсистему пожежної безпеки (ППБ) для СРБ, щоб надійно та ефективно захистити житло та майно від пожежі.

Тому, актуальним є вивчення існуючих підходів до пожежної безпеки, створення та реалізація алгоритмів і протоколів для вчасного виявлення та локалізації пожежі, а також розробка методів автоматичного оповіщення та взаємодії з іншими системами РБ, щоб зменшити ризик і максимізувати безпеку в житлових зонах.

Мета роботи – розробка програмно-апаратного забезпечення ППБ для моніторингу і раннього виявлення пожеж в системі РБ.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Задачі моніторингу пожежної безпеки

У 2023 році в Україні зареєстровано 67 934 пожежі. 1 472 людини загинули в результаті пожеж, у тому числі 40 дітей. В середньому щодня в Україні виникало 186 пожеж, у результаті яких було завдано 226 мільйонів гривень матеріальних збитків. Кожного дня пожежі призводять до смерті чотирьох людей і пошкоджують 78 будівель [1].

Будинок є об'єктом, який є потенційно вразливим до пожежі. Під час пожежі в будинку безпека людей повинна бути головним пріоритетом. Необережність із полум'ям є найпоширенішою причиною пожеж. Найчастіше це через необережність під час паління та короткі замикання [1].

Отже, для того, щоб люди могли самостійно евакуюватися, мешканці будинків, які постраждали від стихійних лих, повинні отримати інформацію у вигляді раннього оповіщення про пожежу. І саме для цього використовуються автоматичні системи пожежної безпеки.

Системи домашньої автоматизації стали набагато популярнішими в останні роки. І з кожним роком це впливає на кількість пожеж. Порівняно з 2022 роком, кількість пожеж зменшилася на 10% [1]. Це пов'язано з тим, СДА дозволяють нам контролювати різні частини нашого житла, системи розумного будинку стають все більш поширеними. Розумні будинки забезпечують безпеку та захищають житло від будь-яких надзвичайних ситуацій. Стандартні функції РБ включають захист від злоумисників, автоматизацію дверей і воріт, охоронну сигналізацію та системи запобігання надзвичайним ситуаціям, запобігання пожежам і задимленню, запобігання витоку води та підйому рівня води. Нові датчики дозволять швидше та більш надійно виявити пожежі. Бездротові системи дозволять пожежникам розробляти стратегії гасіння пожеж ще до прибуття на місце пожежі, звільняючи від необхідності прокладання кабелів [2].

1.2 Огляд існуючих апаратних рішень

Огляд існуючих аналогів є надзвичайно важливим етапом у вивченні та впровадженні нових методів і технологій, оскільки тема ПБ залишається надзвичайно важливою через постійний розвиток технологій у цій галузі. Щоб зробити обґрунтовані висновки та визначити перспективні напрямки подальших досліджень, буде розглянуто датчики ПБ від різних компаній.

1.2.1 Датчик Google Nest Protect

Google Nest Protect - це інтелектуальний датчик диму та вуглекислого газу, розроблений компанією Google, щоб допомогти людям залишатися в безпеці в своїх будинках. Основною метою Nest Protect є попередження споживачів про потенційно небезпечні ситуації, пов'язані з пожежами або викидами газів, а також виявлення таких ситуацій [3]. Зовнішній вигляд пристрою представлено на рис. 1.1.

Переваги:

- виявлення диму, який має розділений спектр;
- сумісність з proSmart-пристроями;
- сповіщення proVoice.

Недоліки:

- висока ціна: від 2300 грн до 4500 грн;
- іноді можуть виникати ложні сигнали тривоги;
- деяким користувачам не вдалося приєднатися до застосунку [4].

1.2.2 FIBARO Smoke Sensor - FIBCFGSD-002

Датчик FIBARO Smoke Sensor FIBCFGSD-002 розроблений польською компанією FIBARO, яка спеціалізується на розробці різноманітних продуктів для РБ (рис.1.1).



Рисунок 1.1 – Датчик Google Nest Protect [5]

Невеликий датчик виконує свою основну мету – виявлення усіх потенційних загроз. Датчик виконує не тільки виявлення диму, а ще й вимірювання температури [6].

Переваги:

- має високу чутливість до диму;
- компактний;
- сповіщення на підключений смартфон;
- довго працює без заміни батареї.

Недоліки:

- висока ціна: від 2700грн до 5000 грн;
- може припинити роботу при відключенні від мережи Bluetooth або Z-Wave;
- потребує додатковий контролер;
- не завжди спрацьовує сигнал тривоги [7].

Зовнішній вигляд пристрою представлено на рис. 1.2.



Рисунок 1.2 – FIBARO Smoke Sensor - FIBEFGSD-002 (FIB_FGSS-101) [6]

1.2.3 Ajax FireProtect Plus

Датчик від української компанії Ajax Systems. Датчики пожежі, системи контролю доступу, системи відеоспостереження, датчики затоплення та інші пристрої є частиною бездротових систем безпеки Ajax Systems [8].

Переваги:

- має високу чутливість до диму;
- захист від підміни;
- сповіщення на підключений смартфон;
- працює без заміни батареї до 4 років.

Недоліки:

- для більших функцій треба окремо докупити спеціальну місячну підписку на систему Ajax;
- не завжди сумісний з іншими системами безпеки.

Зовнішній вигляд пристрою представлено на рис. 1.3.



Рисунок 1.3 – Ajax FireProtect Plus[9]

Порівняльну характеристику датчиків наведено у табл. 1.1.

Таблиця 1.1 – Порівняльна характеристика датчиків [3], [4], [6] - [8]

Критерій порівняння	Google Nest Protect	FIBARO Smoke Sensor - FIBCFGSD-002 (FIB_FGSS-101)	Ajax FireProtect Plus
1	2	3	4
Детектори	Дим, вуглекислий газ, вологість, температура	Дим, вуглекислий газ, підвищена температура	Дим, вуглекислий газ, підвищена температура
Бездротове підключення	+	+	+
Оповіщення на смартфон	+	+	+

Продовження таблиці 1.1

1	2	3	4
Зручність	-	+	-
Операційна стійкість	-	-	+
Залежність від мережі	+	+	+
Не потребують додаткове обладнання	+	-	-
Чутливість до хибних спрацьовувань	+	-	+
Вартість	Висока ціна	Висока ціна	Середня ціна, однак потребує додаткові витрати

1.3 Огляд існуючих програмно-апаратних рішень

1.3.1Notifier Honeywell

Honeywell розробляє технології для забезпечення життєдіяльності будівель. Вони пропонують бездротові датчики диму та станції пожежної сигналізації, а також ряд пристроїв, які можуть працювати з іншими дротовими продуктами Honeywell, такими як сирени та панелі управління [26].

FCM-1 від Honeywell NOTIFIER – це інтелектуальний адресний модуль керування, який контролює проводку для підключення пристроїв, таких як сирени, стробоскопи або дзвінки, які потребують зовнішнього джерела живлення [27].

Переваги:

- високий рівень стійкості до помилок;
- існують різні варіанти системи, починаючи від невеликих установок і закінчуючи великими комплексними мережами;
- інтуїтивно зрозумілий інтерфейс.

Недоліки:

- висока вартість;
- залежність від навчання персоналу;
- великі витрати на обслуговування.

Зовнішній вигляд пристрою представлено на рис. 1.4.

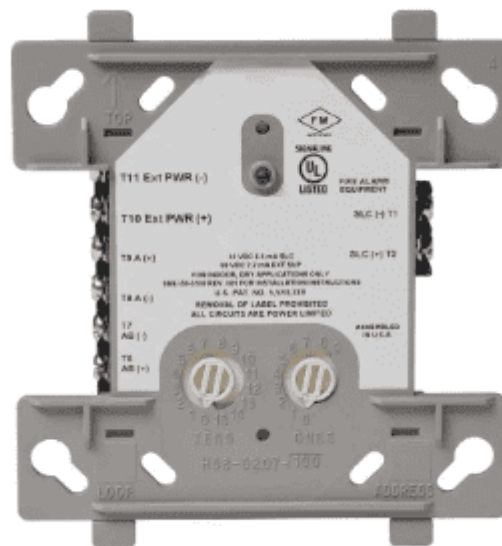


Рисунок 1.4 – Honeywell Notifier Fire Alarm Relay Module [27]

1.3.2 Desigo Fire Safety

Desigo CC - це відкрита платформа для управління. Може підтримувати різні, за об'ємом, системи [28].

Desigo Fire Safety, інтегрована система пожежної сигналізації від Siemens, може виявити, сигналізувати та реагувати на пожежі у різних типах

будівель. У системі є багато можливостей для підвищення рівня ПБ та захисту людей і майна [29].

Переваги:

- систему легко налаштувати для як великих так і маленьких об'єктів;

- швидка передача сигналів;

- легке обслуговування.

Недоліки:

- складно інтегрувати з іншими системами;

- залежність від навчання персоналу;

- висока вартість встановлення;

- великі витрати на обслуговування системи;

- встановлення системи потребує значних змін у інфраструктурі будівлі.

Зовнішній вигляд пристрою представлено на рис. 1.5.



Рисунок 1.5 – Desigo Fire Safety [29]

1.3.3 Bosch Building Integration System

Програмне рішення BIS об'єднує різні підсистеми безпеки Bosch, такі як управління доступом, відеоспостереження, пожежна сигналізація, системи оповіщення та захист від незаконного проникнення. Система може бути адаптована до різних розмірів об'єктів, починаючи від невеликих будівель і закінчуючи великими корпоративними комплексами [30].

Переваги:

- зберігає історію подій;
- широка інтеграція та інтуїтивно зрозумілий інтерфейс.

Недоліки:

- висока вартість;
- залежність від навчання персоналу;
- великі витрати на обслуговування системи.

1.4 Постановка завдань дипломної роботи

Отже, роблячи висновок огляду існуючих аналогів, можна сказати що основними проблемами є висока вартість систем, потреба в додатковому обладнанні, що ще більше впливає на ціну. Тому використання недорогих рішень значно знизить вартість системи, зробивши її доступнішою для широкого кола користувачів.

Мета роботи – розробка програмно-апаратного забезпечення підсистеми пожежної безпеки для моніторингу і раннього виявлення пожеж в системі "розумний" будинок.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- виконати аналіз вимог до проєктованої ППБ;
- обрати апаратні та програмні засоби розробки;
- виконати програмну реалізацію та тестування ППБ;
- розробити інструкцію з використання ППБ.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз вимог

Проєктowana ППБ має забезпечувати:

- моніторинг температури. Система повинна постійно перевіряти рівень температури в приміщенні. Коли температура перевищує порогове значення, система повинна перейти до перевірки наявності пожежі;
- моніторинг полум'я. При перевищенні температурного порогу система повинна перевіряти наявність полум'я. Якщо виявиться полум'я, система повинна надіслати повідомлення про небезпеку та увімкнути спринклер і сирену;
- моніторинг диму. Система повинна перевіряти рівень диму. Коли рівень диму перевищує поріг, система повинна надіслати повідомлення, увімкнути сирену та автоматично відкрити вентиляцію та вікна;
- при виявленні небезпечних ситуацій, таких як пожежа, витік газу або висока температура, система повинна надсилати користувачеві SMS-повідомлення;
- у випадку виявлення небезпеки система повинна вмикати сирену;
- на LCD-дисплеї система повинна відображати відповідні повідомлення.

2.2 Розробка архітектури системи

Архітектура ППБ (рис. 2.1) складається з основних компонентів: центрального контролера Arduino, датчиків, системи відображення інформації та оповіщення.

Зчитування даних датчиків, обробка цих даних і керування іншими частинами системи належать до обов'язків центрального контролера. Мікроконтролер Arduino є основою контролера, виконує всю логіку та зв'язок.

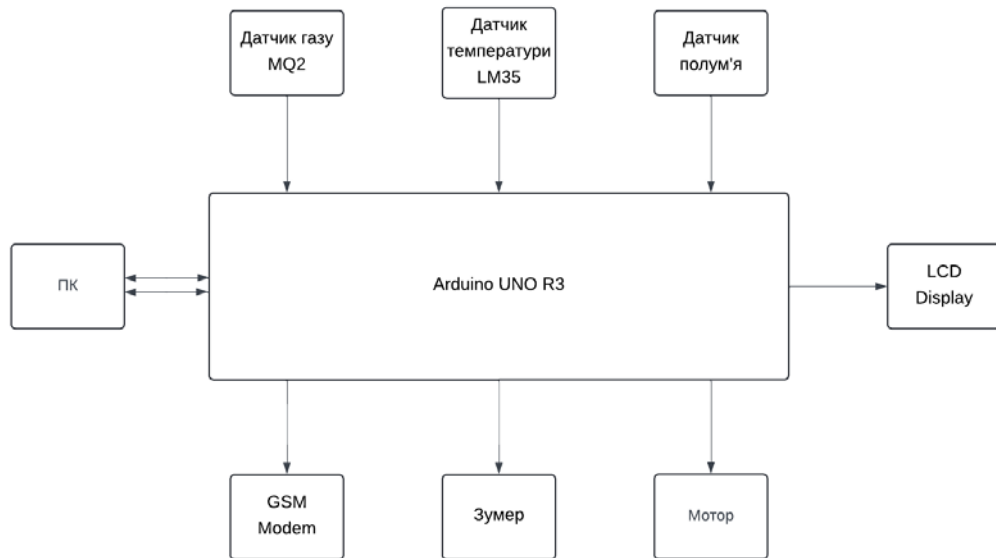


Рисунок 2.1 – Архітектура підсистеми

Забезпечуючи збір даних з навколишнього середовища, датчики є важливою частиною системи. Для зчитування даних датчики підключаються до аналогового входу Arduino. Датчики можна класифікувати на три основні категорії:

- датчик газу: контролює наявність газів, таких як метан, пропан і дим, у повітрі;
- датчик температури: надає можливість вимірювати температуру повітря;
- датчик полум'я використовується для виявлення осередка загоряння.

Системи відображення інформації відповідають за взаємодію з користувачем. Ця система складається з LCD-дисплея, який відображає поточні показники датчиків та стан системи та сповіщення. У разі небезпеки, Arduino активує зумер і відправляє SMS-повідомлення на заздалегідь встановлені номери телефонів, що відбувається через GSM модем.

Такий підхід до розробки архітектури системи гарантує, що система працює добре, є надійною та доступною для широкого кола користувачів.

2.3 Вибір мови та середовища програмування

Arduino — це венчурна та клієнтська локальна організація, яка є організатором обладнання та програмування з відкритим вихідним кодом, який допомагає планувати та виготовляти одноплатні мікроконтролери та набори мікроконтролерів для побудови комп'ютеризованих пристроїв[14]. Arduino створив власне програмне забезпечення під назвою інтегроване середовище розробки (IDE), яке підтримує мови програмування C та C++ для написання коду. Плати Arduino можуть зчитувати вхідні дані і перетворювати їх на вихідні [15]. Так за допомоги Arduino кожен має можливість для вдосконалення свого життя.

На платформах Arduino зазвичай використовують мову програмування C++. Використання C++ дозволяє ефективно керувати ресурсами, що є важливим для мікроконтролерів з обмеженою пам'яттю та обчислювальними можливостями. C++ є потужним інструментом для процедурного та об'єктно-орієнтованого програмування, що дозволяє створювати більш структурований і модульний код. Можливість використовувати ООП у C++ дозволяє структурувати код у вигляді класів і об'єктів. Розподіл коду на логічні модулі допомагає краще організувати його. Це особливо корисно для великих проєктів, де розділення функціональності на окремі класи та модулі є життєво важливим.

Через обмежені ресурси цих пристроїв і відсутність рідної підтримки цієї мови на них, зазвичай не використовується Python для програмування мікроконтролерів Arduino. Тим не менш, Python можна використовувати в проєктах з Arduino різними способами. Найчастіше він використовується для керування, налаштування, збору даних, обробки та інтеграції з іншими системами. Python можна використовувати для написання скриптів, які взаємодіють з Arduino через послідовний порт. Це особливо корисно для збору даних, керування Arduino з комп'ютера або розробки прототипів.

Незважаючи на те, що Java не використовується безпосередньо для програмування мікроконтролерів Arduino, вона може бути корисною для створення програм, які працюють з комп'ютерами або серверами через серіальну комунікацію. Коли проєкт вимагає інтеграції з серверними системами, базами даних або веб-додатками, Java може надати інструменти та бібліотеки, необхідні для цього, додатково дозволяючи програмі працювати на різних платформах, таких як Windows, macOS і Linux.

C++ найкращий варіант для прямого програмування Arduino, оскільки він ефективно використовує ресурси та має високу продуктивність. Ідеально підходить для низькорівневого апаратного забезпечення та проєктів реального часу. Таблиця 2.1 містить порівняння C++ і інших мов програмування.

Таблиця 2.1 – Порівняння C++ та інших мов програмування

Критерій порівняння	C++	Java	Python
Пряме програмування Arduino	Так (основна мова для Arduino IDE)	Ні (непрактично для мікроконтролерів)	Ні (непряме використання через скрипти)
Продуктивність	Висока	Середня	Середня
Швидкість розробки	Середня	Низька	Висока
Типові сценарії використання	Пряме підключення Arduino, проєкти в реальному часі та ефективне використання ресурсів	Додаток для зовнішнього керування Arduino, інтегрований у великі системи	Для швидкого прототипування та інтеграції з науковими додатками зовнішні скрипти для керування Arduino

2.4 Вибір апаратних рішень

Для того, щоб продовжити проектування підсистеми, необхідно визначити її компоненти відповідно до заданих функцій.

2.4.1 Датчик газу

Датчик газу MQ2 (рис. 2.2) відстежує концентрацію різних газів у повітрі, включаючи пропан, метан, водень, алкоголь, димові гази та чадний газ. Газ виявляють за допомогою зміни значення опору: чутливий матеріал змінює свій опір, коли стикається з газом [10]. Датчик працює під напругою 5 В. Він здатний ідентифікувати гази в діапазоні концентрацій від 300 до 10 000 ppm.



Рисунок 2.2 – Датчик газу MQ2 [11]

Порівняльну характеристику датчиків газу наведено в табл. 2.2.

Порівняльну характеристику датчиків температури наведено в табл. 2.3.

Таблиця 2.3 – Порівняння датчиків температури [13], [33]-[34]

Критерій порівняння	LM35	TMP36	DHT22
Діапазон вимірювання	-55°C до +150°C	-40°C до +125°C	-40°C до +80°C
Точність	±0.5°C	±2°C	±0.5°C
Критерій порівняння	LM35	TMP36	DHT22
Час відгуку	<1 секунда	<1 секунда	<2 секунди
Калібрування	Не потрібно	Не потрібно	Потрібно
Ціна	Низька	Низька	Середня

2.4.3 Arduino

Однією з найпопулярніших плат Arduino є Arduino UNO R3. У порівнянні з оригінальною платою Arduino UNO R3 версія CH340 дешевша, що робить її доступною для бюджетних проєктів. Порівняння різних моделей Arduino наведено в табл. 2.4.

Таблиця 2.4 – Порівняння різних моделей Arduino [16], [35]-[36]

Критерій порівняння	Arduino UNO R3 (CH340)	Arduino Mega 2560	Arduino Due
1	2	3	4
Пам'ять	32 KB	256 KB	512 KB
Розмір	68.6 мм x 53.4 мм	101.6мм x 53.3 мм	102 мм x 53 мм

Продовження таблиці 2.4

1	2	3	4
Критерій порівняння	Arduino UNO R3 (CH340)	Arduino Mega 2560	Arduino Due
Кількість цифрових входів/виходів	14 (6 PWM)	54 (15 PWM)	54 (12 PWM)
Частота тактування	16 MHz	16 MHz	84 MHz
Ціна	Низька	Середня	Висока

Зовнішній вигляд плати Arduino UNO R3 наведено на рис.2.4.



Рисунок 2.4 – Arduino UNO R3 (CH340) [16]

2.4.4 LCD Display

Рідкокристалічні дисплеї (LCD Display), також відомі як РК-дисплеї, є найкращим вибором для портативних пристроїв з живленням від батареї, від наручних годинників до телевізорів і портативних комп'ютерів [17]. У РК-дисплеї використовується або активна матриця, або пасивна екранна сітка.

Більшість електронних пристроїв використовують технологію рідкокристалічних дисплеїв для відображення інформації. Рідина споживає менше енергії, ніж світлодіоди або електронно-променева трубка [18].

Переваги LCD-дисплеїв: РК-дисплеї споживають лише мікровати, а не міліватти, як світлодіоди, і вони споживають меншу кількість енергії порівняно з ЕПТ і світлодіодами. Порівняно з електронно-променевими трубками та світлодіодами, РК-дисплеї легші та тонші. Крім того, вони недорогі [18].

Зовнішній вигляд LCD дисплею наведено на рис.2.5.

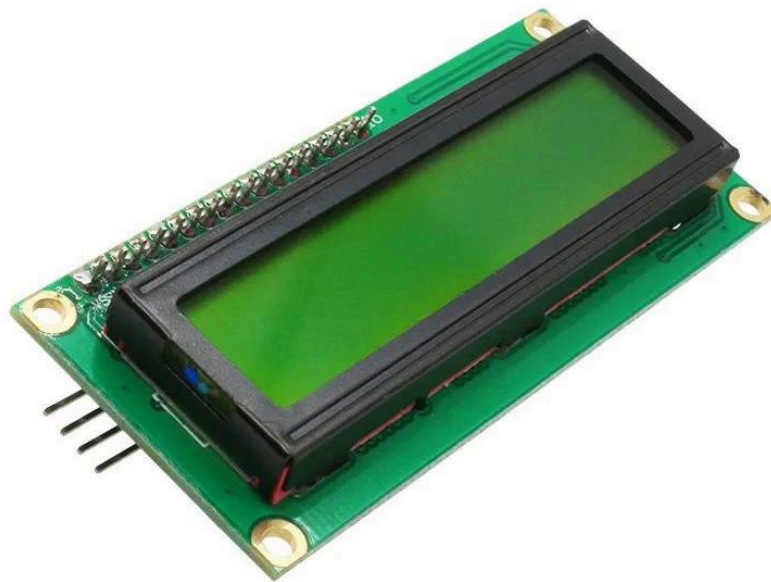


Рисунок 2.5 – LCD дисплей 16X2[19]

2.4.5 Датчик полум'я

Датчик полум'я використовує інфрачервоне випромінювання. Інфрачервоне випромінювання будь-якого гарячого тіла можна виявити за допомогою інфрачервоного фотодіода (ІЧ-фотодіода). Після цього це значення порівнюється з попереднім. Порогове значення випромінювання змінює вивід датчика. Датчик полум'я може виявити вогонь або інші світлові потоки на довжині хвилі 760–1100нм. Датчик інфрачервоного полум'я має два виводи: цифровий (LOW/HIGH) і аналоговий [20].

Зовнішній вигляд датчика полум'я наведено на рис.2.6.

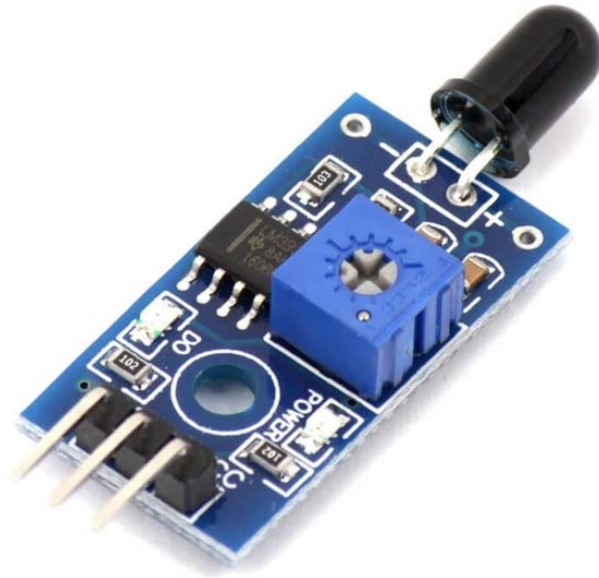


Рисунок 2.6 – Flame Sensor [21]

2.4.6 GSM Modem

GSM-модуль (рис.2.7) - це мікросхема або схема, яка встановлює зв'язок між мобільним пристроєм і системою GSM. Приймаючи SIM-карту від будь-якого оператора мережі GSM, він працює так само, як мобільний телефон із власним номером телефону [22].

2.4.7 Зумер

Зумер (рис.2.8) є електронною частиною, яка використовується для створення звукових сигналів. Це пристрій звукової сигналізації, який може бути електромеханічним, механічним або п'єзоелектричним. Різні електронні пристрої використовують стандартні зумери для передачі звукових сигналів, попередження або вказівки подій, сигналізації та таймери [24].



Рисунок 2.7 – Modem GSM GPRS SIM900 [23]



Рисунок 2.8 – Зумер [25]

2.5 Висновки до розділу 2

У цьому розділі були розглянуті вимоги до підсистеми пожежної безпеки, розроблено архітектуру, обрано мову програмування та апаратні рішення.

Розглянуто можливість використання інших мов програмування, таких як Python та Java, які виявилися менш практичними для прямого програмування мікроконтролерів Arduino.

Було визначено, які датчики будуть використовуватися. Хоча усі елементи легко замінюються аналогами, тож користувач може обрати найбільш підходящий варіант саме для нього. Система була розроблена таким чином, щоб вона була ефективною, надійною та доступною для використання в різних умовах.

3 КОНСТРУЮВАННЯ ПІДСИСТЕМИ ПРОТИПОЖЕЖНОЇ БЕЗПЕКИ

3.1 Алгоритм роботи підсистеми

Схему алгоритму роботи підсистеми наведено на рис.3.1.

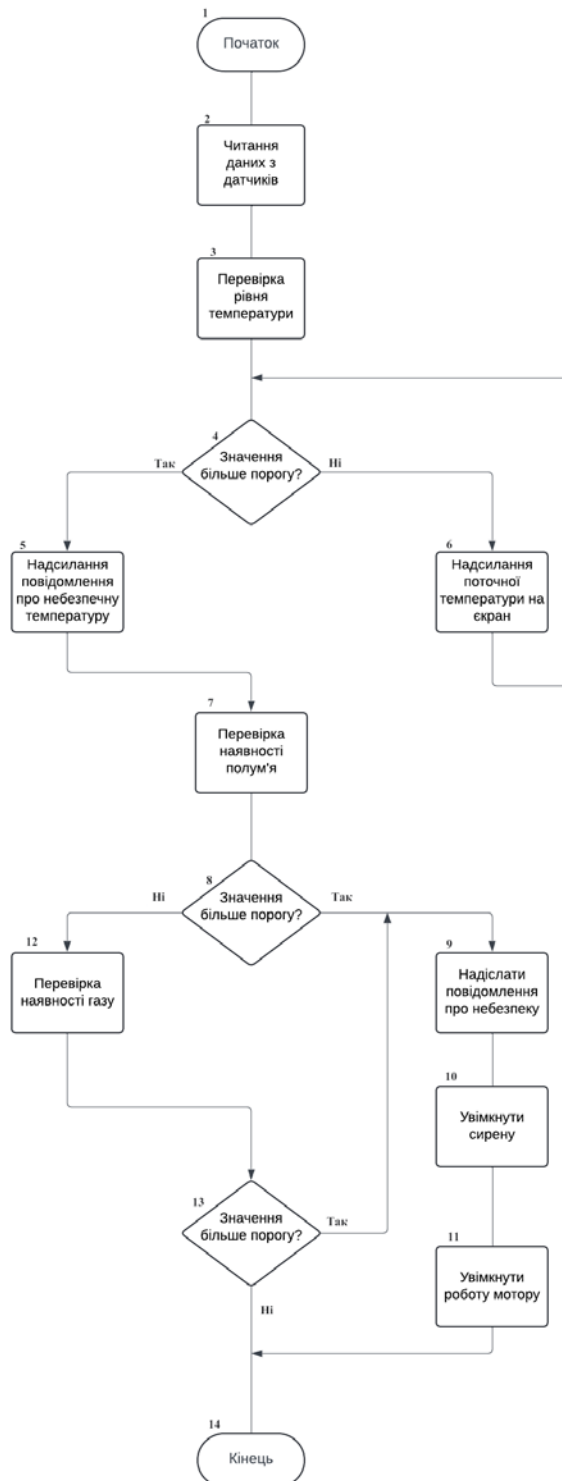


Рисунок 3.1 – Схема алгоритму роботи підсистеми

Блок 1. Початок процесу моніторингу.

Блок 2. Читання даних з датчиків. Система отримує дані від різних датчиків.

Блок 3. Перевірка рівня температури. Система перевіряє поточний рівень температури.

Блок 4. Чи більше значення температури за порогове? Якщо рівень температури перевищує встановлене порогове значення, переходимо до блоку 5. Якщо ні, переходимо до блоку 6.

Блок 5. Надсилання повідомлення про небезпечну температуру. Система надсилає повідомлення про небезпечний рівень температури.

Блок 6. Надсилання поточної температури на екран. Якщо температура в нормі, система відправляє поточну температуру на екран.

Блок 7. Перевірка наявності полум'я. Система перевіряє, чи є ознаки полум'я.

Блок 8. Чи більше значення за порогове? Якщо виявлено полум'я, переходимо до блоку 9. Якщо ні, переходимо до блоку 12.

Блок 9. Надіслати повідомлення про небезпеку. Система надсилає повідомлення про небезпеку через виявлення полум'я чи витоку газу.

Блок 10. Увімкнути сирену. Якщо виявлено небезпеку, система вмикає сирену.

Блок 11. Увімкнути роботу мотору. Система запускає мотор для вентиляції чи спринклеру.

Блок 12. Перевірка наявності газу. Система перевіряє рівень газу в приміщенні.

Блок 13. Чи більше значення за порогове? Якщо рівень газу перевищує встановлене порогове значення, переходимо до блоку 9. Якщо ні, переходимо до блоку 14.

Блок 14. Кінець. Завершення процесу моніторингу.

3.2 Опис апаратної реалізації підсистеми

Схему ППБ, створену у Proteus 8 Professional наведено на рис. 3.2.

Мікроконтролер постійно стежить за станом датчиків газу, полум'я та температури. Коли мікроконтролер включений, система запускає всі потрібні пін-коди, регулює дисплей LCD та встановлює початкові значення датчиків.

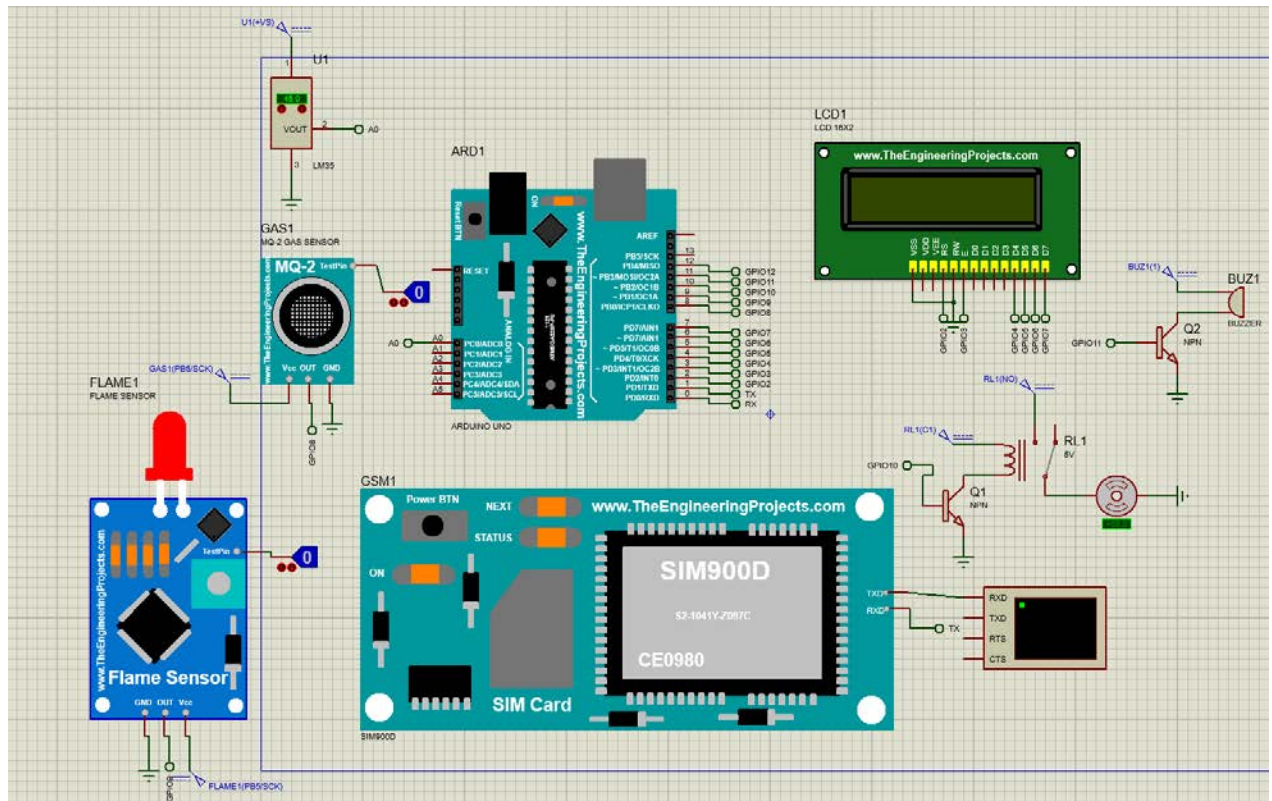


Рисунок 3.2 - Рішення підсистеми пожежної безпеки у Proteus 8 Professional

Основна структура роботи у безпервному циклі циклу система перевіряє стан датчиків газу, полум'я та температури. Виявлення небезпеки пожежі: якщо виявиться газ, полум'я або висока температура (понад 50 градусів за Цельсієм), система розпізнає пожежу.

Користувач отримує SMS-повідомлення, коли система вмикає сирену (buzzer), спринклер (motor_pin) і сирену. Відправка повідомлень виконується за допомогою функції SendMessage. Якщо жодна небезпечна ситуація не виявлена, система вимикає сирену та спринклер і відображає це на LCD.

3.3 Опис програмної реалізації підсистеми

Програмна реалізація використовує мікроконтролер Arduino для збору даних з датчиків, відображення інформації на LCD-дисплеї та вжиття заходів при виявленні небезпеки.

Таблиця 3.1 містить команди для підключення бібліотек та визначення постійних значень, які використовуються програмою для підключення та керування різними пристроями та датчиками. Константи визначають номери пінів для підключення різних пристроїв, а бібліотеки надають готові функції для роботи з апаратним забезпеченням.

Таблиця 3.1 – Бібліотеки та константи

Команда	Опис
<code>#include <LiquidCrystal.h></code>	Підключення бібліотеки для керування LCD дисплеєм
<code>const int gas = 8;</code>	Пін для датчика газу MQ2
<code>const int flame = 9;</code>	Пін для датчика полум'я
<code>const int motor_pin = 10;</code>	Пін для мотора
<code>const int buzzer_Pin = 11;</code>	Пін для зумера
<code>const int lm35_Pin = A0;</code>	Аналоговий пін для датчика температури LM35
<code>LiquidCrystal lcd(2, 3, 4, 5, 6, 7);</code>	Створення об'єкта lcd для керування LCD дисплеєм

Далі ми ініціалізуємо змінні, що використовуються в програмуванні для керування ППБ, для моніторингу стану датчиків. Ці змінні необхідні для зберігання значення температури та поточного стану датчиків полум'я та газу наведено у таблиці 3.2.

Таблиця 3.2 – Ініціалізація змінних

Змінна	Тип	Початкове значення	Опис
bool gas_Status	bool	LOW	Стан датчика газу
bool flame_Status	bool	LOW	Стан датчика полум'я
float temperature	float	0.0	Поточна температура

У функції `setup()`, що наведена у таблиці 3.3, містяться команди, які виконуються при запуску програми. Вона регулює серійний зв'язок, режими пінів, внутрішню опорну напругу для аналогового зчитування, ініціалізує дисплей і виводить початкове повідомлення.

Таблиця 3.3 – Функція `setup()`

Команда	Опис
1	2
<code>Serial.begin(9600);</code>	Ініціалізація серійного зв'язку зі швидкістю 9600 бод
<code>pinMode(buzzer_Pin, OUTPUT);</code>	Встановлення режиму роботи піну для зумера як вихід
<code>pinMode(motor_pin, OUTPUT);</code>	Встановлення режиму роботи піну для мотора як вихід
<code>pinMode(gas, INPUT);</code>	Встановлення режиму роботи піну для датчика газу як вхід
<code>pinMode(flame, INPUT);</code>	Встановлення режиму роботи піну для датчика полум'я як вхід

Продовження таблиці 3.3

1	2
<code>analogReference(INTERNAL);</code>	Використання внутрішньої опорної напруги 1.1В для аналогового зчитування
<code>lcd.begin(16, 2);</code>	Ініціалізація LCD дисплея з розміром 16 символів на рядок і 2 рядки
<code>lcd.print("Fire DETECTION");</code>	Виведення тексту "Fire DETECTION" на дисплей
<code>lcd.setCursor(0, 1);</code>	Встановлення курсору на другий рядок дисплея
<code>lcd.print("SYSTEM");</code>	Виведення тексту "SYSTEM" на дисплей
<code>delay(1000);</code>	Затримка на 1 секунду
<code>lcd.clear();</code>	Очищення дисплея

Команди, які можуть бути виконані в у безкінечному циклі, доступні за допомогою функції `loop()`. Вони оновлюють змінну температуру з затримкою у 100 мілісекунд і зчитують стан датчиків. У таблиці 3.4 наведені умови та дії, що виконуються у функції `loop()` при виявленні небезпеки.

Таблиця 3.4 – Функція `loop()`

Команда	Опис
1	2
<code>gas_Status = digitalRead(gas);</code>	Зчитування стану датчика газу
<code>flame_Status = digitalRead(flame);</code>	Зчитування стану датчика полум'я
<code>temperature = readTemperature();</code>	Зчитування температури

Продовження таблиці 3.4

1	2
<code>delay(100);</code>	Затримка на 100 мілісекунд
<code>if (gas_Status == HIGH && flame_Status == HIGH && temperature >= 50)</code>	Виведення "Fire Detected", увімкнення зумера і мотора
<code>else if (flame_Status == HIGH && temperature >= 50)</code>	Виведення "Fire Detected", увімкнення зумера і мотора
<code>else if (flame_Status == HIGH)</code>	Виведення "Fire Detected", увімкнення зумера і мотора
<code>else if (gas_Status == HIGH)</code>	Виведення "Gas Leak Detected", увімкнення зумера і мотора
<code>else if (temperature >= 50)</code>	Виведення "High Temp Detected!" та поточної температури, увімкнення зумера і мотора
<code>else</code>	Вимкнення зумера і мотора при відсутності небезпеки

Команди функції `readTemperature()` містяться в таблиці 3.5 і використовуються для зчитування аналогового значення термометра, перетворення його у напругу та обчислення температури.

Таблиця 3.5 – Функція `readTemperature()`

Команда	Опис
1	2
<code>int sensorValue = analogRead(lm35_Pin);</code>	Зчитування аналогового значення з піну датчика температури

Продовження таблиці 3.5

1	2
<code>float voltage = sensorValue * (1.1 / 1023.0);</code>	Перетворення значення напруги з урахуванням внутрішньої опорної напруги 1.1В
<code>float temperatureC = voltage * 100;</code>	Перетворення напруги у температуру (чутливість 10мВ/°С)
<code>return temperatureC;</code>	Повернення значення температури

Таблиця 3.6 містить команди функції `SendMessage()`, які використовуються для відправки SMS-повідомлень за допомогою модуля GSM.

Таблиця 3.6 – Функція `SendMessage()`

Команда	Опис
<code>Serial.println("AT+CMGF=1");</code>	Встановлення режиму текстових повідомлень для GSM модуля
<code>Serial.println("AT+CMGS=\"+380977877501\"\\r");</code>	Введення номера телефону для відправки повідомлення
<code>Serial.println((char)26)</code>	Завершення повідомлення та його відправка

3.4 Висновки до розділу 3

Алгоритм роботи ППБ забезпечує повне та ефективне виявлення пожеж у приміщенні. Початковим кроком процесу є моніторинг системи для вимірювання температури. Якщо температура перевищує порогове значення, система перевіряє наявність пожежі. При виявленні загрози надсилаються повідомлення та увімкнуться сирена та спринклер. Якщо рівень диму перевищує порогове значення, використовуються додаткові функції, такі як відправка повідомлень і автоматичне відкриття вікон і вентиляції.

Мікроконтролер використовується в апаратній реалізації підсистеми, щоб стежити за станом датчиків газу, полум'я та температури. У випадку виявлення загрози система автоматично вмикає сирену та спринклер і відправляє користувачеві SMS-повідомлення.

Програмна реалізація, яка базується на мікроконтролері Arduino, дозволяє збирати дані з датчиків, відображати інформацію на LCD-дисплеї та вживати заходів, необхідних при виявленні небезпеки. Коди налаштовують пін-коди для взаємодії з датчиками та іншими компонентами, а також обробляють сигнали, отримані від них, щоб визначити наявність небезпеки та забезпечити відповідне реагування.

За допомогою цього алгоритму, апаратної та програмної реалізації система мінімізує ризики та потенційні збитки, забезпечуючи надійний контроль та швидке реагування на пожежну небезпеку.

4 ІНСТРУКЦІЯ ЩОДО ВИКОРИСТАННЯ ПІДСИСТЕМИ ПРОТИ-ПОЖЕЖНОЇ БЕЗПЕКИ

Ця підсистема ППБ розроблена таким чином, щоб вона сама виявила пожежу та реагувала на неї. Вона складається з GSM-модуля для відправки повідомлень, сирени та механізму автоматичного відкриття вікон і вентиляції, а також датчиків температури, диму та полум'я, які можна підключити до центрального контролера (Arduino).

Для початку роботи, потрібно підключити компоненти. Підключити датчики температури, диму та полум'я до відповідних входів контролера Arduino згідно зі схемою.

Датчик температури (рис. 4.1) ми підключаємо VOUT -> A0, бо датчик LM35 видає аналоговий сигнал, пропорційний виміряній температурі. Для цього Arduino має спеціальні аналогові входи, що вимірюють напругу.

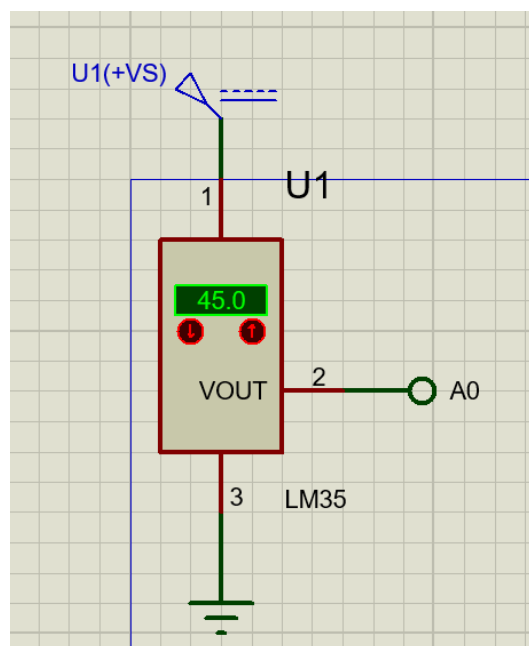


Рисунок 4.1 – Підключення датчика температури

Датчик полум'я, що представлений на рис. 4.2, ми підключаємо до D9. Цей пін використовується для зчитування цифрового сигналу датчика полум'я. При виявленні пожежі датчик полум'я видає високий сигнал (HIGH).

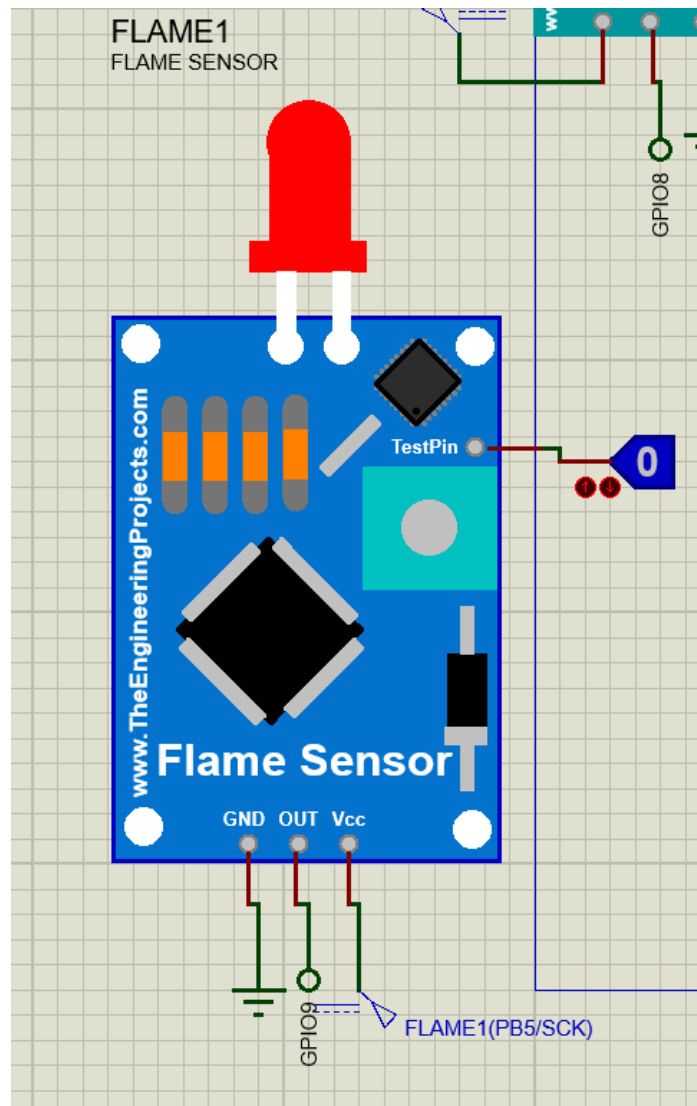


Рисунок 4.2 – Підключення датчика полум'я

Газовий датчик (MQ-2) підключається до піну D8 (рис. 4.3). Газовий датчик використовує цей пін для зчитування цифрового сигналу. Цей пін вказує на наявність чи відсутність газу вище певного порогу.

Для керування мотором (рис. 4.4), підключається пін D10, що керує реле яке вмикає або вимикає мотор. Arduino видає сигнал на цей пін, щоб активувати транзистор, який, у свою чергу, активує реле.

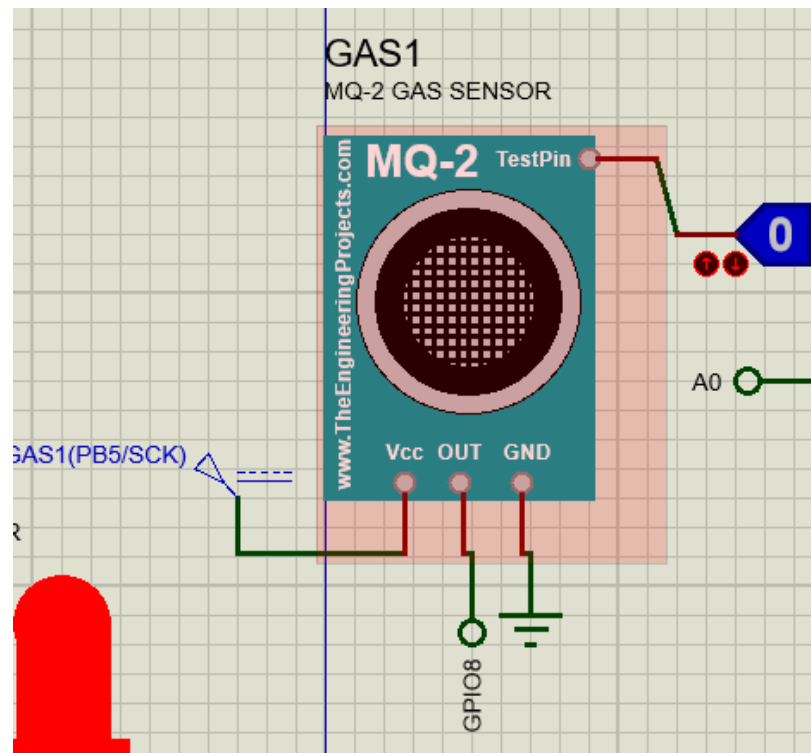


Рисунок 4.3 – Підключення датчику газу

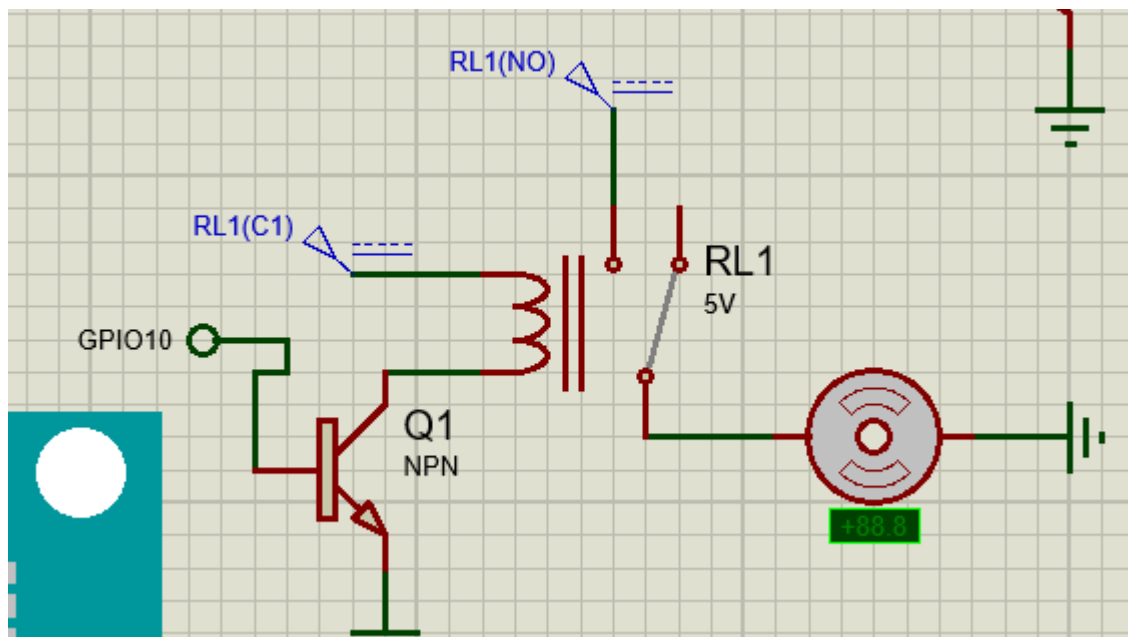


Рисунок 4.4 – Підключення мотору

Бuzzer (рис. 4.5), підключається до пін D11, Arduino видає сигнал на цей пін, щоб ввімкнути або вимкнути бuzzer. Пін D11 обрано, оскільки він є цифровим виходом, здатним видавати звук через бuzzer.

На рисунку 4.6, LCD підключається до пінів D2, D3, D4, D5, D6, D7. РК-дисплей потребує кілька цифрових пінів для передачі даних та керування.

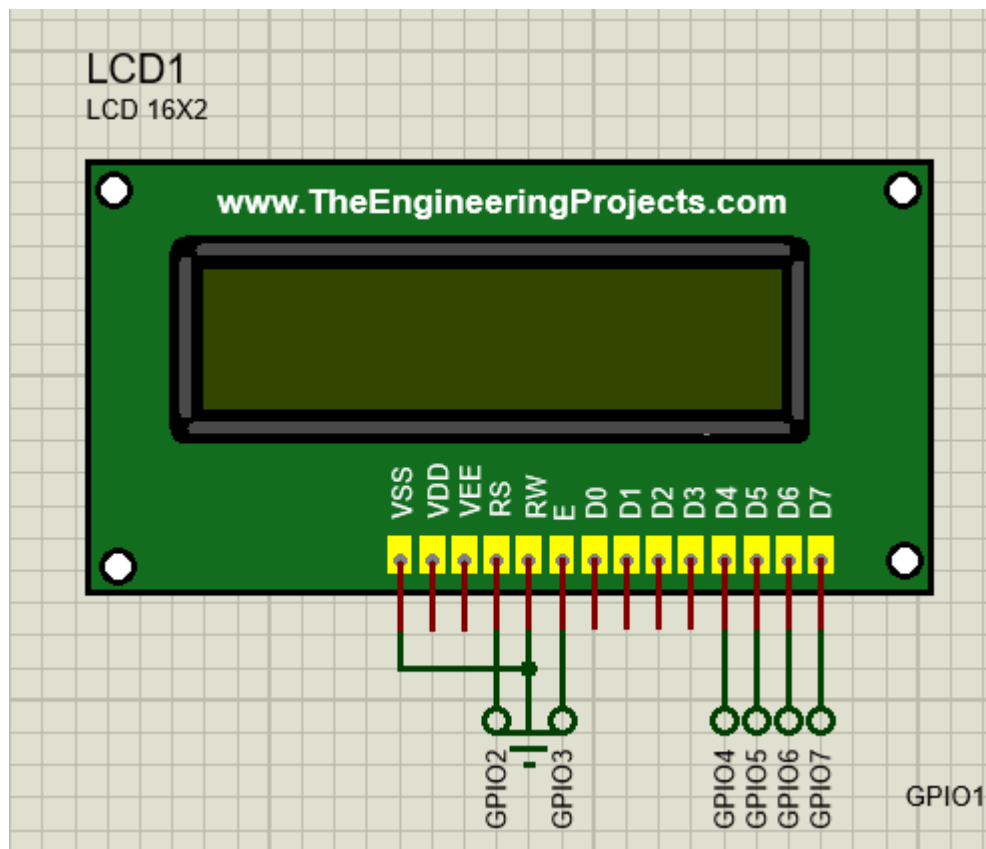


Рисунок 4.6 – Підключення LCD

GSM модуль, представлений на рисунку 4.7, використовує піни RXD і TXD, бо вони використовуються для передачі та прийому даних між Arduino та GSM модулем. Для взаємодії з контролером Arduino GSM-модуль використовує послідовний зв'язок (UART). Для обміну даними піни TXD (передача) і RXD (прийм) модулю підключаються до відповідних пінів Arduino.

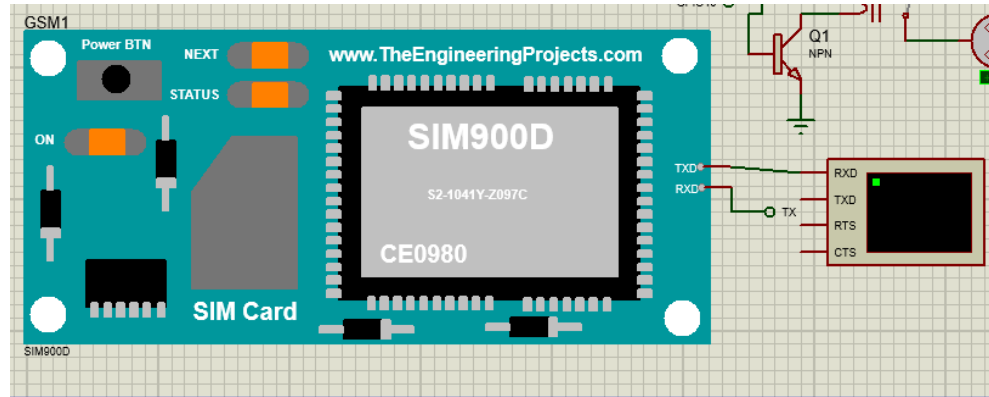


Рисунок 4.7 – Підключення GSM модулю

Після підключення усіх елементів, подаємо живлення до системи, та запускаємо її. Користувач має змогу бачити на дисплеї поточну температуру, та чи виявлено полум'я, чи безпеки немає. Роботу системи при підключенні усіх датчиків та елементів представлено на рисунку 4.8.

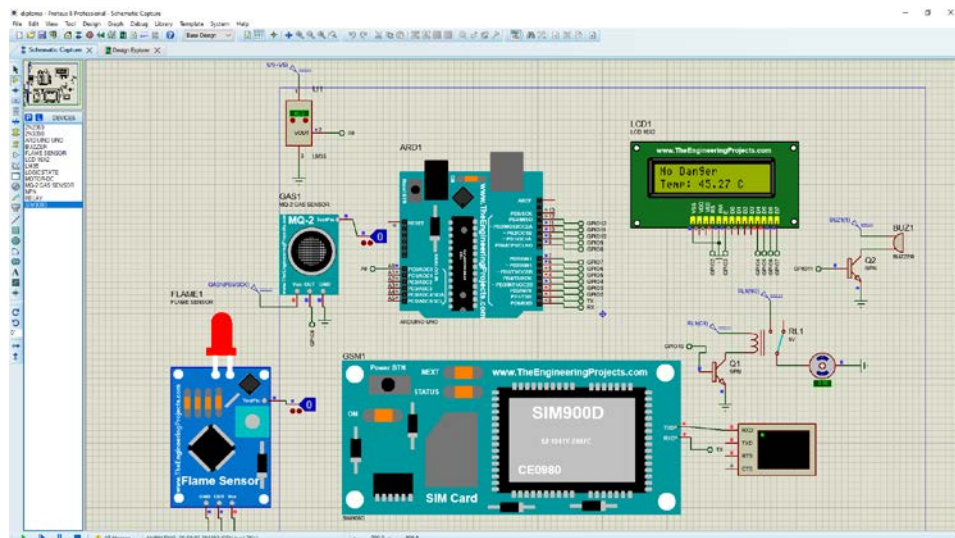


Рисунок 4.8 –Робота системи

У разі пожежи, користувач отримує відповідне повідомлення, та умикається сирена разом зі сплінкером (рис. 4.9).

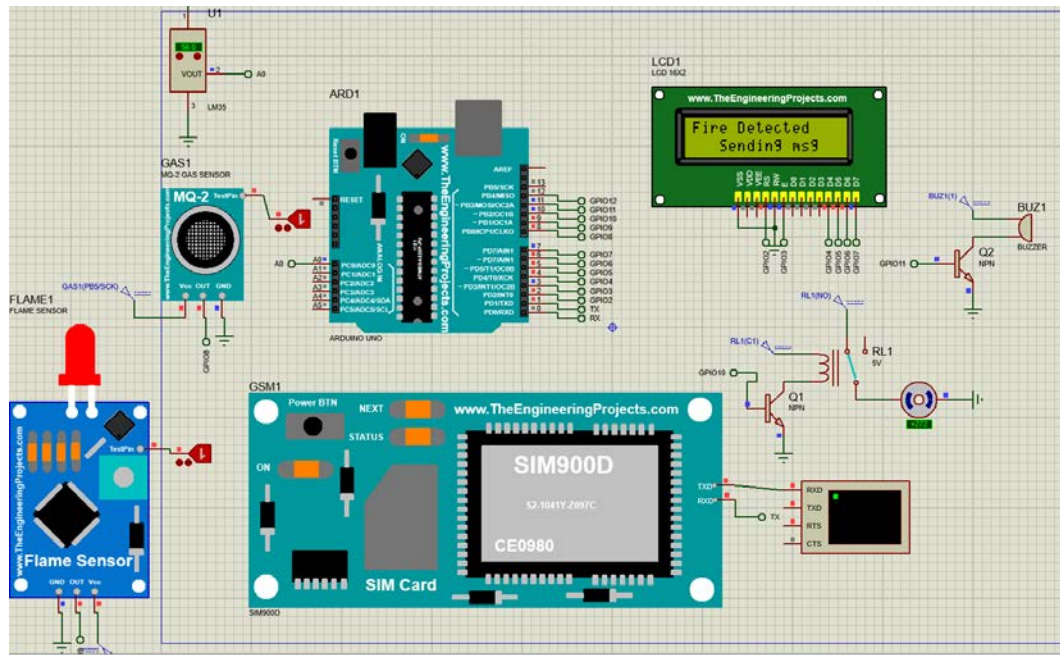


Рисунок 4.9 –Робота системи під час пожежі

У випадку, якщо стався витік газу (рис. 4.10), користувачеві буде надіслано повідомлення що знайдено витік газу, буде увімкнена сирена, та спрацює мотор, який автоматично відкриє вікна та вентиляцію.

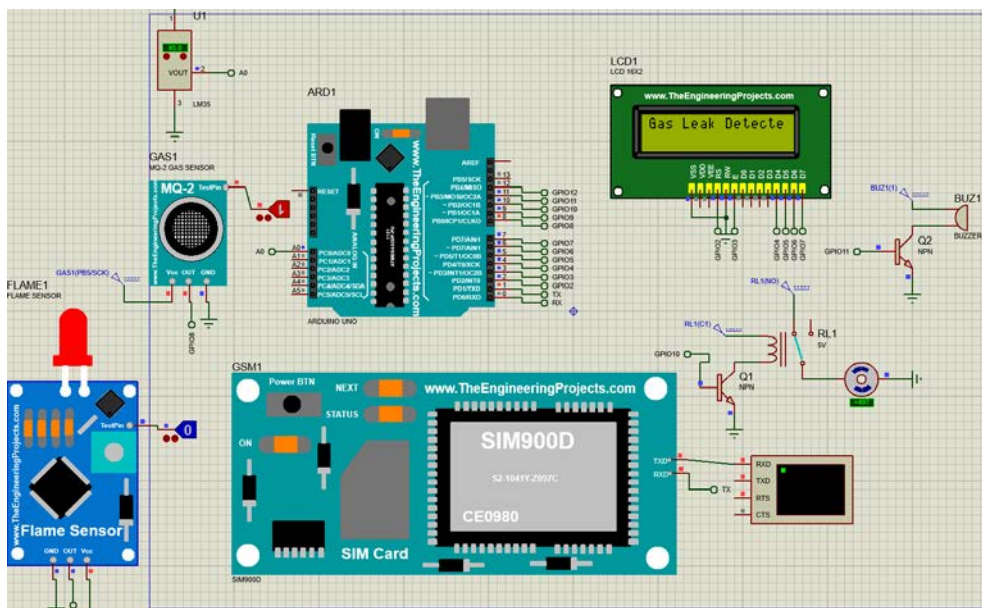


Рисунок 4.10 –Робота системи під час витоку газу

У разі, якщо було помічено лише високу температуру, без газу та полум'я, користувач отримує попередження про це з даною у реальному часі температурою (рис. 4.11).

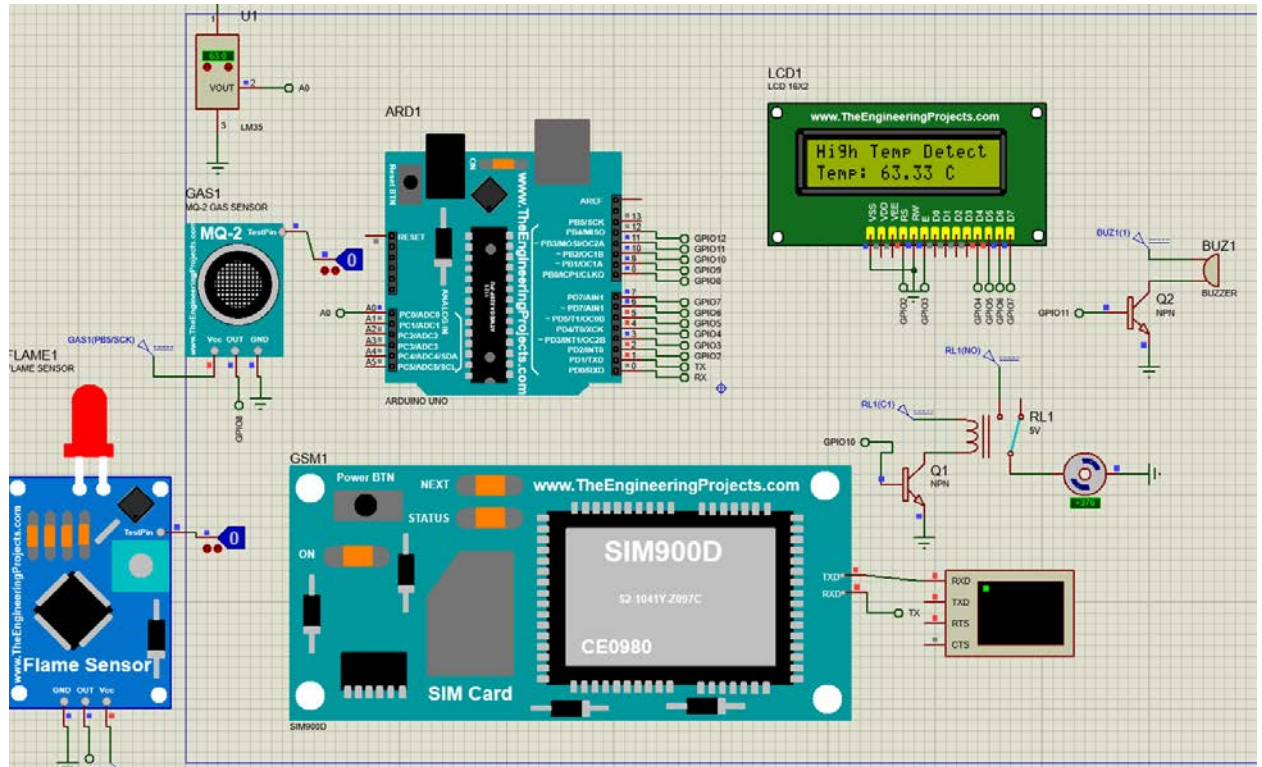


Рисунок 4.11 – Робота системи під час високої температури

ВИСНОВКИ

У дипломній роботі було розроблено ППБ для СРБ. Основна мета цієї роботи полягала в розробці надійної підсистеми виявлення пожежі, яка могла б сповіщати користувача про потенційну загрозу та оперативно реагувати на виявлення небезпечних ситуацій.

ППБ РБ демонструє високу ефективність і надійність. Вона пропонує багаторівневий захист, який включає віддалене інформування користувача, автоматичні дії для гасіння пожежі та звукове сповіщення. Такі системи можуть значно підвищити безпеку мешканців і знизити ризики пожеж.

Подальший розвиток системи може включати інтеграцію з іншими підсистемами розумного будинку, наприклад, системою керування освітленням чи клімат-контролем для автоматичного реагування на надзвичайні ситуації. Також перспективним напрямком є розробка мобільного додатку для віддаленого моніторингу та управління станом пожежної безпеки.

Результати роботи були представлені на НПК «Тиждень науки-2024».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Аналітична довідка про пожежі та їх наслідки в Україні за 12 місяців 2023 року [Електронний ресурс]. – Режим доступу: <https://idundcz.dsns.gov.ua/upload/2/0/1/8/2/6/2/analitchna-dovidka-pro-pojeji-122023.pdf>.
2. Пономаренко Є. О. Моделювання системи датчиків на платформі arduino для забезпечення пожежної безпеки / Є.О. Пономаренко, І.О. Розломий // Наука про цивільний захист як шлях становлення молодих вчених. Матеріали Всеукраїнської науково-практичної конференції курсантів, студентів, ад'юнктів (аспірантів). – Черкаси: Черкаський інститут пожежної безпеки імені Героїв Чорнобиля НУЦЗ України. – 2022. – С. 131-132.
3. S3000BWES Protect Carbon Monoxide Battery [Electronic source]. – Access mode: <https://www.amazon.com/S3000BWES-Protect-Carbon-Monoxide-Battery/dp/B00XV1RCRY>.
4. Google Nest Smoke Detector Review [Electronic source]. – Access mode: <https://www.bobvila.com/articles/google-nest-smoke-detector-review/>.
5. Датчик диму і чадного газу [Електронний ресурс]. – Режим доступу: <https://homehub.com.ua/uk/datchiki-dima/nest-protect-uk.html>.
6. Датчик задимлення і температури Fibaro Smoke Sensor [Електронний ресурс]. – Режим доступу: <https://homehub.com.ua/datchiki-dyma/fibaro-smoke-sensor-fibefgsd-002-fibfgss-101.html>.
7. Smoke Sensor FGSD-002 Test [Electronic source]. – Access mode: <https://forum.fibaro.com/topic/22140-smoke-sensor-fgsd-002-test/>.
8. Why Ajax [Electronic source]. – Access mode: <https://ajax.systems/ua/why-ajax/>.
9. FireProtectPlus Jeweller [Електронний ресурс]. – Режим доступу: <https://ajax.systems/ua/products/fireprotectplus/>.
10. IOT based Fire Detection System / [D. Pandey, R. Pawar, J. Sharma et al.] // Recent Trends in Computer Graphics and Multimedia Technology. – 2021. –

Vol. 3, Issue 1. – P. 1-7.

11. Модуль датчика дыму MQ-2. [Электронный ресурс]. – Режим доступа: <https://arduino.ua/ru/prod298-modyl-datchika-dima-mq-2>.

12. LM35 Precision Centigrade Temperature Sensors datasheet (Rev. H) [Electronic source]. – Access mode: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1188696/TGS/LM35.html>.

13. Датчик температуры LM35DZ. [Электронный ресурс]. – Режим доступа: https://arduino.ua/prod933-datchik-temperatyri-lm35dz?utm_source=GoogleMerchant&utm_medium=Merchant&utm_campaign=Google&utm_content=DAT152.

14. Rana, S. FIRE ALARM - A SAFETY DEVICE (THERMISTOR-TEMPERATURE SENSOR) / S. Rana, V. Dahiya, R. W. Mosobi. // International Research Journal of Modernization in Engineering Technology and Science. – 2021. – Vol. 3, Issue 6. – P. 3355 – 3360.

15. About Arduino [Electronic source]. – Access mode: <https://www.arduino.cc/en/about>.

16. Arduino UNO R3 CH340 [Электронный ресурс]. – Режим доступа: https://arduino.ua/prod2610-arduino-uno-r3-ch340?utm_source=GoogleMerchant&utm_medium=Merchant&utm_campaign=Google&utm_content=APC156.

17. Sarma, K. R. Liquid Crystal Displays [Electronic source] / K. R. Sarma. – Access mode: <http://www.autex.spb.su/download/wavelet/books/sensor/CH92.PDF>.

18. Construction and Working Principle of Custom LCD Display [Electronic source]. – Access mode: <https://www.cdtech-lcd.com/news/construction-and-working-principle-of-custom-lcd-displays.html>

19. LCD 1602 дисплей 16x2 зеленый [Электронный ресурс]. – Режим доступа: https://electrochip.mk.ua/ru/product/lcd-1602-dysplej-16h2-zelenyj/?srsltid=AfmBOop4b5jQ39XTFzW5zWMtLNgzvsmRn--yKnYLGiuL0VZGoGB_r5SvKP0.

20. Jobit, J. Interfacing Flame Sensor with Arduino. [Electronic source]/ J. Jobit. – Access mode: <https://circuitdigest.com/microcontroller-projects/interfacing-flame-sensor-with-arduino>.

21. Модуль GSM GPRS SIM900 Shield для Arduino [Електронний ресурс]. – Режим доступу: https://rozetka.com.ua/ua/396911547/p396911547/?utm_l=r&gad_source=4&gclid=Cj0KCQjw0ruyBhDuARIsANSZ3wqB1dmz9enzVtV7J30xvoSH5fu6fxLGoJ669Kf0uRIcqgqg-5YKTHIAaAqdfEALw_wcB.

22. Wireless Display using GSM and Arduino / [K. Kumar, K. Ritu, M. Singh et al.] // International Journal of Scientific Research in Computer Science, Engineering and Information Technology. – 2018. – Vol. 3, Issue 5. – P. 229-234.

23. Модуль GSM GPRS SIM900 Shield для Arduino. [Електронний ресурс]. – Режим доступу: https://epicentrk.ua/shop/mplc-modul-gsm-gprs-sim900-shield-dlya-arduino-gsm-850900-1800-1900-mgts-1ecd5d4f-ef05-6f9a-8228-91d4aa7ddf88.html?srsltid=AfmBOorLrS59Uh88wtfVA37CZNHWPk-aBs4KZjGRltuQ5x94D_3u8nGBrvE.

24. Активний динамік 12мм 5V Buzzer [Електронний ресурс]. – Режим доступу: https://arduino.ua/prod603-aktivnii-dinamik-12mm-5v-buzzer?utm_source=GoogleMerchant&utm_medium=Merchant&utm_campaign=Google&utm_content=AOC240.

25. Активний динамік 12мм 5V Buzzer [Електронний ресурс]. – Режим доступу: https://arduino.ua/prod603-aktivnii-dinamik-12mm-5v-buzzer?utm_source=GoogleMerchant&utm_medium=Merchant&utm_campaign=Google&utm_content=AOC240.

26. Honeywell Buildings [Electronic source]. – Access mode: <https://www.honeywellbuildings.in/#about>.

27. Notifier Modules [Electronic source]. – Access mode: <https://www.honeywellbuildings.in/fire/fire-detection-and-alarm-system/notifier/modules>.

28. Desigo CC [Електронний ресурс]. – Режим доступу:

<https://www.siemens.com/ua/uk/produkty/avtomatyzatsiya-ta-bezpeka-budivel/systemy-avtomatyzatsiyi-ta-dyspetcheryzatsiyi-budivel/desigo/systemy-avtomatyzatsiyi-budivel-desigo/desigo-cc.html>.

29. Desigo Fire Safety UL [Electronic source]. – Access mode: <https://www.siemens.com/global/en/products/buildings/fire-safety/protection-systems/desigo-fire-safety-ul.html>.

30. Bosch Security [Electronic source]. – Access mode: <https://www.boschsecurity.com/xc/en/>.

31. Модуль датчика газу MQ-7 [Електронний ресурс]. – Режим доступу: <https://arduino.ua/ru/prod1389-modyl-datchika-gaza-mq-7>.

32. Модуль датчика газу CCS811 [Електронний ресурс]. – Режим доступу: <https://www.kosmodrom.ua/datchik-gazu/ccs811-sensor-module.html>.

33. Датчик температури TMP36 від SparkFun [Електронний ресурс]. – Режим доступу: <https://arduino.ua/ru/prod687-datchik-temperatyri-tmp36-ot-sparkfun>.

34. Датчик вологості і температури DHT22 [Електронний ресурс]. – Режим доступу: <https://arduino.ua/ru/prod301-datchik-vlajnosti-i-temperatyri-dht22>.

35. Arduino Mega 2560 [Електронний ресурс]. – Режим доступу: <https://www.mini-tech.com.ua/arduino-mega-2560>.

36. Arduino Due [Електронний ресурс]. – Режим доступу: <https://www.mini-tech.com.ua/arduino-due>.

ДОДАТОК А
Текст програми

```
#include <LiquidCrystal.h>
```

```
const int gas = 8, flame = 9, motor_pin = 10, buzzer_Pin = 11;
```

```
const int lm35_Pin = A0; // Пін, до якого підключений сенсор LM35
```

```
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
```

```
bool gas_Status = LOW;
```

```
bool flame_Status = LOW;
```

```
float temperature = 0.0;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    pinMode(buzzer_Pin, OUTPUT);
```

```
    pinMode(motor_pin, OUTPUT);
```

```
    pinMode(gas, INPUT);
```

```
    pinMode(flame, INPUT);
```

```
    analogReference(INTERNAL);
```

```
    lcd.begin(16, 2);
```

```
    lcd.print("Fire DETECTION");
```

```
    lcd.setCursor(0, 1);
```

```
    lcd.print("SYSTEM");
```

```
    delay(1000);
```

```
    lcd.clear();
```

```
}
```

```
void loop() {
```

```
    gas_Status = digitalRead(gas);
```

```
    flame_Status = digitalRead(flame);
```

```
    temperature = readTemperature();
```

```

delay(100);

if (gas_Status == HIGH && flame_Status == HIGH && temperature >=
50) {
    lcd.clear();
    lcd.print("Fire Detected");
    lcd.setCursor(3,2);
    lcd.print("Sending msg");
    delay(1000);
    digitalWrite(buzzer_Pin, HIGH);
    delay(100);
    digitalWrite(motor_pin, HIGH);
    delay(100);

} else if (flame_Status == HIGH && temperature >= 50) {
    lcd.clear();
    lcd.print("Flame and High Temp Detected");
    lcd.setCursor(3,2);
    lcd.print("Sending msg");
    delay(1000);
    digitalWrite(buzzer_Pin, HIGH);
    delay(100);
    digitalWrite(motor_pin, HIGH);
    delay(100);

} else if (gas_Status == HIGH) {
    lcd.clear();
    lcd.print("Gas Leak Detected");
    lcd.setCursor(0, 1);
    digitalWrite(buzzer_Pin, HIGH);

```

```
digitalWrite(motor_pin, HIGH);
delay(100);

} else if (temperature >= 50) {
    lcd.clear();
    lcd.print("High Temp Detected!");
    lcd.setCursor(0, 1);
    lcd.print("Temp: ");
    lcd.print(temperature);
    lcd.print(" C");
    delay(500);
    digitalWrite(buzzer_Pin, LOW);
    digitalWrite(motor_pin, LOW);
    delay(100);

    } else if (flame_Status == HIGH) {
        lcd.clear();
        lcd.print("Flame Detected!");
        lcd.setCursor(3,2);
        lcd.print("Sending msg");
        delay(1000);
        digitalWrite(buzzer_Pin, HIGH);
        delay(100);
        digitalWrite(motor_pin, HIGH);
        delay(100);
    } else {
        lcd.clear();
        lcd.print("No Danger");
        lcd.setCursor(0, 1);
```

```

    lcd.print("Temp: ");
    lcd.print(temperature);
    lcd.print(" C");
    delay(500);
    digitalWrite(buzzer_Pin, LOW);
    digitalWrite(motor_pin, LOW);
    delay(100);
}
}

float readTemperature() {
    int sensorValue = analogRead(lm35_Pin);
    // Приведення значення до напруги з внутрішнім опорним значенням

```

1.1B

```

    float voltage = sensorValue * (1.1 / 1023.0);
    // Перетворення напруги у температуру (з урахуванням чутливості

```

10мВ/°C)

```

    float temperatureC = voltage * 100;
    return temperatureC;
}

```

```

void SendMessage(String message) {

```

```

    Serial.println("AT+CMGF=1"); // Встановлює модуль GSM в режим

```

тексту

```

    delay(1000); // Затримка на 1000 мілісекунд або 1 секунда
    Serial.println("AT+CMGS=\"" + 380977877501 + "\"r"); // номер мобільного
    delay(1000);
    Serial.println(message); // Текст SMS, який ви хочете відправити
    delay(100);
    Serial.println((char)26); // ASCII код CTRL+Z
    delay(1000);
}

```

ДОДАТОК Б
Слайди презентації



Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дипломна робота на тему
Програмна реалізація підсистеми
пожежної безпеки для системи розумний
будинки

Виконала:
ст. гр. КНТ 210

Керівник:
к.т.н., доцент

Наталя ЧУБКО

Анжеліка ПАРХОМЕНКО



Рисунок Б.1 – Слайд 1



Аналіз існуючих рішень для пожежної безпеки

1

Google Nest Protect

2

FIBARO Smoke Sensor - FIBCFGSD-002

3

Ajax FireProtect Plus

Критерій порівняння	Google Nest Protect	FIBARO Smoke Sensor - FIBCFGSD-002 (FIB_FGSS-101)	Ajax FireProtect Plus
Детектори	Дим, вуглекислий газ, вологість, температура	Дим, вуглекислий газ, підвищена температура	Дим, вуглекислий газ, підвищена температура
Бездротове підключення	+	+	+
Оповіщення на смартфон	+	+	+
Зручність	-	+	-
Операційна стійкість	-	-	+
Залежність від мережі	+	+	+
Не потребують додаткове обладнання	+	-	-
Чутливість до хибних спрацьовувань	+	-	+
Вартість	Висока ціна	Висока ціна	Середня ціна, однак потребує додаткові витрати

2

Рисунок Б.2 – Слайд 2

Мета та завдання роботи

Мета дипломної роботи - розробка програмно-апаратного забезпечення підсистеми пожежної безпеки для моніторингу і раннього виявлення пожеж в системі "розумний" будинок.

Завдання дипломної роботи:

- виконати аналіз вимог до проєктованої підсистеми пожежної безпеки;
- обрати апаратні та програмні засоби розробки;
- виконати програмну реалізацію та тестування підсистеми пожежної безпеки;
- розробити інструкцію з використання підсистеми пожежної безпеки.

3

Рисунок Б.3 – Слайд 3

Аналіз вимог до підсистеми пожежної безпеки

Система повинна постійно перевіряти рівень температури в приміщенні

Система повинна перевіряти наявність полум'я

Система повинна перевіряти рівень диму

Система повинна надсилати користувачеві SMS-повідомлення

Система повинна вмикати сирену при небезпеці

На LCD-дисплеї система повинна відображати відповідні повідомлення

4

Рисунок Б.4 – Слайд 4

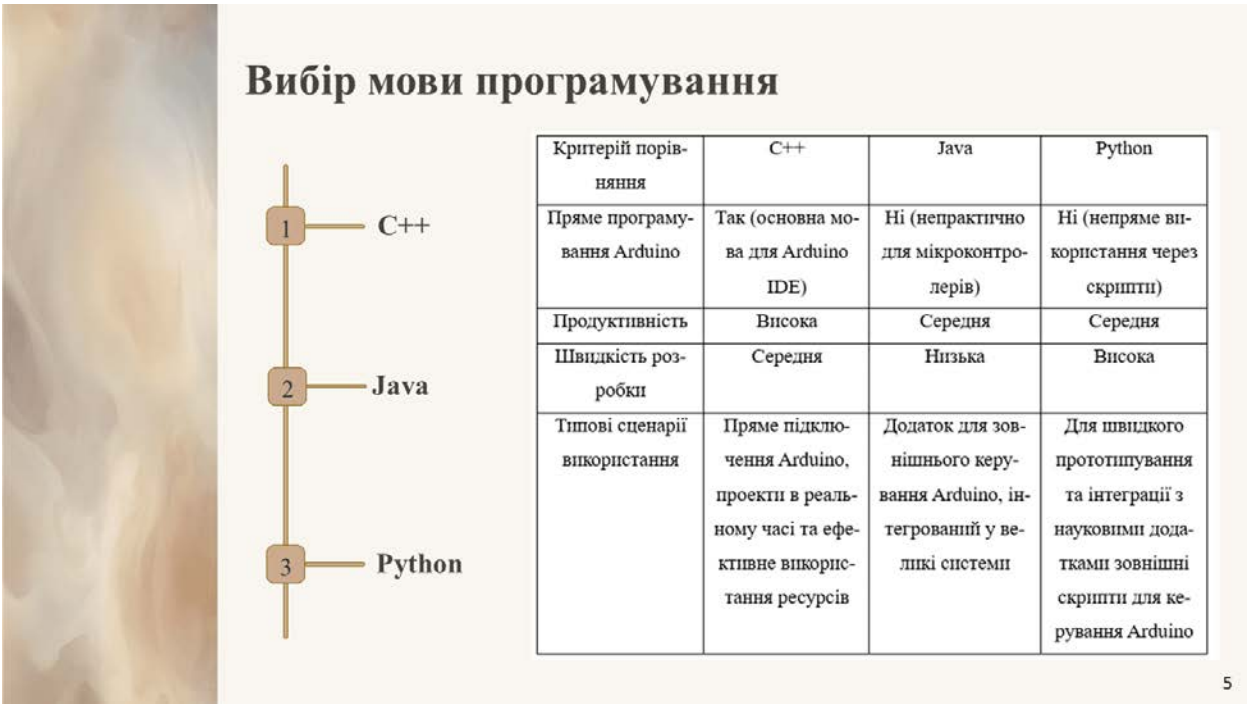


Рисунок Б.5 – Слайд 5



Рисунок Б.6 – Слайд 6

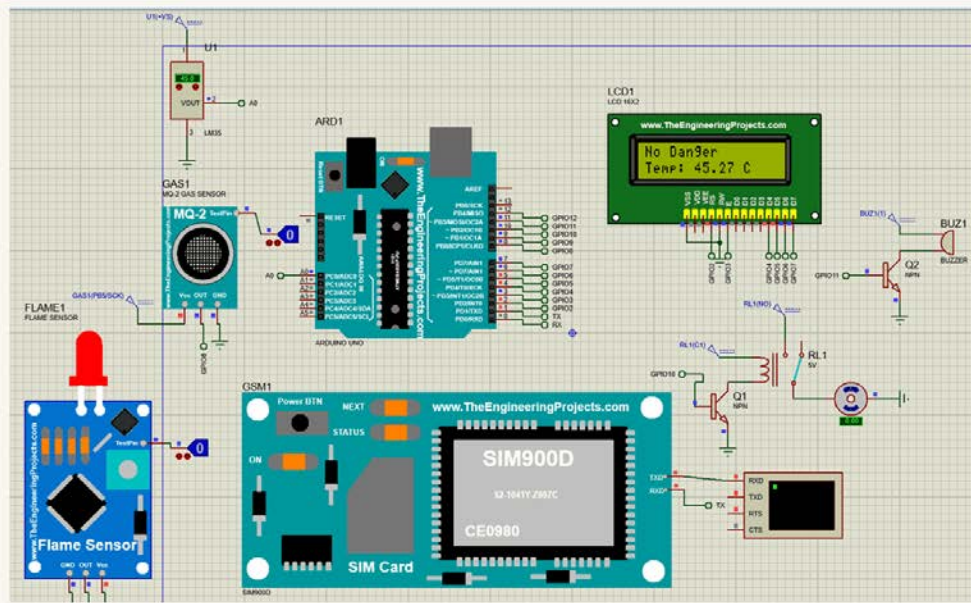


Рисунок Б.7 – Слайд 7



Рисунок Б.8 – Слайд 8

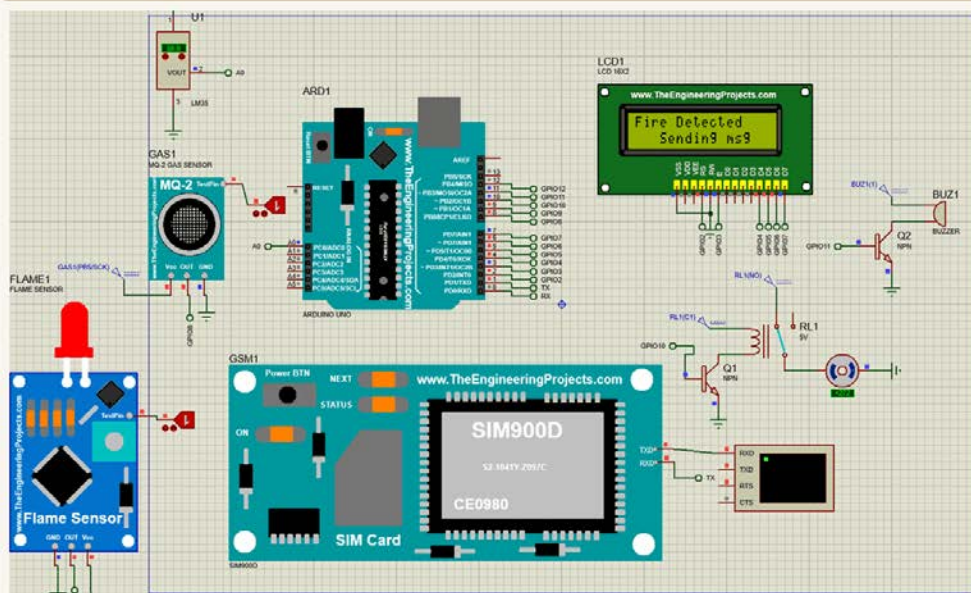
Модель підсистеми створена в Proteus 8 Professional



9

Рисунок Б.9 – Слайд 9

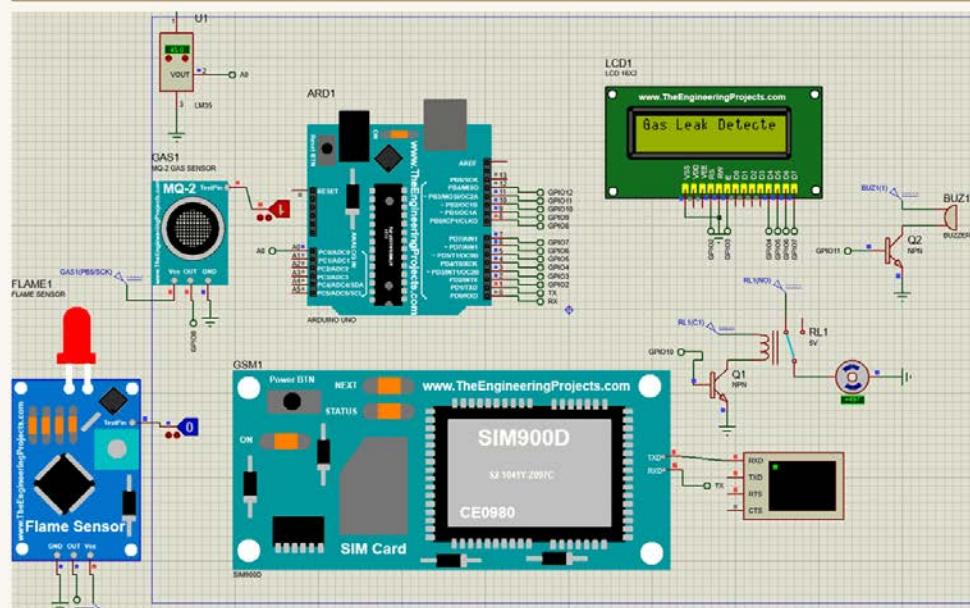
Моделювання роботи підсистеми під час пожежі



10

Рисунок Б.10 – Слайд 10

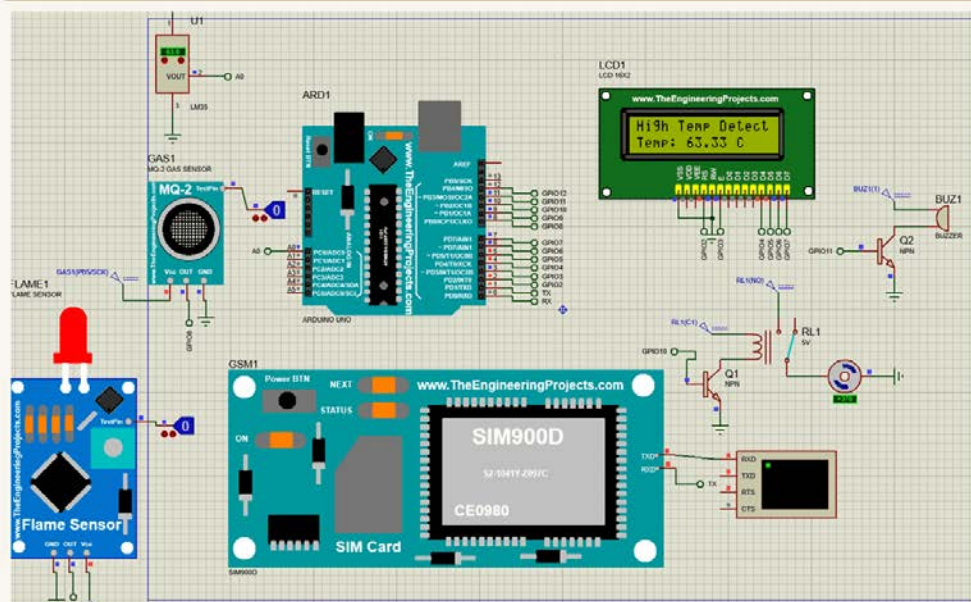
Модельовання роботи підсистеми під час витoku газу



11

Рисунок Б.11 – Слайд 11

Модельовання роботи підсистеми під час високої температури



12

Рисунок Б.12 – Слайд 12

Висновки

У дипломній роботі було розроблено ППБ для СРБ. ППБ РБ демонструє високу ефективність і надійність. Вона пропонує багаторівневий захист, який включає віддалене інформування користувача, автоматичні дії для гасіння пожежі та звукове сповіщення. Такі системи можуть значно підвищити безпеку мешканців і знизити ризики пожеж.

Подальший розвиток системи може включати інтеграцію з іншими підсистемами розумного будинку, наприклад, системою керування освітленням чи клімат-контролем для автоматичного реагування на надзвичайні ситуації. Також перспективним напрямком є розробка мобільного додатку для віддаленого моніторингу та управління станом пожежної безпеки.

Результати роботи були представлені на НПК «Тиждень науки-2024».



Рисунок Б.13 – Слайд 13