

Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра спеціалізованих комп'ютерних систем

ТОРІШНИЙ БОГДАН ВОЛОДИМИРОВИЧ

КОМП'ЮТЕРНО-ІНТЕГРОВАНА СИСТЕМА ЗАХИСТУ ВІД ПОЖЕЖ/ COMPUTER-
INTEGRATED FIRE PROTECTION SYSTEM

Спеціальність 151 - Автоматизація та комп'ютерно-інтегровані технології
Освітньо-професійна програма – Автоматизація та комп'ютерно-інтегровані
технології

Випускна кваліфікаційна робота за ступенем вищої освіти "бакалавр"

Виконав студент групи АКІТ-41
Вадим Грама

Науковий керівник:
к.т.н., доц. Н.Г. Яцків

Дипломну роботу допущено до захисту:

" ____ " _____ 20__ р.

Завідувач кафедри

А.І.Сергін

Тернопіль 2026

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	7
1 АНАЛІЗ КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ СИСТЕМ ПРОТИПОЖЕЖНОГО ЗАХИСТУ ОБ'ЄКТІВ	10
1.1 Функціональне призначення та задачі сучасних систем пожежної сигналізації	10
1.2 Аналіз архітектури типових систем моніторингу та протипожежної безпеки	12
1.3 Класифікація автоматичних засобів виявлення пожежі та методів їх інтеграції	14
1.4 Обґрунтування розробки інтелектуальної системи захисту з функцією зонального керування	19
2 ОБґРУНТУВАННЯ ТЕХНОЛОГІЇ ПРОЄКТУВАННЯ КОМП'ЮТЕРНО-ІНТЕГРОВАНОЇ СИСТЕМИ ПРОТИПОЖЕЖНОГО ЗАХИСТУ ТА РОЗРОБКА ФУНКЦІОНАЛЬНОЇ СХЕМИ АВТОМАТИЗАЦІЇ	24
2.1 Аналіз контрольованого об'єкта та визначення зон пожежної небезпеки	24
2.2 Розроблення функціональної схеми автоматизації системи	26
2.3 Обґрунтування вибору технічних засобів	30
2.3.1 Мікроконтролер ESP32 як центральний вузол обробки даних	30
2.3.2 Цифровий датчик температури та вологості DHT22	31
2.3.3 Аналоговий газо-димовий сенсор MQ-2	32
2.3.4 Релейний модуль для зонального керування електроживленням	33

2.3.5 Звуковий оповіщувач та аварійне освітлення	34
2.3.6 Хмарна платформа Blynk та Telegram-сповіщення	35
2.4 Розроблення алгоритму роботи системи	36
3 ПРОГРАМНО-АЛГОРИТМІЧНА РЕАЛІЗАЦІЯ ТА МОДЕЛЮВАННЯ СИСТЕМИ	39
3.1 Загальна структура програмної реалізації системи	39
3.2 Алгоритм обробки сигналів та визначення станів системи	42
3.3 Логіка зонального реагування та керування виконавчими пристроями	48
3.4 Реалізація контролю основного та резервного живлення	52
3.5 Моделювання системи у середовищі Wokwi	56
3.6 Перевірка роботи системи в основних сценаріях	60
3.7 Інтеграція з хмарною платформою Blynk та Telegram- сповіщеннями	79
ВИСНОВКИ	89
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	92
ДОДАТКИ	93

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підп.	Дата		

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АЦП – аналого-цифровий перетворювач

МК – мікроконтролер

ПЗ – програмне забезпечення

ППК – прилад приймально-контрольний

СПС – система пожежної сигналізації

СПЗ – система протипожежного захисту

API (Application Programming Interface) – інтерфейс прикладного програмування

GPIO (General Purpose Input/Output) – цифровий інтерфейс загального призначення (вхід/вихід)

HTTPS (Hypertext Transfer Protocol Secure) – захищений протокол передачі даних

IoT (Internet of Things) – Інтернет речей

LED (Light-Emitting Diode) – світлодіод

PPM (Parts Per Million) – одиниця вимірювання концентрації газу (частин на мільйон)

Wi-Fi (Wireless Fidelity) – технологія бездротового локального зв'язку

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підп.	Дата		

ВСТУП

Актуальність теми. Забезпечення пожежної безпеки об'єктів житлового, адміністративного та промислового фонду залишається одним із найгостріших питань техногенної безпеки. Згідно статистичних даних, значна частина пожеж виникає саме через несправність електромереж, перевантаження ліній та людський фактор у побутових зонах (кухні, підсобні приміщення). Традиційні системи пожежної сигналізації, які використовуються на більшості об'єктів, часто обмежуються локальним звуковим сповіщенням. Такі сповіщення є малоефективними коли відсутній персонал на місці або ж у нічний час.

У сучасному світі розвиток систем протипожежного захисту характеризується глобальним переходом від автономних дискретних пристроїв до складних комп'ютерно-інтегрованих рішень. Впровадження концепції «Інтернету речей» (IoT) дозволяє модернізувати пасивну систему сповіщення на активну систему захисту. Використання сучасних високопродуктивних мікроконтролерів, зокрема серії ESP32, надає умови для реалізації інтелектуальних алгоритмів обробки сигналів у реальному часі. А це, в свою чергу, мінімізує ймовірність хибних спрацювань, яка є однією з головних проблем сучасних систем безпеки.

Важливим є питання комплексного реагування на загрозу. Типові системи не мають зв'язку з енергосистемою будівлі. Тоді як запропонована робоча модель з зональним відключенням електроживлення, дозволяє автоматично локалізувати джерело загоряння, спричинене коротким замиканням або перегрівом електрощитової. Інтеграція з хмарними сервісами (Blynk) та месенджерами (Telegram) забезпечить миттєве інформування відповідальних осіб незалежно від того де вони знаходяться. Це є критично важливо для швидкої організації евакуації та пожежогасіння.

Додатковим аргументом на користь важливості запропонованої моделі є економічна доцільність. Ця доцільність полягає в тому, що дана розробка є ощадливою і в той же час функціональною системою за принципами

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підп.	Дата		

відкритості та сумісності. А це в свою чергу дозволить масово впроваджувати засоби автоматичного захисту навіть на об'єктах, де встановлення дорогих промислових систем є фінансово недоступним. Таким чином, розроблення комп'ютерно-інтегрованої системи, що поєднує моніторинг, автоматичне керування навантаженням та резервовані канали зв'язку, є вчасним та затребуваним кроком у напрямку цифровізації систем безпеки.

Мета роботи – створення проекту комп'ютерно-інтегрованої системи протипожежного захисту [1], яка забезпечує багаторівневий моніторинг середовища, автоматичне зонування аварійних ситуацій та дистанційне інформування користувача про стан об'єкта.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести аналіз існуючих методів та технічних засобів автоматичного виявлення ознак пожежі.
2. Обґрунтувати вибір апаратної платформи та сенсорного забезпечення для моніторингу трьох зон: житлового сектору, кухні та електрощитової.
3. Розробити алгоритм функціонування системи з урахуванням різних станів небезпеки («Норма», «Підозра», «Пожежа»).
4. Реалізувати механізм автоматичного захисного відключення електропостачання та активації аварійного освітлення.
5. Спроектувати програмне забезпечення для контролера ESP32 та налаштувати інтерфейс віддаленого доступу.

Об'єкт дослідження – процеси автоматизації моніторингу та захисту об'єктів від пожеж з використанням мікропроцесорної техніки.

Предмет дослідження – алгоритми обробки даних з сенсорів температури і диму та методи хмарної інтеграції систем безпеки.

Методи дослідження базуються на використанні теорії автоматичного керування, методів об'єктно-орієнтованого програмування та імітаційного моделювання в середовищі Wokwi.

Практичне значення одержаних результатів. Розроблена система може бути використана як ощадливе та продуктивне рішення для модернізації

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підп.	Дата		

систем пожежної безпеки в існуючих житлових та офісних приміщеннях без необхідності прокладання складних кабельних мереж. Використання відкритої платформи ESP32 та хмарних сервісів дозволяє легко масштабувати систему та інтегрувати її в існуючі концепції «Розумного будинку». Основні результати роботи можуть бути впроваджені при проектуванні автоматизованих вузлів контролю електрощитових та кухонних зон для підвищення рівня техногенної безпеки.

Напрямки подальшого розвитку. У перспективі модернізація системи планується за рахунок: впровадження алгоритмів штучного інтелекту для попередження пожежі на основі аналізу зміни показників сенсорів та засобів самодіагностики датчиків для виявлення їх несправності або забруднення, налаштування єдиної системи з автономними модулями пожежогасіння (наприклад, генераторами вогнегасного аерозолю) та розширення списку газів, що підлягають моніторингу (чадний газ CO, природний газ).

Апробація. Богдан Торішній. Зональне керування та хмарне сповіщення в системі протипожежного захисту на базі ESP32 // Збірник матеріалів всеукраїнської науково-практичної конференції «Автоматизація та комп'ютерно-інтегровані технології» (АКІТ - 2026), Тернопіль, 2026. -С.182-186.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підп.	Дата		

1 АНАЛІЗ КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ СИСТЕМ ПРОТИПОЖЕЖНОГО ЗАХИСТУ ОБ'ЄКТІВ

1.1 Функціональне призначення та задачі сучасних систем пожежної сигналізації.

У сучасних умовах стрімкого розвитку промислової та житлової інфраструктури питання забезпечення техногенної безпеки набувають особливого значення. Сучасна СПС (система пожежної сигналізації)[2] сьогодні розглядається не як автономний пристрій детектування, а як складна багаторівнева комп'ютерно-інтегрована система, що функціонує в єдиному інформаційному просторі об'єкта. Основним функціональним призначенням такої системи є безперервний моніторинг середовища з метою ідентифікації небезпечних факторів пожежі на етапі їх зародження та автоматизація комплексних заходів із захисту людей і матеріальних цінностей.

На відміну від застарілих технічних рішень, архітектура яких базувалася на простій релейній логіці, сучасна комп'ютерно-інтегрована СПС забезпечує автоматизоване керування інженерною інфраструктурою. Це дозволяє не лише констатувати факт виникнення небезпеки, а й активно впливати на стан об'єкта для локалізації осередку загоряння та запобігання поширенню вогню через суміжні комунікації.

До основних функціональних задач, що покладаються на сучасні ТЗ (технічні засоби) протипожежного захисту, належать наступні:

Багатозонний моніторинг та аналіз динаміки змін середовища. Алгоритм функціонування системи передбачає організацію безперервного збору даних з розгалуженої сенсорної мережі. Контроль здійснюється у чітко визначених критичних зонах (кухня, електрощитова, житлові приміщення), де імовірність теплового перевантаження або короткого замикання є найвищою. Використання сучасних сповіщувачів дозволяє аналізувати не лише порогові значення, а й динаміку зміни температури та концентрації продуктів згоряння. Це забезпечує можливість фіксації аномалій ще на стадії термічного

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підп.	Дата		

розкладання матеріалів (тління), що є фундаментальною умовою для мінімізації наслідків пожежі.

Пріоритетність передачі сигналів та багаторівневе оповіщення. Однією з найважливіших задач СПС є організація надійного каналу інформування. У разі ідентифікації загрози система виконує пріоритетну трансляцію тривожних сповіщень. Це реалізується шляхом активації локальних звукових і світлових пристроїв оповіщення на об'єкті, а також через цифрові канали зв'язку (хмарні сервіси, месенджери) як додатковий канал інформування користувача. Використання дубльованих протоколів передачі даних гарантує отримання інформації відповідальними особами навіть у разі пошкодження локальної мережевої інфраструктури об'єкта.

Автоматизація захисних дій та селективне керування навантаженням. Комп'ютерна інтеграція системи дозволяє реалізувати складні сценарії реагування через виконавчі пристрої. Ключовою задачею в даному контексті є селективне (зональне) відключення електроживлення безпосередньо в осередку небезпеки. Це запобігає виникненню вторинних факторів пожежі, таких як вибухи електрообладнання або поширення вогню через пошкоджену ізоляцію кабелів. Паралельно з цим система забезпечує автоматичну активацію аварійного аварійне освітлення, що є необхідним для безпечної евакуації персоналу або мешканців у сутінках або за умови сильного задимлення.

Інтелектуальна обробка даних та фільтрація хибних тривог. Використання обчислювальних потужностей сучасних мікроконтролерів (на базі яких будується ППК) дозволяє впроваджувати алгоритми цифрової фільтрації сигналів. Це дає змогу системі диференціювати реальні загрози від короточасних коливань параметрів, спричинених побутовими процесами (наприклад, приготування їжі на кухні). Завдяки багаторівневій оцінці стану об'єкта («Норма», «Підозра», «Пожежа») суттєво знижується ймовірність виникнення хибні спрацювання системи, що підвищує загальну довіру до системи безпеки.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підп.	Дата		

Самодіагностування та забезпечення відмовостійкості. Важливою функцією є постійний контроль цілісності ліній зв'язку та працездатності сенсорного обладнання. Система виконує періодичне тестування всіх вузлів і за наявності несправностей миттєво інформує про це обслуговуючий персонал. Крім того, архітектура системи передбачає обов'язковий моніторинг стану джерел живлення з можливістю автоматичного переходу на резервний акумуляторний блок у разі деградації або повного зникнення напруги в основній мережі.

Технічна реалізація на базі сучасних обчислювальних платформ. В межах даного дослідження розглядається концепція побудови системи на базі контролера ESP32 [5], що виконує функції центрального вузла обробки даних (аналогічні ППК). Вибір такої платформи обумовлений необхідністю поєднання високої обчислювальної потужності для виконання логічних операцій «датчик-реакція» з вбудованою підтримкою мережевих протоколів для хмарної інтеграції. Такий підхід відповідає актуальним тенденціям цифровізації та дозволяє створювати гнучкі, масштабовані системи захисту, адаптовані під специфічні вимоги конкретного об'єкта автоматизації.

1.2 Аналіз архітектури типових систем моніторингу та протипожежної безпеки.

Архітектура будь-якої СПС визначає надійність, швидкість реакції та складність технічного обслуговування об'єкта. На сучасному етапі розвитку автоматизації виділяють два основних типи побудови архітектури: традиційні провідні ієрархічні системи та сучасні комп'ютерно-інтегровані мережеві структури на базі концепції «Інтернету речей» (IoT).

Традиційна архітектура типової системи пожежної безпеки зазвичай базується на централізованому принципі. Основним вузлом такої структури виступає прилад приймально-контрольний (ППК)[3], до якого за допомогою дротових шлейфів підключаються сповіщувачі. Така побудова має суттєвий недолік: обмежену гнучкість та складність масштабування, особливо в умовах

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підп.	Дата		

уже експлуатованих будівель, де прокладання нових кабельних трас є економічно недоцільним або технічно ускладненим.

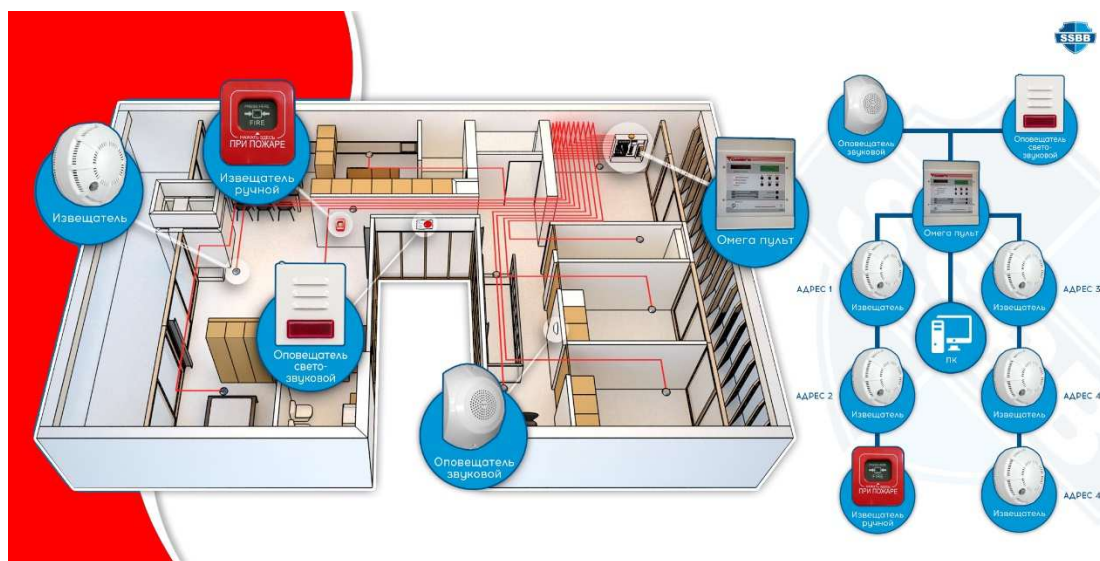


Рисунок 1.1 - Архітектура типової пожежної системи безпеки

У межах даного дослідження розглядається перехід до комп'ютерно-інтегрованої архітектури, яка складається з наступних основних рівнів:

-Рівень збору первинної інформації (Сенсорний рівень)

На цьому рівні функціонують сенсорні елементи та засоби виявлення - датчики температури, диму та, за необхідності, газу. У сучасних системах замість простих порогових сповіщувачів усе частіше застосовуються інтелектуальні сенсори, здатні передавати не лише сигнал «Тривога», а й конкретні числові значення параметрів середовища. Це дозволяє центральному контролеру будувати тренди змін та виявляти загрозу ще на етапі її зародження.

-Рівень обробки та прийняття рішень (Логічний рівень)

Ядром системи виступає мікропроцесорний модуль (у даному випадку - контролер ESP32), який виконує функції центрального вузла обробки даних, аналогічні ППК. Архітектура цього рівня передбачає програмовану логіку, що дозволяє реалізувати алгоритми фільтрації шумів та перевірки достовірності сигналів. Використання мікроконтролерів із вбудованими модулями бездротового зв'язку дозволяє частково зменшити залежність від громіздкої

провідної інфраструктури та реалізувати більш гнучкі топології побудови системи.

-Рівень виконавчих механізмів (Рівень реагування)

Цей рівень включає блоки реле, звукові оповісники (сирени) та системи аварійного освітлення. Особливістю сучасної архітектури є можливість зонального керування - система може автономно знеструмити окрему кімнату або вузол (наприклад, електрощитову), не порушуючи роботу інших ділянок об'єкта, що підвищує загальну живучість системи.

-Інформаційно-комунікаційний рівень (Рівень візуалізації)

Це верхній рівень архітектури, який забезпечує взаємодію між технічними засобами та кінцевим користувачем. Завдяки використанню протоколів Wi-Fi та хмарних платформ (наприклад, Blynk [13]) архітектура стає більш прозорою: дані про стан кожної зони доступні у режимі реального часу на мобільних пристроях. Додаткове інформування користувача через месенджери (наприклад, Telegram [14]) дозволяє оперативно отримувати сповіщення про стан системи навіть поза межами об'єкта.

Таким чином, аналіз показує, що сучасна архітектура пожежного моніторингу еволюціонує у напрямку поєднання централізованих та комп'ютерно-інтегрованих підходів. Це дозволяє створювати системи, які є не лише засобами сигналізації, а й активними елементами керування безпекою, що поєднують функції вимірювання, логічної обробки та дистанційного моніторингу.

1.3 Класифікація автоматичних засобів виявлення пожежі та методів їх інтеграції.

Ефективність функціонування будь-якої комп'ютерно-інтегрованої системи протипожежного захисту значною мірою залежить від правильного вибору, розміщення та способу підключення засобів виявлення пожежі. Саме сенсорна частина системи забезпечує первинне отримання інформації про стан

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підп.	Дата		

середовища, тому від її точності та стабільності залежить швидкість реагування всієї системи.

У сучасних системах пожежної сигналізації основну роль виконують пожежні сповіщувачі, які класифікуються залежно від фізичного параметра, на який вони реагують. До таких параметрів належать температура, наявність диму, продуктів згоряння, полум'я або небезпечних газів. У межах даної роботи основна увага приділяється тепловому та димовому каналам виявлення, оскільки саме вони найбільш доцільні для моделювання системи захисту житлового приміщення, кухні та електрощитової.

Згідно з методами детектування, основні технічні засоби виявлення пожежі можна поділити на такі групи:

Теплові пожежні сповіщувачі:



Рисунок 1.2 - Сповіщувач тепловий пожежний

Теплові пожежні сповіщувачі реагують на зміну температури навколишнього середовища. Вони застосовуються у приміщеннях, де підвищення температури є одним із головних показників виникнення небезпечної ситуації, наприклад у електрощитових, технічних приміщеннях або зонах із підвищеним електричним навантаженням.

За принципом роботи теплові сповіщувачі поділяються на:

- максимальні - спрацьовують при досягненні заданого температурного порогу;

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підп.	Дата		

- диференційні - реагують на швидкість зростання температури;
- максимально-диференційні - поєднують обидва принципи роботи.

У даній роботі для моделювання теплового каналу використовуються цифрові сенсори температури типу DHT22 [9]. Вони не є сертифікованими пожежними сповіщувачами, однак у межах навчально-дослідного прототипу дозволяють отримувати числові значення температури та передавати їх на центральний контролер для подальшої обробки. Завдяки цьому система може не лише контролювати перевищення встановленого порогу, а й аналізувати динаміку нагрівання у певній зоні.

Такий підхід є особливо важливим для електрощитової, де небезпечна ситуація може розвиватися поступово через перегрів проводки, перевантаження електричних ліній або несправність обладнання. У цьому випадку аналіз швидкості зростання температури дозволяє виявити потенційну загрозу ще до появи відкритого полум'я.

Димові пожежні сповіщувачі:



Рисунок 1.3 - Сповіщувач димовий пожежний

Димові пожежні сповіщувачі [4] призначені для виявлення продуктів згоряння у повітрі. Вони є одними з найпоширеніших засобів раннього виявлення пожежі, оскільки в багатьох випадках займання супроводжується появою диму ще до значного підвищення температури.

Найбільш поширеними є оптико-електронні димові сповіщувачі, які реагують на наявність аерозольних частинок у повітрі. Такі пристрої ефективні для житлових кімнат, коридорів та інших приміщень, де раннє виявлення задимлення має критичне значення для своєчасного оповіщення людей.

У межах розроблюваного навчально-дослідного прототипу імітація димового каналу реалізується за допомогою аналогових сенсорів типу MQ-2 [11]. Такий сенсор може реагувати не лише на дим, а й на деякі горючі гази, зокрема пропан, бутан, водень та інші леткі речовини. Це дозволяє розширити функціональні можливості системи та використати її для додаткового контролю небезпечних факторів у зоні кухні.

Водночас слід зазначити, що MQ-2 не є повноцінним сертифікованим пожежним димовим сповіщувачем. У даній роботі він застосовується як сенсорний елемент для моделювання каналу виявлення диму та газоподібних продуктів, що дозволяє дослідити принцип роботи системи в умовах зміни параметрів середовища.

Комбіновані та інтелектуальні засоби виявлення:



Рисунок 1.4 - Димо-тепловий сповіщувач

Сучасні тенденції розвитку систем пожежної сигналізації передбачають використання комбінованих засобів виявлення, які поєднують кілька каналів контролю. Наприклад, один пристрій може одночасно аналізувати рівень

задимленості, температуру та швидкість її зміни. Такий підхід дозволяє підвищити достовірність виявлення пожежі та зменшити кількість хибних спрацювань.

У межах даної роботи комбінований принцип реалізується не на рівні окремого промислового сповіщувача, а програмно - за допомогою мікроконтролера ESP32. Контролер отримує дані з різних сенсорів, порівнює їх із заданими пороговими значеннями та формує загальний стан контрольованої зони.

Для цього використовується багаторівнева логіка оцінювання стану системи, наприклад:

- «Норма» - параметри середовища перебувають у допустимих межах;
- «Підозра» - зафіксовано незначне або короткочасне відхилення одного з параметрів;
- «Пожежа» - виявлено стійке перевищення критичних параметрів або одночасну появу кількох ознак небезпеки.

Такий підхід дозволяє системі враховувати специфіку різних приміщень. Наприклад, для кухні короткочасне підвищення рівня диму або пари не завжди свідчить про пожежу, тому доцільно враховувати додаткові параметри: температуру, тривалість перевищення порогу та динаміку зміни показників. Для електрощитової, навпаки, навіть поступове стабільне зростання температури може бути ознакою небезпечного перегріву, тому для цієї зони пороги реагування мають бути більш чутливими.

Завдяки програмному поєднанню даних із різних каналів система може більш точно визначати характер небезпеки та знижувати ймовірність хибних спрацювань, не втрачаючи здатності до раннього виявлення реальної загрози.

Методи інтеграції сповіщувачів у систему:

Інтеграція засобів виявлення у загальну структуру комп'ютерно-інтегрованої системи здійснюється за кількома основними принципами.

Аналогова інтеграція передбачає зчитування безперервного або умовно безперервного сигналу з сенсорів, який надалі перетворюється у цифрове

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підп.	Дата		

значення та обробляється контролером. Такий підхід використовується, зокрема, для сенсорів типу MQ-2. Його перевагою є можливість аналізувати не лише сам факт перевищення порогу, а й характер зміни сигналу в часі.

Цифрова інтеграція використовується для сенсорів, які передають дані у вигляді готових цифрових значень. Прикладом є сенсор DHT22, який передає інформацію про температуру та вологість. Це спрощує підключення до мікроконтролера та дозволяє безпосередньо використовувати отримані значення в алгоритмах обробки.

Програмно-логічна інтеграція полягає в об'єднанні даних з різних зон у єдину логічну модель стану системи. Завдяки цьому кожен сенсор розглядається не як окремий ізольований пристрій, а як частина загальної системи моніторингу. Наприклад, перегрів у електрощитовій може стати підставою для зонального відключення електроживлення, а задимлення у житловій кімнаті - для активації сирени та аварійного освітлення.

Таким чином, правильна класифікація та вибір методів інтеграції засобів виявлення дозволяють створити систему, адаптовану до специфіки кожної контрольованої зони. Поєднання теплового, димового та програмно-логічного каналів контролю забезпечує більш достовірне виявлення небезпечних ситуацій, зменшує кількість хибних спрацювань і підвищує ефективність реагування системи на ранніх стадіях виникнення пожежі.

1.4 Обґрунтування розробки інтелектуальної системи захисту з функцією зонального керування.

Проведений аналіз існуючих методів та технічних засобів протипожежного захисту дозволяє зробити висновок, що, незважаючи на широке різноманіття рішень, питання створення гнучких, функціональних та економічно доступних систем для об'єктів малого й середнього масштабу залишається актуальним. Більшість сертифікованих промислових СПС характеризуються високою надійністю та відповідністю нормативним вимогам, однак їх впровадження може супроводжуватися значною вартістю,

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підп.	Дата		

ESP32, та хмарної платформи Blynk дозволяє створити зручний інтерфейс дистанційного моніторингу.

У такій системі користувач отримує не лише повідомлення про тривогу, а й доступ до поточних значень контрольованих параметрів: температури, рівня задимленості або стану окремих зон. Це дає змогу оцінювати ситуацію в режимі реального часу та приймати більш обґрунтовані рішення дистанційно. Додаткове інформування через месенджери, наприклад Telegram, може використовуватися як допоміжний канал сповіщення користувача про зміну стану системи.

3. Впровадження алгоритмів обробки даних (стани загрози)

Однією з поширених проблем систем моніторингу є хибні спрацювання, які можуть виникати через короткочасні коливання параметрів середовища, побутові фактори або зовнішні перешкоди. Тому доцільним є застосування багаторівневої логіки оцінювання ситуації.

У межах розроблюваної системи передбачається використання трьох основних станів:

- «Норма» - параметри середовища перебувають у допустимих межах;
- «Підозра» - зафіксовано відхилення одного або кількох параметрів, однак умови для автоматичного реагування ще не досягнуті;
- «Пожежа» - виявлено стійке перевищення критичних параметрів або одночасну появу кількох ознак небезпеки.

Така програмна фільтрація сигналів, реалізована на рівні центрального вузла обробки даних, може підвищити достовірність роботи системи та зменшити кількість хибних спрацювань. Крім того, наявність проміжного стану «Підозра» дозволяє попередньо інформувати користувача без негайного відключення живлення або активації повного сценарію реагування.

4. Економічна ефективність та здатність до розширення

Система побудована на базі доступних деталей та відкритих стандартів. Це значно здешевлює розробку навчального прототипу. Такий підхід ідеально підходить для демонстрації «розумного» пожежного моніторингу в квартирах, невеликих офісах та інших малих об'єктах.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підп.	Дата		

Завдяки модульній структурі можна легко збільшувати кількість зон контролю. При цьому не потрібно переробляти всю систему з нуля. До системи швидко підключаються нові датчики, модулі керування або способи оповіщення користувачів. Таке рішення дозволяє легко налаштувати під потреби та особливості конкретного приміщення.

5. Адаптація алгоритмів під різні зони об'єкта

Головною перевагою розроблюваної системи є можливість легко налаштувати під конкретне приміщення. Це дозволяє задавати різні правила реагування для кожної зони. У різних приміщеннях однакові значення температури, рівня задимленості або газу можуть мати різне значення з точки зору безпеки. Тому використання єдиних порогових значень для всього об'єкта не завжди є доцільним.

Наприклад, для кухні короточасне підвищення температури або рівня задимленості може бути пов'язане з нормальними побутовими процесами, такими як приготування їжі або утворення пари. У такому випадку система не повинна одразу переходити в аварійний режим, а має враховувати тривалість відхилення, динаміку зміни параметрів та наявність додаткових ознак безпеки.

Для електрощитової підхід має бути більш чутливим, оскільки навіть поступове стабільне зростання температури може свідчити про перевантаження лінії, перегрів контактів або несправність електрообладнання. У такій зоні доцільно встановлювати нижчі пороги попередження та передбачати швидший перехід до стану «Підозра». Це дозволяє адаптувати систему до особливостей кожного приміщення та підвищити точність реагування.

6. Підвищення інформативності системи для користувача

Ще одним важливим аргументом на користь розробки комп'ютерно-інтегрованої системи є підвищення інформативності процесу моніторингу. У традиційних системах користувач часто отримує лише загальний сигнал тривоги, без детального розуміння того, у якій саме зоні виникла проблема та які параметри стали причиною спрацювання.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підп.	Дата		

У запропонованому рішенні передбачається можливість відображення стану кожної контрольованої зони окремо. Користувач може бачити поточні значення температури, рівня задимленості, стан живлення, активність виконавчих пристроїв та загальний статус системи. Такий підхід дозволяє швидше оцінити характер ситуації та прийняти більш обґрунтоване рішення.

Крім того, така детальна інформація може бути корисною не лише під час аварійної ситуації, а й у процесі профілактичного контролю. Наприклад, регулярне підвищення температури в електрощитовій або повторювані спрацювання в одній і тій самій зоні можуть свідчити про наявність технічної проблеми, яку доцільно усунути ще до виникнення реальної небезпеки. Таким чином, система виконує не лише функцію реагування, а й допомагає здійснювати попереджувальний моніторинг стану об'єкта.

Таким чином, розробка комп'ютерно-інтегрованої системи захисту з функцією зонального керування є обґрунтованою з технічної та практичної точки зору. Запропонований підхід поєднує моніторинг параметрів середовища, алгоритмічну обробку даних, дистанційне інформування та автоматизоване керування окремими зонами об'єкта. Це дозволяє дослідити принципи побудови гнучких систем протипожежного моніторингу та реагування на базі сучасних мікроконтролерних платформ.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підп.	Дата		

2 ОБҐРУНТУВАННЯ ТЕХНОЛОГІЇ ПРОЄКТУВАННЯ КОМП'ЮТЕРНО-ІНТЕГРОВАНОЇ СИСТЕМИ ПРОТИПОЖЕЖНОГО ЗАХИСТУ ТА РОЗРОБКА ФУНКЦІОНАЛЬНОЇ СХЕМИ АВТОМАТИЗАЦІЇ

2.1 Аналіз контрольованого об'єкта та визначення зон пожежної небезпеки.

У процесі проєктування систем протипожежного захисту важливим етапом є аналіз об'єкта, для якого створюється система. Перед вибором датчиків, виконавчих пристроїв та алгоритмів реагування необхідно визначити, які саме зони потребують контролю, які небезпечні фактори можуть у них виникати та які дії система повинна виконувати у разі виявлення загрози.

Поділ об'єкта на окремі зони потрібен для того, щоб система могла точніше реагувати. Небезпека в різних приміщеннях може проявлятися по-різному, тому однаковий алгоритм для всього об'єкта не завжди буде правильним. Наприклад, для кухні, житлової кімнати та електрощитової варто передбачати різні умови спрацювання, оскільки джерела пожежної небезпеки в цих зонах трохи відрізняються, як і пороги спрацювань датчиків.

У даній роботі розглядається комп'ютерно-інтегрована система протипожежного захисту, яка призначена для контролю кількох умовних зон об'єкта. Такий підхід дозволяє системі не лише виявляти загальні ознаки пожежі, а й визначати, у якій саме зоні виникла небезпека, які параметри змінилися та які дії необхідно виконати. Це є основою для реалізації зонального керування, аварійного освітлення, локального оповіщення та дистанційного інформування користувача.

У межах розроблюваної системи виділено такі основні контрольовані зони:

- кухня;
- житлова кімната;

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підп.	Дата		

- електрощитова;
- контрольовані електричні точки підвищеного ризику (розетки).

Кухня є однією з найбільш пожежонебезпечних зон, оскільки в ній можуть одночасно бути присутні нагрівальні прилади, електричне навантаження, газове обладнання або горючі матеріали. Для цієї зони важливим є контроль температури, рівня задимленості та наявності газу. При цьому система повинна враховувати, що короткочасна поява пари або незначного задимлення може бути пов'язана з нормальними побутовими процесами, наприклад приготуванням їжі. Тому для кухні доцільно використовувати не миттєве аварійне реагування, а багаторівневу оцінку стану.

Житлова кімната також потребує постійного контролю, оскільки в ній можуть розміщуватися електроприлади, зарядні пристрої, подовжувачі або інші джерела потенційної небезпеки. Для цієї зони основну увагу доцільно приділити контролю температури та задимленості. У разі виявлення небезпечного стану система повинна активувати локальне оповіщення, аварійне освітлення та передати повідомлення користувачу.

Електрощитова є критично важливою зоною, оскільки саме в ній зосереджені елементи розподілу електроживлення. Основними небезпечними факторами тут є перегрів контактів, перевантаження ліній, коротке замикання або несправність електрообладнання. Навіть поступове зростання температури в цій зоні може свідчити про розвиток аварійної ситуації, тому для електрощитової доцільно встановлювати більш чутливі умови реагування.

Окрему увагу в системі приділено контролю контрольовані електричні точки підвищеного ризику (розетки). У межах моделі такі ділянки розглядаються як потенційні місця виникнення короткого замикання або локального перегріву. Для їх контролю можуть використовуватися температурні сенсори, розміщені поблизу відповідних електричних вузлів. Якщо температура біля розетки перевищує допустиме

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підп.	Дата		

значення, система може сформувати попередження або виконати зональне відключення живлення.



Рисунок 2.1 - Електрощитова будинку

Таким чином, поділ об'єкта на окремі контрольовані зони дозволяє реалізувати більш точну та гнучку логіку роботи системи. Замість загального реагування на всю будівлю система аналізує, де саме виникла небезпека, який характер має відхилення та які дії необхідно виконати в конкретній ситуації, що підвищує ефективність протипожежного захисту та дозволяє зменшити кількість необґрунтованих спрацювань.

2.2 Розроблення функціональної схеми автоматизації системи.

Функціональна схема автоматизації відображає загальну будову розроблюваної системи та показує взаємозв'язок між її основними елементами. За допомогою такої схеми можна зрозуміти, які сенсори використовуються для отримання інформації, який пристрій обробляє ці

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підп.	Дата		

дані та які виконавчі елементи активуються у разі виникнення небезпечної ситуації.

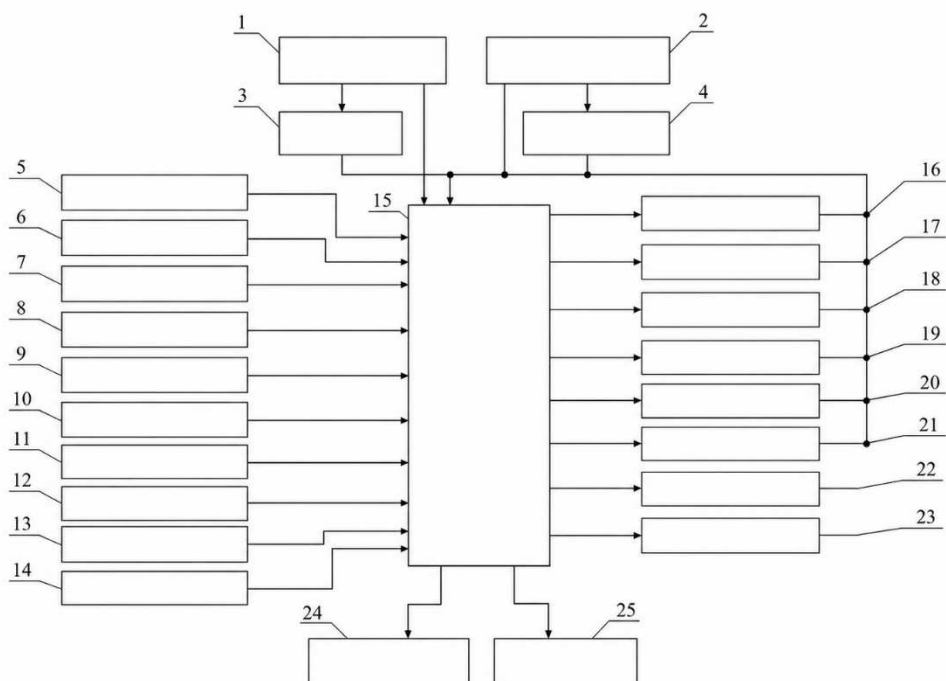


Рисунок 2.2 – Функціональна схема системи автоматизації

У даній роботі система побудована за принципом: сенсори → мікроконтролер ESP32 → обробка даних → виконавчі пристрої та дистанційне сповіщення користувача. Така структура дозволяє поєднати локальне реагування на об'єкти з можливістю віддаленого контролю через хмарну платформу Blynk та Telegram-сповіщення.

Функціональна схема системи автоматизації наведена на рисунку 2.2. Центральним елементом схеми є мікроконтролер ESP32 (15), який виконує функції основного вузла обробки даних і керування. До нього надходять сигнали від сенсорів температури, каналів диму та газу, а також інформація про стан основного живлення. Після обробки цих даних ESP32 формує керуючі сигнали для світлової індикації, аварійного освітлення, сирени, імітації зонального відключення живлення, а також передає дані до Blynk і Telegram.

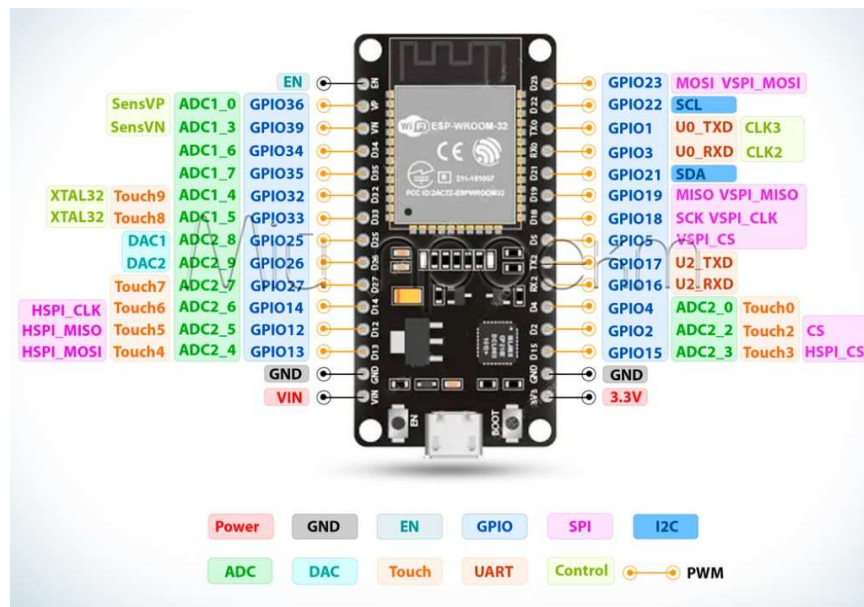


Рисунок 2.3 - Мікроконтролер ESP32

У верхній частині схеми показано систему живлення. Основне живлення 230 В (1) використовується як штатне джерело енергії для об'єкта. Для контролю його наявності передбачено окремий вхід (3), сигнал з якого надходить до мікроконтролера. Також у схемі враховано резервне живлення від акумулятора (2) та індикацію його роботи (4). У межах моделі резервне живлення показано функціонально, тобто як режим, у якому система продовжує працювати після втрати основного живлення.

Ліва частина функціональної схеми відповідає за збір первинної інформації. Для контролю температури використовуються сенсори DHT22 у житловій кімнаті (5), кухні (6), електрощитовій (7), біля розетки кімнати (8) та біля розетки кухні (9). Такий набір сенсорів дозволяє контролювати не тільки загальну температуру в приміщеннях, а й локальний перегрів у контрольованих електричних точках підвищеного ризику.

Окремо на схемі показано канали контролю диму та газу. Канал диму кімнати (10), канал диму кухні (11) та канал газу кухні (12) у середовищі Wokwi реалізуються за допомогою потенціометрів. Вони імітують зміну аналогового сигналу від відповідних датчиків, що дозволяє перевіряти реакцію системи на режими WARNING і FIRE.

Після отримання даних ESP32 порівнює їх із заданими пороговими значеннями та визначає стан кожної контрольованої зони. У роботі системи передбачено три основні стани: «Норма», «Підозра» та «Пожежа». У стані «Норма» всі параметри перебувають у допустимих межах. У стані «Підозра» система формує попередження, але ще не активує повний аварійний сценарій. Якщо параметри досягають критичного рівня, система переходить у стан «Пожежа» та виконує аварійне реагування.

Права частина схеми відображає виконавчі елементи системи. До них належать сирена (16), основне освітлення кімнати (17), основне освітлення кухні (18), аварійне освітлення кімнати (19), аварійне освітлення кухні (20) та загальне аварійне освітлення (21). У разі виникнення небезпечної ситуації ESP32 може вимкнути основне освітлення відповідної зони, увімкнути аварійне освітлення та активувати сирену.

Окремо на схемі показано сигнали відключення кімнати (22) та кухні (23). У середовищі Wokwi ці елементи реалізовано за допомогою LED-індикаторів, які імітують роботу релейних модулів. Тобто світлодіод не відключає реальне навантаження, а показує факт формування мікроконтролером команди на зональне відключення. У реальній системі ці сигнали могли б подаватися на релейні модулі, контактори або інші комутаційні пристрої.

Нижня частина схеми відображає засоби дистанційного моніторингу та сповіщення. Хмарна платформа Blynk (24) використовується для відображення поточних значень датчиків, станів зон, стану живлення та актуальних повідомлень системи. Telegram-сповіщення (25) застосовуються як додатковий канал оперативного інформування користувача про важливі події, зокрема втрату основного живлення, перехід у стан «Підозра» або «Пожежа», витік газу чи аварійний стан в електрощитовій.

Важливо, що дистанційні сервіси не замінюють локальну логіку роботи системи. Основні рішення приймаються безпосередньо мікроконтролером ESP32. Навіть якщо інтернет-з'єднання буде недоступним, система зможе продовжувати локальний моніторинг,

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підп.	Дата		

активувати аварійне освітлення, сирену та формувати сигнали відключення відповідних зон. Blynk і Telegram у цьому випадку виконують роль додаткових засобів контролю та інформування.

2.3 Обґрунтування вибору технічних засобів.

Вибір технічних засобів є важливим етапом розроблення комп’ютерно-інтегрованої системи протипожежного захисту. Компоненти системи повинні забезпечувати отримання даних із контрольованих зон, їх обробку, формування керуючих сигналів та передачу інформації користувачу. При цьому для навчально-дослідного прототипу важливо, щоб обрані елементи були доступними, простими для підключення та придатними для моделювання роботи системи в середовищі Wokwi [15].

У даній роботі технічні засоби підбиралися з урахуванням функцій, які повинна виконувати система: контроль температури, виявлення задимленості або газу, зональне відключення живлення, локальне оповіщення, аварійне освітлення та дистанційне інформування користувача.

2.3.1 Мікроконтролер ESP32 як центральний вузол обробки даних

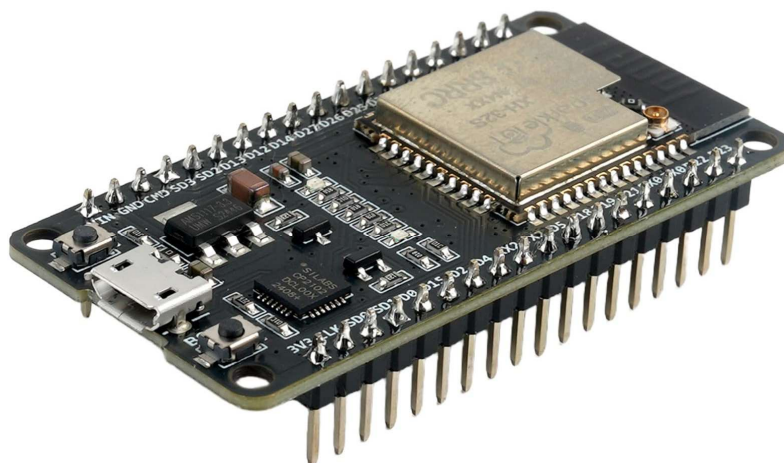


Рисунок 2.4 - Мікроконтролер ESP32

Основним елементом системи обрано мікроконтролер ESP32. Він виконує функції центрального вузла обробки даних: отримує сигнали від

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підп.	Дата		

сенсорів, аналізує їх відповідно до заданих умов та формує команди для виконавчих пристроїв.

Вибір ESP32 обґрунтований тим, що цей контролер має достатню кількість входів і виходів для підключення датчиків, реле, індикаторів, сирени та аварійного освітлення. Також важливою перевагою є наявність вбудованого Wi-Fi модуля, що дозволяє реалізувати дистанційний моніторинг через Blynk та надсилання повідомлень через Telegram.

У межах даної роботи ESP32 не розглядається як промисловий сертифікований прилад пожежної сигналізації, а використовується як центральний мікроконтролер навчально-дослідного прототипу. Його застосування дозволяє перевірити логіку роботи системи, реалізувати багаторівневу оцінку станів «Норма», «Підозра» та «Пожежа», а також організувати керування окремими зонами об'єкта.

2.3.2 Цифровий датчик температури та вологості DHT22



Рисунок 2.5 - Датчик температури та вологості DHT22

Для контролю температури у контрольованих зонах використовується цифровий датчик DHT22 [18]. Він дозволяє отримувати числові значення температури та передавати їх на ESP32 для подальшої обробки.

У системі DHT22 використовується для моделювання теплового каналу виявлення небезпеки. За його допомогою можна відстежувати не лише перевищення встановленого температурного порогу, а й зміну температури в часі. Це важливо, оскільки поступове або швидке зростання температури може свідчити про перегрів електрообладнання, розетки або електрощитової.

Окремо слід зазначити, що DHT22 не є сертифікованим тепловим пожежним сповіщувачем. У даній роботі він використовується як доступний сенсорний елемент для навчального моделювання роботи системи протипожежного моніторингу.

2.3.3 Аналоговий газо-димовий сенсор MQ-2



Рисунок 2.6 - Аналоговий газо-димовий сенсор MQ-2

Для імітації димового та газового каналу використовується аналоговий сенсор MQ-2. Він реагує на наявність диму та деяких горючих газів, тому підходить для моделювання небезпечних ситуацій у зоні кухні та житлового приміщення.

Перевагою MQ-2 є те, що він передає аналоговий сигнал, рівень якого змінюється залежно від концентрації газів або диму. Це дозволяє ESP32 не просто отримувати дискретний сигнал «є/немає небезпеки», а аналізувати рівень відхилення та порівнювати його з пороговими значеннями [17].

У межах даної роботи MQ-2 використовується не як сертифікований димовий пожежний сповіщувач, а як сенсор для моделювання задимленості та витоку газу.

2.3.4 Релейний модуль для зонального керування електроживленням

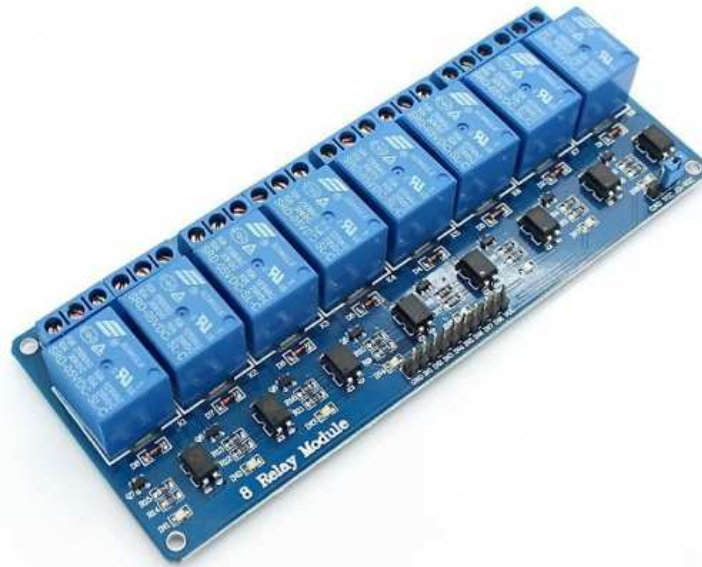


Рисунок 2.6 - Релейний модуль

Для реалізації зонального керування в системі передбачено використання релейних модулів. Вони виконують роль виконавчих пристроїв, які за командою ESP32 можуть вмикати або вимикати живлення окремих ділянок об'єкта.

Застосування реле дозволяє реалізувати одну з основних функцій розроблюваної системи - селективне відключення живлення в зоні, де виявлено небезпеку. Наприклад, при перегріві в електрощитовій або небезпечному стані на кухні система може відключити відповідну зону, не знеструмлюючи весь об'єкт.

У моделі релейний модуль використовується для демонстрації принципу керування навантаженням. У реальній системі для роботи з електромережею необхідно враховувати вимоги електробезпеки, використовувати відповідні комутаційні пристрої та забезпечувати захист силових ліній.

2.3.5 Звуковий оповіщувач та аварійне освітлення



Рисунок 2.7 - Звуковий оповіщувач



Рисунок 2.8 - Аварійне освітлення

Для локального інформування про небезпеку в системі використовується звуковий оповіщувач, тобто сирена або бузер. Його завданням є швидке привернення уваги людей, які перебувають на об'єкті, у разі переходу системи в аварійний стан.

Аварійне освітлення використовується для забезпечення видимості у разі пожежі або зникнення основного живлення. У моделі воно реалізується за допомогою світлодіодних індикаторів, які показують активацію

					ДП.АКТ. 10658921.00.00.000 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підп.	Дата		

аварійного режиму. Такий елемент є важливим, оскільки в реальних умовах пожежа або аварія електромережі може супроводжуватися відключенням основного освітлення.

Звуковий оповісчувач та аварійне освітлення працюють як локальні засоби реагування. Вони не залежать від наявності доступу до інтернету і повинні активуватися безпосередньо за командою контролера при виявленні стану «Пожежа».

2.3.6 Хмарна платформа Blynk та Telegram-сповіщення

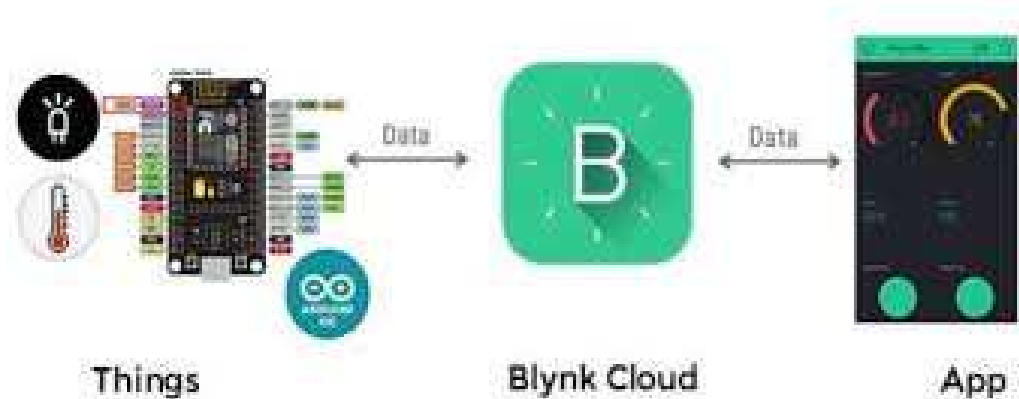


Рисунок 2.9 - Платформа Blynk



Рисунок 2.10 – Telegram

Для забезпечення дистанційного моніторингу та інформування користувача в системі передбачено використання хмарної платформи Blynk і Telegram-сповіщень. Ці засоби не є основною логікою безпеки системи, а

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підп.	Дата		

виконують роль додаткового інтерфейсу для відображення даних, надсилання повідомлень та зручного контролю стану об'єкта.

Хмарна платформа Blynk використовується для візуалізації основних параметрів системи. Через мобільний або вебінтерфейс користувач може переглядати поточні значення датчиків, стан окремих зон, інформацію про наявність основного живлення, активність аварійного освітлення та режим роботи системи. Це дозволяє не лише отримати загальний сигнал тривоги, а й побачити, які саме параметри змінилися та в якій зоні виникла небезпека.

Telegram-сповіщення використовуються як додатковий канал оперативного інформування користувача. У разі переходу системи в стан «Підозра» або «Пожежа» користувач отримує повідомлення з коротким описом ситуації. Наприклад, система може повідомити про підвищення температури в електрощитовій, задимлення в кімнаті, витік газу на кухні або зникнення основного живлення.

Важливо, що Blynk і Telegram не замінюють локальну роботу системи. Навіть у разі відсутності доступу до інтернету основні функції - обробка сигналів, активація сирени, аварійного освітлення та зональне відключення живлення - повинні виконуватися безпосередньо мікроконтролером ESP32. Тобто хмарні сервіси використовуються як засоби зручного моніторингу та сповіщення, а не як основний елемент прийняття рішень.

2.4 Розроблення алгоритму роботи системи.

Алгоритм роботи розроблюваної системи побудований на постійному зчитуванні даних із сенсорів, аналізі отриманих значень та виборі відповідного режиму реагування. Основна ідея полягає в тому, що система не повинна одразу переходити в аварійний режим при кожному незначному відхиленні параметрів. Замість цього використовується кілька станів, які дозволяють точніше оцінити рівень небезпеки.

У роботі системи передбачено такі основні стани:

- «Норма»;

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підп.	Дата		

- «Підозра»;
- «Пожежа»;
- «Відсутність основного живлення»;
- «Витік газу».

У стані «Норма» всі контрольовані параметри перебувають у допустимих межах. Температура, рівень задимленості та показники газового сенсора не перевищують встановлених порогів. У цьому режимі система виконує моніторинг зон, передає поточні дані до інтерфейсу Blynk та очікує можливих змін стану. Сирена, аварійне освітлення та релейне відключення зон у цьому режимі не активуються.

Стан «Підозра» використовується тоді, коли один із параметрів починає виходити за межі нормального значення, але ситуація ще не є критичною. Наприклад, у кімнаті може бути зафіксовано підвищення рівня диму, а в електрощитовій - поступове зростання температури. У такому випадку система формує попередження для користувача через Blynk або Telegram, але ще не виконує повне аварійне реагування, проте система запитує чи потрібно виконувати зональне відключення живлення у даній ділянці, надаючи користувачу більше прозорості і контролю. Такий проміжний режим дозволяє зменшити кількість хибних спрацювань і проінформувати користувача.

Стан «Пожежа» активується у разі досягнення критичних значень або появи кількох ознак небезпеки одночасно. Наприклад, якщо в певній зоні спостерігається високий рівень задимленості разом із підвищенням температури, система визначає це як аварійну ситуацію. У цьому режимі активується сирена, вмикається аварійне освітлення, надсилається повідомлення користувачу та виконується зональне відключення живлення у небезпечній ділянці.

Окремо в алгоритмі враховується стан «Відсутність основного живлення». Якщо система фіксує зникнення напруги в основній мережі, вона повинна перейти на резервне живлення. При цьому робота ESP32, сенсорів та засобів оповіщення зберігається. Також у цьому режимі

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підп.	Дата		

активується аварійне освітлення, а користувач отримує повідомлення про втрату основного живлення.

Стан «Витік газу» передбачений для кухонної зони, оскільки саме там існує найбільша ймовірність появи такої небезпеки. Якщо датчик газу фіксує підвищення його рівня, система переходить у стан попередження та повідомляє користувача про можливий витік. При рівні «Підозра» виконується лише сповіщення, без вимкнення кухні та без увімкнення сирени. Якщо значення газу досягає критичного рівня «Пожежа», система автоматично вимикає кухонну зону, вмикає аварійне освітлення, активує сирену та надсилає повідомлення користувачу. Таким чином, система розділяє звичайне попередження про витік газу та критичну аварійну ситуацію.

Загальна послідовність роботи системи виглядає так: спочатку ESP32 зчитує дані з усіх підключених сенсорів, після чого порівнює їх із заданими пороговими значеннями. Далі визначається стан кожної контрольованої зони. Якщо параметри в нормі, система продовжує моніторинг. Якщо виявлено відхилення, формується попередження. Якщо ж параметри досягають критичних значень, запускається аварійний сценарій реагування.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підп.	Дата		

3 ПРОГРАМНО-АЛГОРИТМІЧНА РЕАЛІЗАЦІЯ ТА МОДЕЛЮВАННЯ СИСТЕМИ

3.1 Загальна структура програмної реалізації системи.

Програмна реалізація розроблюваної системи побудована на базі мікроконтролера ESP32, який виконує функції центрального вузла керування. Саме він забезпечує зчитування даних із сенсорів, аналіз поточного стану контрольованих зон, прийняття рішення щодо рівня небезпеки та формування команд для виконавчих пристроїв. Основна логіка системи реалізована у програмному коді, завантаженому в ESP32, і працює автономно незалежно від наявності зовнішнього підключення до мережі Інтернет.

Програмна структура системи побудована за циклічним принципом і реалізована мовою C++ з використанням Arduino Framework [16] для мікроконтролера ESP32 [7]. Після подачі живлення мікроконтролер виконує ініціалізацію всіх підключених компонентів: температурних сенсорів DHT22, аналогових входів для каналів диму та газу, входу контролю основного живлення, а також виходів керування сиреною, аварійним освітленням і світловими індикаторами стану системи. У середовищі Wokwi сигнали зонального відключення живлення змодельовано за допомогою LED-індикаторів, які показують факт формування відповідної команди контролером. Після завершення ініціалізації система переходить у безперервний режим моніторингу.

У процесі роботи програма послідовно зчитує дані з усіх контрольованих каналів. До них належать температура в житловій кімнаті, кухні, електрощитовій, температура розеткових точок, рівень задимленості в кімнаті та кухні, рівень газу на кухні, а також стан основного джерела живлення. Отримані значення порівнюються з наперед заданими порогоми, після чого для кожної зони формується поточний стан.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підп.	Дата		

У системі реалізовано три базові стани: NORMAL, WARNING та FIRE. Стан NORMAL відповідає нормальній роботі, коли всі параметри перебувають у допустимих межах. Стан WARNING формується тоді, коли хоча б один із контрольованих параметрів перевищує попереджувальний поріг, але ще не досягає критичного значення. Стан FIRE відповідає аварійному режиму, коли значення параметрів досягають критичного рівня і система повинна негайно виконати захисні дії.

Якщо ж аварійний стан виникає в електрощитовій, система переходить у режим глобального аварійного відключення. У цьому випадку відключаються основні навантаження всього об'єкта, активується загальне аварійне освітлення та запускається сирена. Такий сценарій обґрунтований тим, що електрощитова є центральним вузлом розподілу живлення, тому її перегрів або пожежа може становити загрозу для всієї електромережі будинку або квартири.

Окремо в алгоритмі системи передбачено контроль витоку газу в кухонній зоні. Для цього використовуються аналогові показники газового сенсора MQ-2, які постійно аналізуються мікроконтролером ESP32. Якщо рівень газу перевищує попереджувальний поріг, система переходить у стан WARNING та формує попередження для користувача. Якщо концентрація газу досягає критичного рівня, активується стан FIRE. У такому випадку система виконує аварійне реагування: активує сирену, вмикає аварійне освітлення та формує команду на відключення кухонної зони. Такий підхід дозволяє використовувати систему не лише для виявлення пожежі, а й для контролю потенційно небезпечних витоків газу.

Блок-схема загальної логіки роботи системи

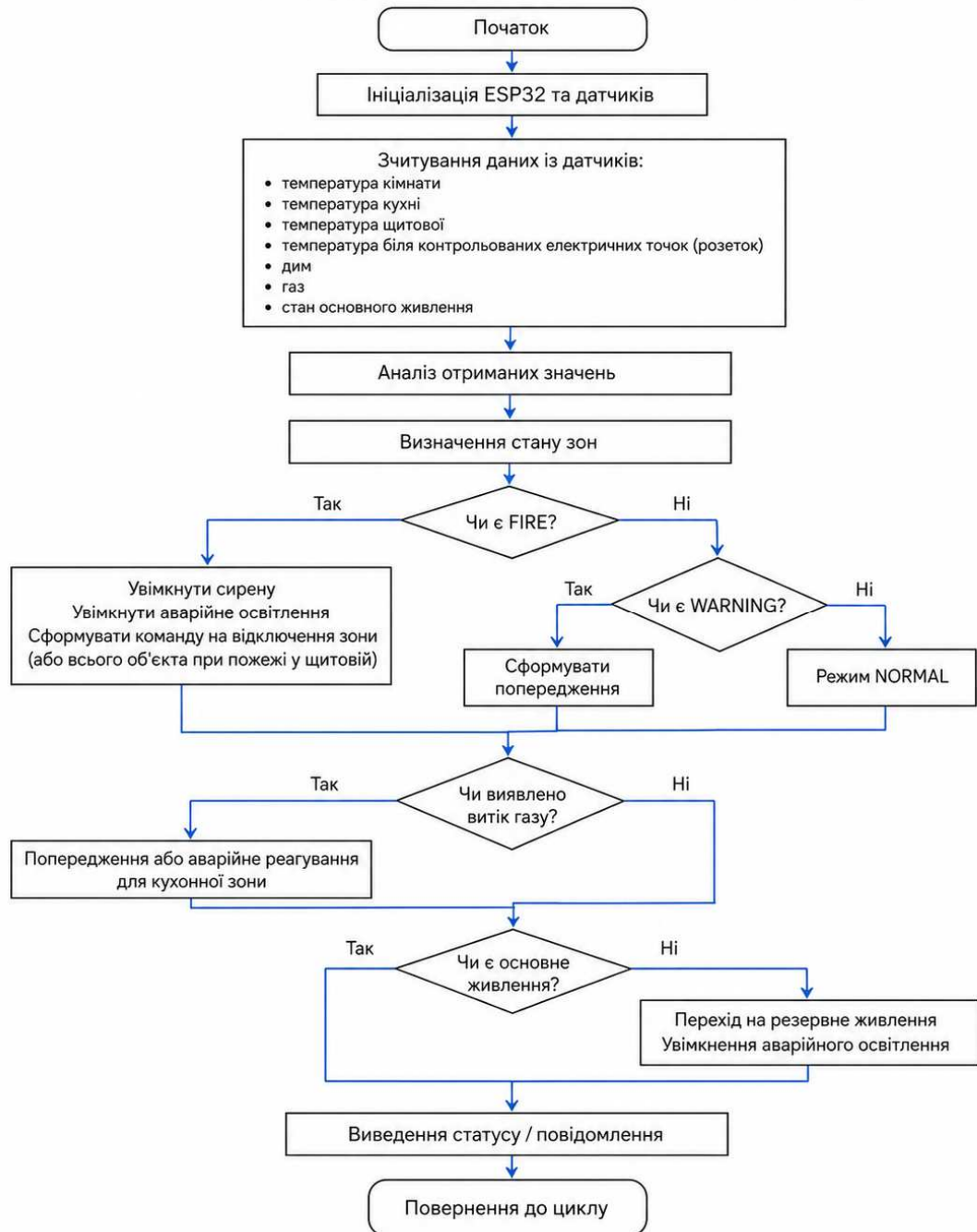


Рисунок 3.1 - Блок-схема логіки системи

Оскільки в середовищі Wokwi датчики типу MQ-2 були змодельовані умовно, для імітації сигналів диму та газу використовувалися потенціометри. Вони дозволяють вручну змінювати аналогове значення та перевіряти реакцію системи на режими WARNING і FIRE.

3.2 Алгоритм обробки сигналів та визначення станів системи.

Після ініціалізації обладнання система переходить до основного циклу роботи, у якому постійно виконується зчитування даних із сенсорів, аналіз отриманих значень та визначення стану кожної контрольованої зони.

У програмі передбачено декілька контрольованих зон: житлова кімната, кухня, електрощитова, а також контрольовані електричні точки біля розеток кімнати та кухні. Для кожної зони аналізуються відповідні параметри: температура, рівень задимленості, рівень газу або температура біля електричних точок. Отримані значення порівнюються з попередньо заданими порогами, після чого система визначає поточний стан зони.

Для опису станів у програмі використовується перелік ZoneState, який містить три можливі режими: NORMAL, WARNING та FIRE. Стан NORMAL означає, що параметри перебувають у допустимих межах. Стан WARNING відповідає попереджувальному режиму, коли зафіксовано перевищення попереджувального порогу. Стан FIRE використовується тоді, коли параметри досягають критичного рівня і потрібно активувати аварійне реагування. Повний програмний код системи є в Додаток А.

```
enum ZoneState {  
    NORMAL,  
    WARNING,  
    FIRE  
};  
  
ZoneState roomState = NORMAL;  
ZoneState kitchenState = NORMAL;  
ZoneState panelState = NORMAL;  
ZoneState roomSocketState = NORMAL;  
ZoneState kitchenSocketState = NORMAL;
```

Для кожного типу параметра в програмі задано окремі порогові значення. Температурні пороги використовуються для кімнати, кухні, електрощитової та розеткових точок. Наприклад, для кімнати й кухні попереджувальний температурний поріг становить 45 °С, а критичний - 60 °С. Для електрощитової ці значення вищі: 55 °С для режиму WARNING та 75 °С для режиму FIRE. Окремо задано пороги для диму й газу:

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підп.	Дата		

попереджувальний рівень - 1500, критичний - 2800 умовних одиниць аналогового сигналу.

```
// Temperature thresholds
const float ROOM_TEMP_WARNING = 45.0;
const float ROOM_TEMP_FIRE    = 60.0;

const float KITCHEN_TEMP_WARNING = 45.0;
const float KITCHEN_TEMP_FIRE    = 60.0;

const float PANEL_TEMP_WARNING = 55.0;
const float PANEL_TEMP_FIRE    = 75.0;

const float ROOM_SOCKET_TEMP_WARNING = 55.0;
const float ROOM_SOCKET_TEMP_FIRE    = 70.0;

const float KITCHEN_SOCKET_TEMP_WARNING = 55.0;
const float KITCHEN_SOCKET_TEMP_FIRE    = 70.0;

// Analog thresholds
const int SMOKE_WARNING_LEVEL = 1500;
const int SMOKE_FIRE_LEVEL    = 2800;

const int GAS_WARNING_LEVEL = 1500;
const int GAS_FIRE_LEVEL    = 2800;
```

Для визначення стану температурного каналу використовується функція, яка отримує поточне значення температури та два пороги: попереджувальний і критичний. Якщо значення температури перевищує критичний поріг, функція повертає стан FIRE. Якщо температура перевищує лише попереджувальний поріг, повертається стан WARNING. Якщо ж температура нижча за обидва пороги, система залишає стан NORMAL.

Окремо враховано ситуацію, коли датчик температури повертає некоректне значення nan. У такому випадку функція не переводить систему в аварійний режим, а повертає NORMAL. У реальній системі таку ситуацію доцільно додатково обробляти як несправність датчика, однак у межах даного прототипу це використовується для уникнення помилкового спрацювання під час моделювання.

```
ZoneState getTempState(float temp, float warningLevel, float
fireLevel) {
    if (isnan(temp)) {
        return NORMAL;
```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк. 45
Зм.	Арк.	№ докум.	Підп.	Дата		

```

    }

    if (temp >= fireLevel) {
        return FIRE;
    }

    if (temp >= warningLevel) {
        return WARNING;
    }

    return NORMAL;
}

```

Для аналогових каналів, які використовуються для імітації диму та газу, застосовується схожий принцип. Мікроконтролер зчитує аналогове значення з відповідного входу ESP32, після чого порівнює його з порогоми WARNING і FIRE. У середовищі Wokwi ці канали були змодельовані за допомогою потенціометрів. Зміна положення повзунка потенціометра імітує зміну рівня диму або газу, що дозволяє перевірити реакцію системи на різні аварійні сценарії.

```

ZoneState getAnalogState(int value, int warningLevel, int
fireLevel) {
    if (value >= fireLevel) {
        return FIRE;
    }

    if (value >= warningLevel) {
        return WARNING;
    }

    return NORMAL;
}

```

Оскільки одна зона може мати кілька джерел небезпеки, у програмі використовується функція вибору найвищого рівня небезпеки. Наприклад, у кімнаті одночасно аналізуються температура, температура біля розетки та рівень диму. Якщо хоча б один із цих параметрів досягає рівня FIRE, уся зона переходить у стан FIRE. Якщо критичного рівня немає, але один із параметрів перебуває на рівні WARNING, зона переходить у попереджувальний стан.

```

ZoneState maxState(ZoneState a, ZoneState b) {
    if (a == FIRE || b == FIRE) {

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        return FIRE;
    }

    if (a == WARNING || b == WARNING) {
        return WARNING;
    }

    return NORMAL;
}

```

Основна обробка станів виконується у функції analyzeSystem(). У ній окремо аналізуються житлова кімната, кухня та електрощитова. Для кімнати враховується температура приміщення, температура біля розетки та дим. Для кухні додатково враховується газовий канал, оскільки саме ця зона має підвищений ризик витоку газу. Електрощитова контролюється за температурою, оскільки її перегрів може свідчити про перевантаження або несправність електричного обладнання. digitalWrite [8].

```

void analyzeSystem() {
    // ROOM
    ZoneState roomTempState = getTempState(
        roomTemp,
        ROOM_TEMP_WARNING,
        ROOM_TEMP_FIRE
    );
    roomSocketState = getTempState(
        roomSocketTemp,
        ROOM_SOCKET_TEMP_WARNING,
        ROOM_SOCKET_TEMP_FIRE
    );
    ZoneState roomSmokeState = getAnalogState(
        roomSmoke,
        SMOKE_WARNING_LEVEL,
        SMOKE_FIRE_LEVEL
    );
    roomState = NORMAL;
    roomState = maxState(roomState, roomTempState);
    roomState = maxState(roomState, roomSocketState);
    roomState = maxState(roomState, roomSmokeState);
    // KITCHEN
    ZoneState kitchenTempState = getTempState(
        kitchenTemp,
        KITCHEN_TEMP_WARNING,
        KITCHEN_TEMP_FIRE
    );
    kitchenSocketState = getTempState(
        kitchenSocketTemp,
        KITCHEN_SOCKET_TEMP_WARNING,
        KITCHEN_SOCKET_TEMP_FIRE
    );
}

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підп.	Дата		

```

);
ZoneState kitchenSmokeState = getAnalogState(
    kitchenSmoke,
    SMOKE_WARNING_LEVEL,
    SMOKE_FIRE_LEVEL
);
ZoneState kitchenGasState = getAnalogState(
    kitchenGas,
    GAS_WARNING_LEVEL,
    GAS_FIRE_LEVEL
);
kitchenState = NORMAL;
kitchenState = maxState(kitchenState, kitchenTempState);
kitchenState = maxState(kitchenState, kitchenSocketState);
kitchenState = maxState(kitchenState, kitchenSmokeState);
kitchenState = maxState(kitchenState, kitchenGasState);
// PANEL
panelState = getTempState(
    panelTemp,
    PANEL_TEMP_WARNING,
    PANEL_TEMP_FIRE
);
}
// CONTROL OUTPUTS
void controlOutputs() {
    bool roomFire = roomState == FIRE;
    bool kitchenFire = kitchenState == FIRE;
    bool panelFire = panelState == FIRE;
    bool anyFire = roomFire || kitchenFire || panelFire;
    // MAIN POWER INDICATION
    if (mainPowerPresent && !panelFire) {
        digitalWrite(MAIN_POWER_OK_LIGHT_PIN, HIGH);
    } else {
        digitalWrite(MAIN_POWER_OK_LIGHT_PIN, LOW);
    }
    if (!mainPowerPresent || anyFire) {
        digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, HIGH);
    } else {
        digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, LOW);
    }
    // PANEL FIRE GLOBAL SHUTDOWN
    if (panelFire) {
        digitalWrite(ROOM_MAIN_LIGHT_PIN, LOW);
        digitalWrite(KITCHEN_MAIN_LIGHT_PIN, LOW);
        digitalWrite(ROOM_POWER_OFF_PIN, HIGH);
        digitalWrite(KITCHEN_POWER_OFF_PIN, HIGH);
        digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, HIGH);
        digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, HIGH);
        digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, HIGH);
        tone(SIREN_PIN, SIREN_FREQUENCY);
        return;
    }
    // ROOM OUTPUTS
    if (roomFire || !mainPowerPresent) {

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підп.	Дата		

```

    digitalWrite(ROOM_MAIN_LIGHT_PIN, LOW);
} else {
    digitalWrite(ROOM_MAIN_LIGHT_PIN, HIGH);
}
if (roomFire) {
    digitalWrite(ROOM_POWER_OFF_PIN, HIGH);
    digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, HIGH);
} else {
    digitalWrite(ROOM_POWER_OFF_PIN, LOW);
    if (mainPowerPresent) {
        digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, LOW);
    } else {
        digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, HIGH);
    }
}
// KITCHEN OUTPUTS
if (kitchenFire || !mainPowerPresent) {
    digitalWrite(KITCHEN_MAIN_LIGHT_PIN, LOW);
} else {
    digitalWrite(KITCHEN_MAIN_LIGHT_PIN, HIGH);
}
if (kitchenFire) {
    digitalWrite(KITCHEN_POWER_OFF_PIN, HIGH);
    digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, HIGH);
} else {
    digitalWrite(KITCHEN_POWER_OFF_PIN, LOW);
    if (mainPowerPresent) {
        digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, LOW);
    } else {
        digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, HIGH);
    }
}
}

```

У кухонній зоні окреме значення має контроль газу. Якщо показник газового каналу перевищує попереджувальний поріг, система може сформувати попередження для користувача. Якщо ж рівень газу досягає критичного значення, кухня переходить у стан FIRE, після чого система повинна виконати аварійне реагування: сформувати команду на відключення кухонної зони, увімкнути аварійне освітлення та активувати сирену.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підп.	Дата		

3.3 Логіка зонального реагування та керування виконавчими пристроями.

Після визначення стану кожної контрольованої зони система переходить до етапу керування виконавчими пристроями. Саме на цьому етапі результати аналізу даних перетворюються на конкретні дії: увімкнення або вимкнення основного освітлення, активацію аварійного освітлення, запуск сирени та формування сигналу зонального відключення живлення.

У програмі керування виконавчими пристроями реалізовано у функції `controlOutputs()`. Вона перевіряє поточні стани кімнати, кухні та електрощитової, після чого визначає, які виходи ESP32 необхідно активувати. Якщо хоча б одна із зон перебуває у стані FIRE, система реагує на відповідну зону. Якщо аварійний стан виникає в окремій зоні, система реагує саме на цю ділянку, не вимикаючи інші зони без потреби.

```
void controlOutputs() {  
    bool roomFire = roomState == FIRE;  
    bool kitchenFire = kitchenState == FIRE;  
    bool panelFire = panelState == FIRE;  
    bool anyFire = roomFire || kitchenFire || panelFire;
```

У системі передбачено окреме реагування для житлової кімнати. Якщо в кімнаті зафіксовано стан FIRE, мікроконтролер вимикає основне світло кімнати, активує індикатор відключення живлення кімнатної зони та вмикає аварійне освітлення. У моделі Wokwi це реалізовано за допомогою LED-індикаторів: вони не відключають реальне електричне навантаження, а показують факт формування керуючого сигналу з боку ESP32.

```
if (roomFire || !mainPowerPresent) {  
    digitalWrite(ROOM_MAIN_LIGHT_PIN, LOW);  
} else {  
    digitalWrite(ROOM_MAIN_LIGHT_PIN, HIGH);  
}  
  
if (roomFire) {  
    digitalWrite(ROOM_POWER_OFF_PIN, HIGH);  
    digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, HIGH);  
} else {
```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підп.	Дата		

```

digitalWrite(ROOM_POWER_OFF_PIN, LOW);

if (mainPowerPresent) {
    digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, LOW);
} else {
    digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, HIGH);
}
}

```

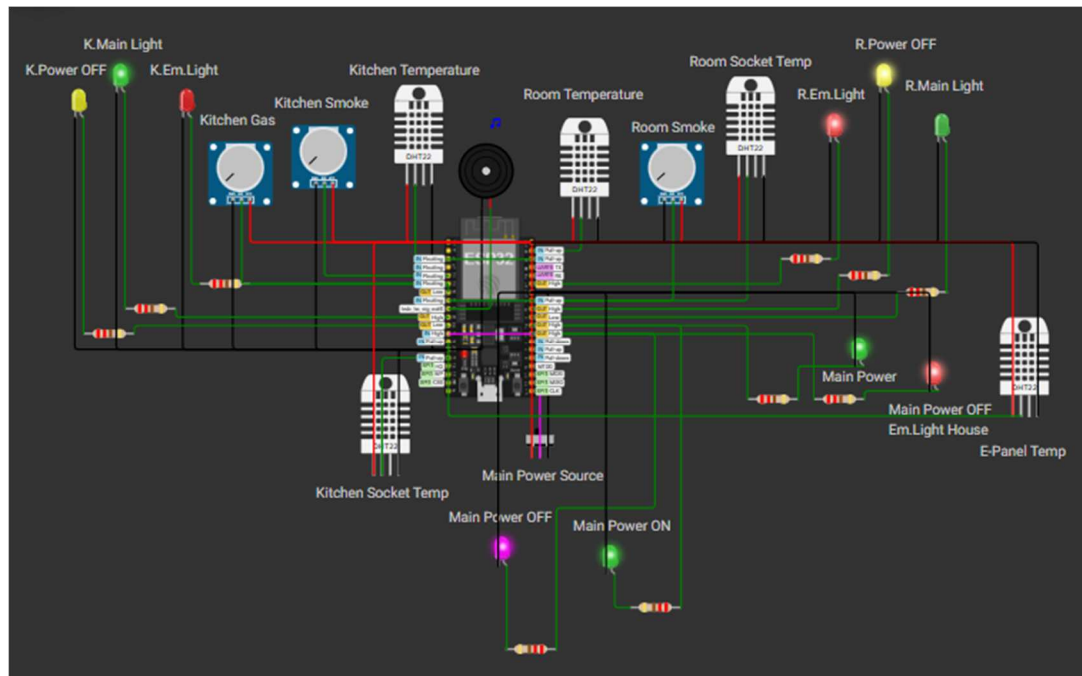


Рисунок 3.2 - Відключення живлення житлової кімнати

Аналогічна логіка реалізована для кухонної зони. Якщо кухня переходить у стан FIRE, система вимикає основне світло кухні, активує сигнал відключення кухонної зони та вмикає аварійне освітлення. Така логіка особливо важлива для кухні, оскільки в цій зоні можуть одночасно виникати кілька небезпечних факторів: перегрів, задимлення або витік газу.

```

if (kitchenFire || !mainPowerPresent) {
    digitalWrite(KITCHEN_MAIN_LIGHT_PIN, LOW);
} else {
    digitalWrite(KITCHEN_MAIN_LIGHT_PIN, HIGH);
}

if (kitchenFire) {
    digitalWrite(KITCHEN_POWER_OFF_PIN, HIGH);
    digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, HIGH);
} else {
    digitalWrite(KITCHEN_POWER_OFF_PIN, LOW);

    if (mainPowerPresent) {
        digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, LOW);
    } else {

```

```

digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, HIGH);
}
}

```

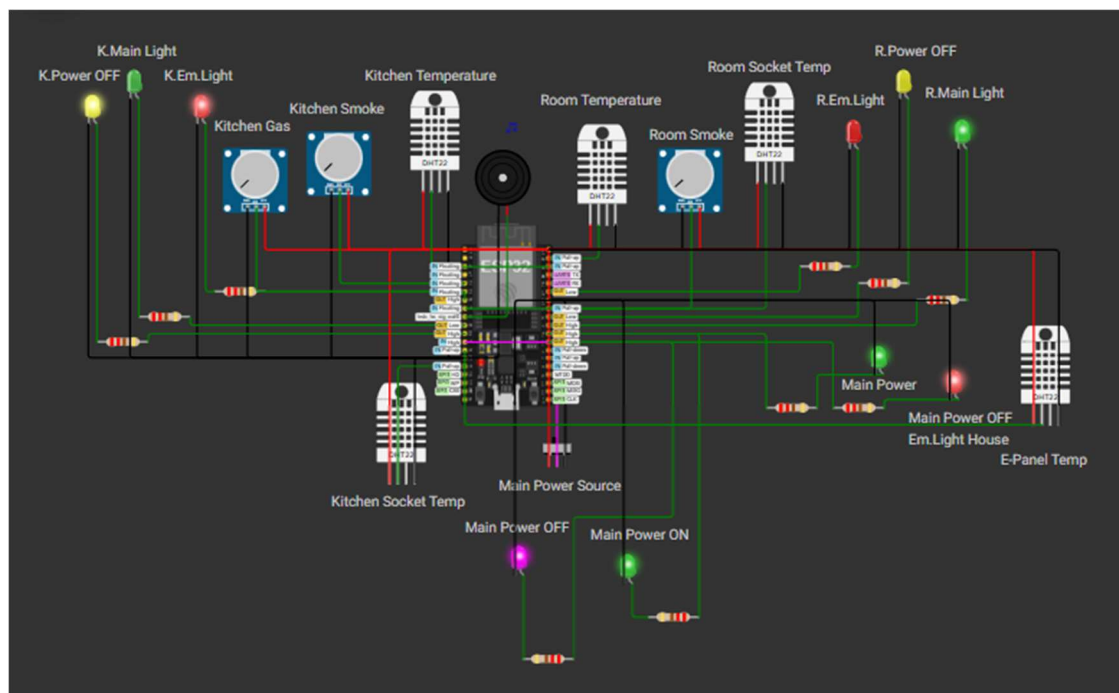


Рисунок 3.3 - Відключення живлення кухні

Окремо в алгоритмі виділено сценарій пожежі або критичного перегріву в електрощитовій. На відміну від кімнати чи кухні, електрощитова є центральною зоною розподілу живлення, тому аварійний стан у ній може становити небезпеку для всього об'єкта. Через це при panelFire система переходить у режим глобального аварійного відключення. У цьому режимі вимикається основне світло кімнати та кухні, активуються індикатори відключення обох зон, вмикається аварійне освітлення, загальне аварійне світло та сирена.

```

if (panelFire) {
    digitalWrite(ROOM_MAIN_LIGHT_PIN, LOW);
    digitalWrite(KITCHEN_MAIN_LIGHT_PIN, LOW);

    digitalWrite(ROOM_POWER_OFF_PIN, HIGH);
    digitalWrite(KITCHEN_POWER_OFF_PIN, HIGH);

    digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, HIGH);
    digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, HIGH);
    digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, HIGH);

    tone(SIREN_PIN, SIREN_FREQUENCY);
    return;
}

```

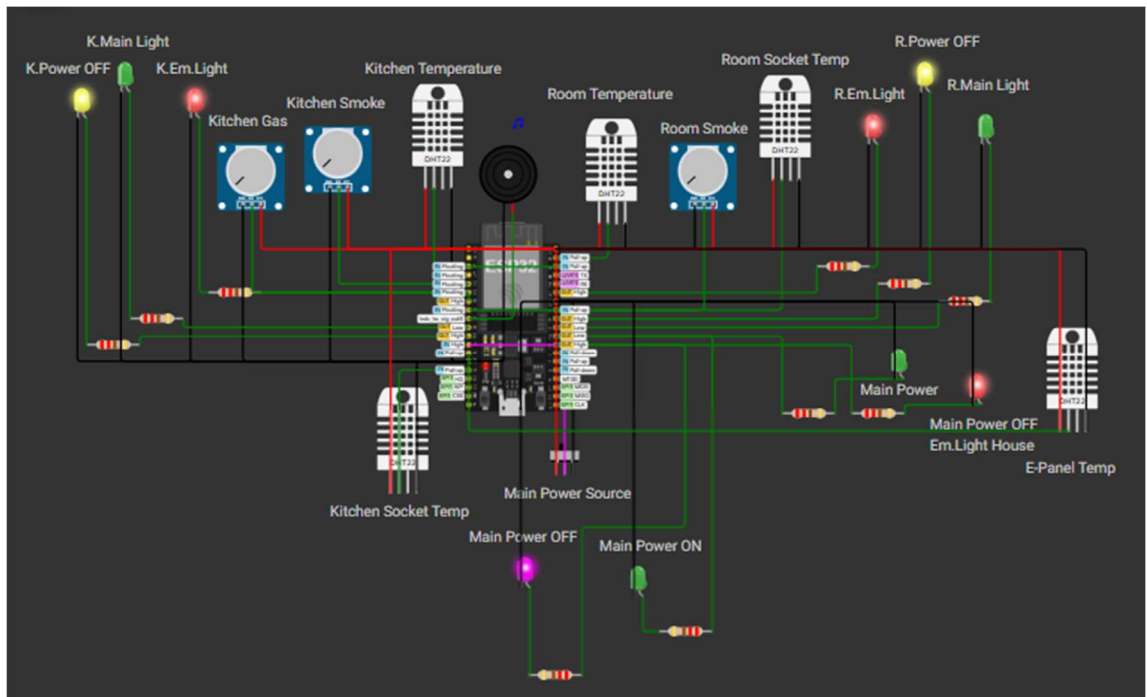


Рисунок 3.4 - Відключення живлення всього житла

Сирена в системі використовується як основний локальний засіб звукового оповіщення. Вона активується у випадку, якщо хоча б одна із зон перебуває у стані FIRE. Якщо ж аварійних станів немає, сирена вимикається. Такий підхід дозволяє уникнути її спрацювання у стані WARNING, коли система лише попереджає користувача про можливу небезпеку.

```
if (anyFire) {
    tone(SIREN_PIN, SIREN_FREQUENCY);
} else {
    noTone(SIREN_PIN);
}
```

Також у системі реалізовано керування індикацією основного живлення та загальним аварійним освітленням. Якщо основне живлення присутнє і в електрощитовій немає пожежі, активується індикатор нормального живлення. Якщо ж основне живлення втрачено або виявлено стан FIRE у будь-якій зоні, вмикається загальне аварійне освітлення.

```
if (mainPowerPresent && !panelFire) {
    digitalWrite(MAIN_POWER_OK_LIGHT_PIN, HIGH);
} else {
    digitalWrite(MAIN_POWER_OK_LIGHT_PIN, LOW);
}
```

```

if (!mainPowerPresent || anyFire) {
    digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, HIGH);
} else {
    digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, LOW);
}

```

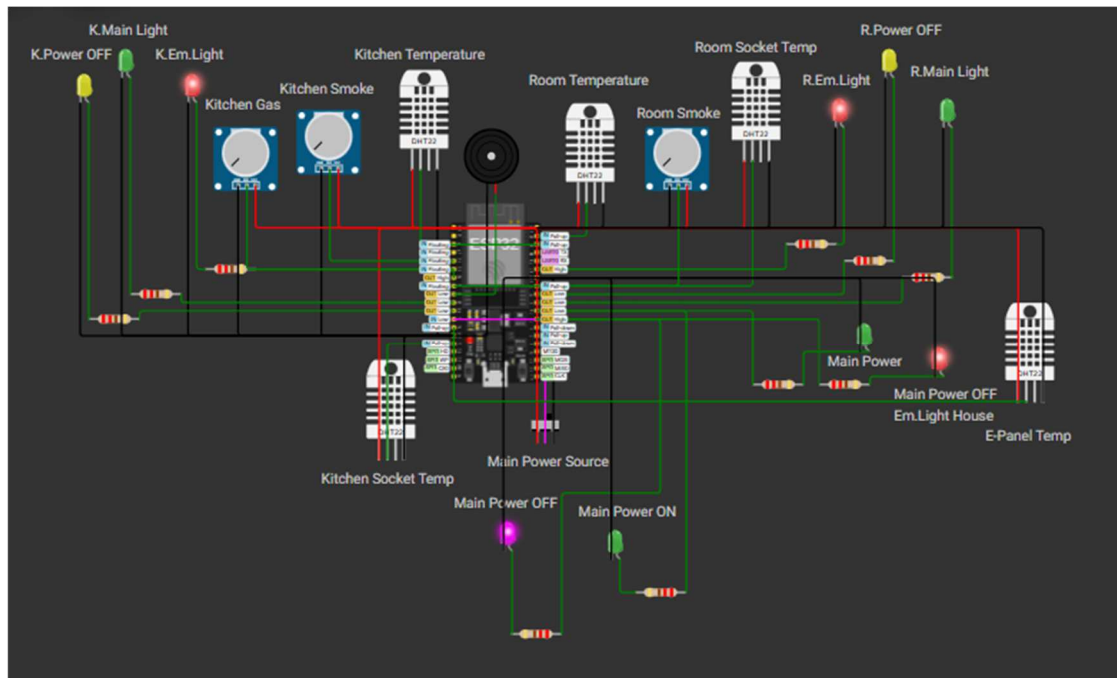


Рисунок 3.5 - Відсутнє живлення (не пожежа)

Важливо зазначити, що в середовищі Wokwi виконавчі пристрої, які відповідають за відключення живлення, змодельовано умовно. Замість реальних релейних модулів використовуються світлодіодні індикатори, які показують факт подачі керуючого сигналу. Наприклад, увімкнення індикатора ROOM_POWER_OFF_PIN означає, що система сформувала команду на відключення кімнатної зони, а увімкнення KITCHEN_POWER_OFF_PIN - команду на відключення кухні. У реальній системі ці сигнали могли б подаватися на релейні модулі або інші комутаційні пристрої.

3.4 Реалізація контролю основного та резервного живлення.

У розроблюваній системі окремо передбачено контроль наявності основного живлення. Це важливо, оскільки під час пожежі або аварійної ситуації електроживлення об'єкта може зникнути повністю або частково. У

такому випадку система не повинна припиняти роботу, адже саме в аварійний момент необхідно зберегти працездатність контролера, сенсорів, сирени та аварійного освітлення.

У програмній реалізації стан основного живлення контролюється через окремий цифровий вхід ESP32. Для цього використовується змінна `mainPowerPresent`, яка приймає значення `true`, якщо основне живлення присутнє, і `false`, якщо воно зникло. Зчитування цього стану виконується у функції `readSensors()` разом із зчитуванням показників температури, диму та газу.

```
#define MAIN_POWER_SOURCE_PIN 14
const int MAIN_POWER_PRESENT_LEVEL = HIGH;

bool mainPowerPresent = true;
```

Зчитування стану основного живлення у функції `readSensors()`:

```
mainPowerPresent = digitalRead(MAIN_POWER_SOURCE_PIN) ==
MAIN_POWER_PRESENT_LEVEL;
```

Якщо основне живлення присутнє, система працює у штатному режимі. У цьому випадку активується індикатор наявності живлення, а аварійне освітлення залишається вимкненим, якщо немає стану `FIRE`. Якщо ж основне живлення зникає, система вимикає індикатор нормального живлення та активує загальне аварійне освітлення.

У моделі Wokwi цей процес показано за допомогою LED-індикаторів. Один індикатор відображає наявність основного живлення, а інший - увімкнення аварійного освітлення при втраті живлення або при виникненні пожежі.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						55
Зм.	Арк.	№ докум.	Підп.	Дата		

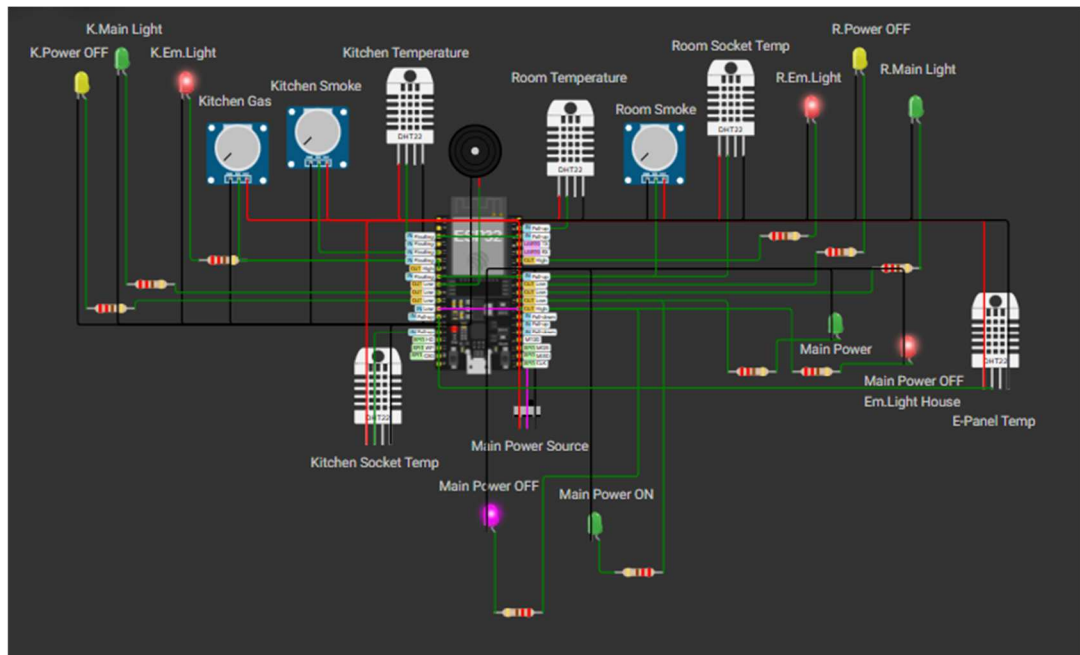


Рисунок 3.6 - Відсутність основного живлення

```

if (mainPowerPresent && !panelFire) {
    digitalWrite(MAIN_POWER_OK_LIGHT_PIN, HIGH);
} else {
    digitalWrite(MAIN_POWER_OK_LIGHT_PIN, LOW);
}

if (!mainPowerPresent || anyFire) {
    digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, HIGH);
} else {
    digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, LOW);
}

```

Особливістю алгоритму є те, що при зникненні основного живлення система не переходить у стан повної зупинки. Навпаки, вона продовжує виконувати моніторинг, а аварійне освітлення вмикається для забезпечення мінімальної видимості на об'єкті. У програмі також передбачено, що при відсутності основного живлення основне світло кімнати та кухні вмикається, а аварійне освітлення відповідних зон активується.

```

if (roomFire || !mainPowerPresent) {
    digitalWrite(ROOM_MAIN_LIGHT_PIN, LOW);
} else {
    digitalWrite(ROOM_MAIN_LIGHT_PIN, HIGH);
}

if (kitchenFire || !mainPowerPresent) {
    digitalWrite(KITCHEN_MAIN_LIGHT_PIN, LOW);
}

```

```

} else {
    digitalWrite(KITCHEN_MAIN_LIGHT_PIN, HIGH);
}

```

При втраті основного живлення система також виводить відповідне повідомлення в Serial Monitor. Це дозволяє під час моделювання перевірити, що контролер правильно визначив відсутність живлення та перейшов до аварійного режиму освітлення.

```

if (!mainPowerPresent) {
    Serial.println("POWER WARNING: MAIN POWER LOST");
    Serial.println("SYSTEM CONTINUES WORKING FROM BACKUP
POWER");
    Serial.println("EMERGENCY LIGHTS ON");
}

```

```

===== SYSTEM STATUS =====
Main Power: LOST

Room Temp: 23.10
Room Socket Temp: 29.90
Room Smoke: 0
Room State: NORMAL

Kitchen Temp: 21.40
Kitchen Socket Temp: 16.30
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: NORMAL

Panel Temp: 27.30
Panel State: NORMAL

POWER WARNING: MAIN POWER LOST
SYSTEM CONTINUES WORKING FROM BACKUP POWER
EMERGENCY LIGHTS ON
=====

```

Рисунок 3.7 - Повідомлення про аварійний режим

У межах моделі резервне живлення не реалізується як окремий фізичний акумуляторний блок, а моделюється логічно. Тобто при зникненні

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підп.	Дата		

сигналу основного живлення програма продовжує працювати, а світлодіодні індикатори показують перехід системи в аварійний режим. У реальній системі для цього необхідно передбачити акумулятор, модуль заряджання та схему автоматичного перемикачання між основним і резервним джерелом живлення.

3.5 Моделювання системи у середовищі Wokwi.

Для перевірки роботи розробленої системи було використано середовище Wokwi, яке дозволяє моделювати роботу мікроконтролерів, датчиків та виконавчих елементів без складання фізичного макета. Це зручно на етапі розробки, оскільки можна швидко перевірити логіку програми, правильність підключення компонентів і реакцію системи на різні аварійні ситуації.

У межах моделі центральним елементом виступає мікроконтролер ESP32, до якого підключено сенсори температури, аналогові входи для імітації диму та газу, світлодіодні індикатори, бузер і вхід контролю основного живлення. Програмна частина реалізована мовою C++ з використанням Arduino Framework, що дозволяє застосовувати стандартні функції `setup()`, `loop()`, `pinMode()`, `digitalWrite()`, `analogRead()` та інші засоби керування входами й виходами ESP32.

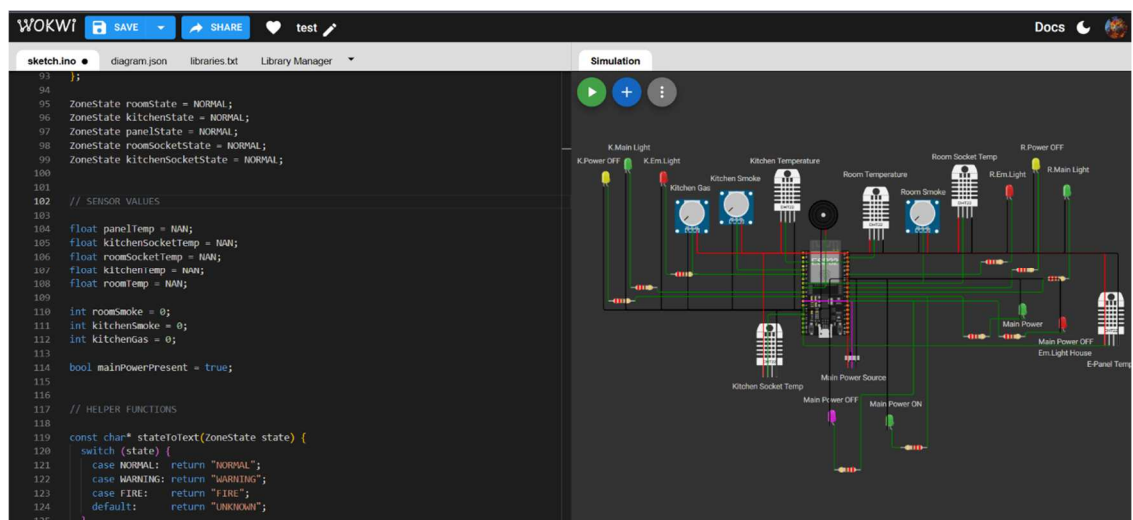


Рисунок 3.8 - Середовище Wokwi

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підп.	Дата		

У моделі Wokwi температурні канали реалізовано за допомогою датчиків DHT22. Вони використовуються для контролю температури в кімнаті, кухні, електрощитовій, а також біля контрольованих електричних точок підвищеного ризику. Такі точки в межах моделі відповідають розеткам або місцям можливого перегріву електричних з'єднань.

Канали диму та газу у Wokwi були змодельовані за допомогою потенціометрів. Це зроблено тому, що в середовищі моделювання не завжди доцільно або можливо використовувати повну модель газо-димового сенсора типу MQ-2. Зміна положення повзунка потенціометра імітує зміну аналогового сигналу від датчика диму або газу. Отримане значення зчитується мікроконтролером ESP32 через аналоговий вхід і порівнюється з порогами WARNING та FIRE.

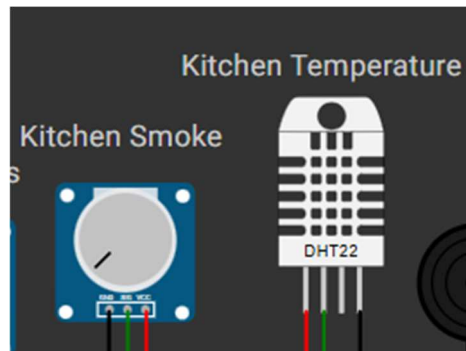


Рисунок 3.9 - Датчик температури DHT22 і потенціометр що імітує газо-димового сенсора MQ-2

Для відображення стану системи використано світлодіодні індикатори. Частина з них показує роботу основного та аварійного освітлення, а частина - факт формування команди на відключення окремої зони. Оскільки в даній моделі реальні релейні модулі не використовувалися, їхню функцію умовно замінено LED-індикаторами.

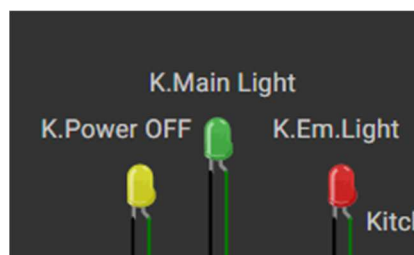


Рисунок 3.10 - LED-індикатори кухні

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підп.	Дата		

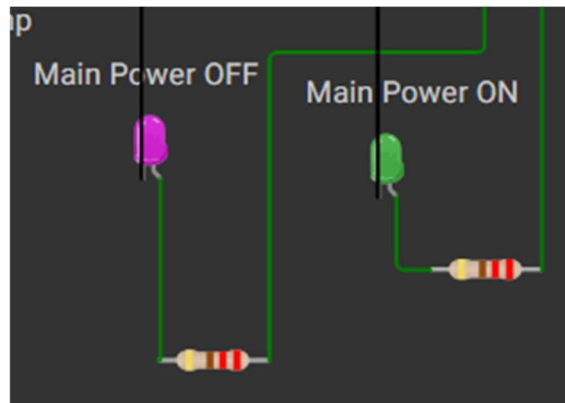


Рисунок 3.11 - LED-індикатори головного живлення

У реальній системі ці сигнали могли б подаватися на релейні модулі, контактори або інші комутаційні пристрої. Проте для перевірки логіки роботи в середовищі Wokwi достатньо LED-індикації, оскільки головним завданням моделі є демонстрація правильності програмного алгоритму.

У програмі передбачено окремі виходи для основного світла кімнати та кухні, аварійного освітлення, індикаторів відключення зон, загального аварійного освітлення та сирени. Звукове оповіщення в моделі реалізовано за допомогою бузера, який активується при переході хоча б однієї зони у стан FIRE. У коді це реалізовано через функцію `tone()`, яка подає сигнал заданої частоти на відповідний вихід ESP32.

```
pinMode(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, OUTPUT);
pinMode(MAIN_POWER_OK_LIGHT_PIN, OUTPUT);

pinMode(ROOM_MAIN_LIGHT_PIN, OUTPUT);
pinMode(ROOM_POWER_OFF_PIN, OUTPUT);
pinMode(ROOM_EMERGENCY_LIGHT_PIN, OUTPUT);

pinMode(KITCHEN_MAIN_LIGHT_PIN, OUTPUT);
pinMode(KITCHEN_POWER_OFF_PIN, OUTPUT);
pinMode(KITCHEN_EMERGENCY_LIGHT_PIN, OUTPUT);

pinMode(SIREN_PIN, OUTPUT);
```



Рисунок 3.12 – Сирена

Для контролю результатів роботи використовується Serial Monitor. У ньому виводяться поточні значення температури, диму, газу, стан основного живлення, а також статус кожної зони. Це дозволяє перевірити, чи правильно система визначає режими NORMAL, WARNING і FIRE, а також чи відповідає реакція виконавчих елементів заданому сценарію.

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 29.90
Room Smoke: 0
Room State: NORMAL

Kitchen Temp: 21.40
Kitchen Socket Temp: 16.30
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: NORMAL

Panel Temp: 27.30
Panel State: NORMAL

=====
  
```

Рисунок 3.13 - Текстове повідомлення в Serial Monitor

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підп.	Дата		

3.6 Перевірка роботи системи в основних сценаріях.

Після побудови моделі системи у середовищі Wokwi було виконано перевірку її роботи в основних режимах. Метою перевірки - підтвердити, що система правильно реагує на зміну параметрів середовища, переходить між станами NORMAL, WARNING і FIRE, а також активує відповідні виконавчі елементи.

Перевірка виконувалася шляхом імітації аварійних ситуацій у різних зонах об'єкта. Температурні значення змінювалися за допомогою датчиків DHT22 [10], а сигнали диму та газу моделювалися потенціометрами. Реакція системи контролювалася за станом LED-індикаторів, сирени та повідомленнями в Serial Monitor.

Нормальний стан усіх датчиків:

У першому сценарії перевірялася робота системи за відсутності небезпечних факторів. Усі значення температури, диму та газу перебували нижче встановлених порогів. У такому режимі кімната, кухня та електрощитова мають стан NORMAL, сирена вимкнена, аварійне освітлення не активується, а сигнали відключення зон не формуються.

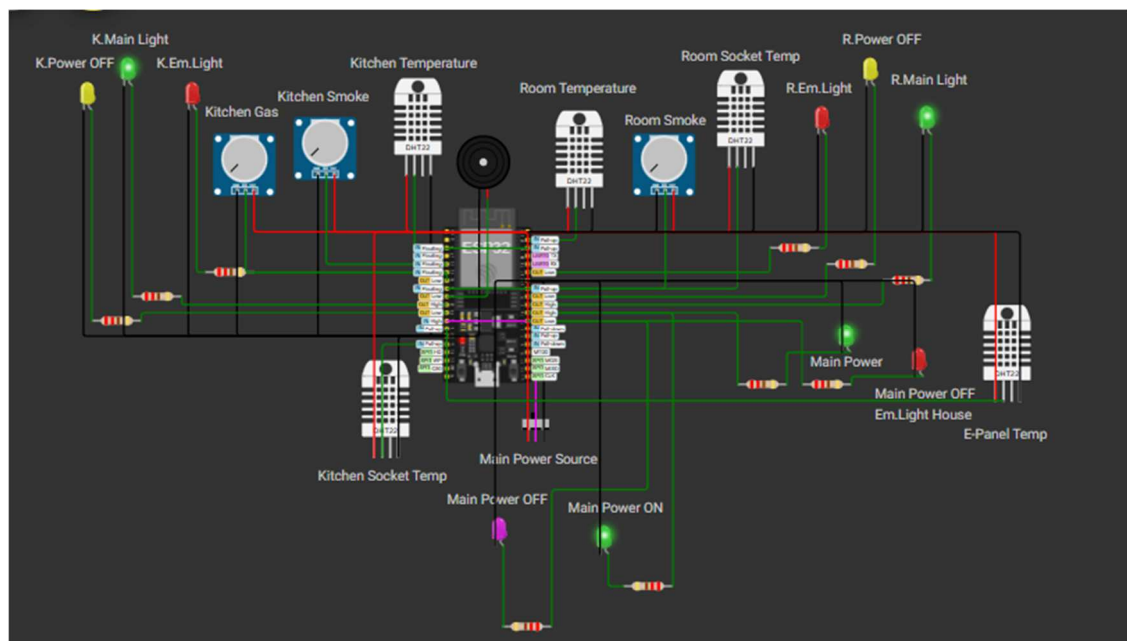


Рисунок 3.14 - Нормальний стан усіх датчиків

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 29.90
Room Smoke: 0
Room State: NORMAL

Kitchen Temp: 21.40
Kitchen Socket Temp: 16.30
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: NORMAL

Panel Temp: 27.30
Panel State: NORMAL

=====

```

Рисунок 3.15 - Нормальний стан усіх датчиків

Втрата основного живлення:

У цьому сценарії перевірялася реакція системи на зникнення основного живлення. При зміні стану відповідного входу ESP32 система визначає втрату основної мережі. Після цього вмикається індикатор наявності основного живлення, активується аварійне освітлення, а в Serial Monitor виводиться повідомлення про перехід системи на резервне живлення.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підп.	Дата		

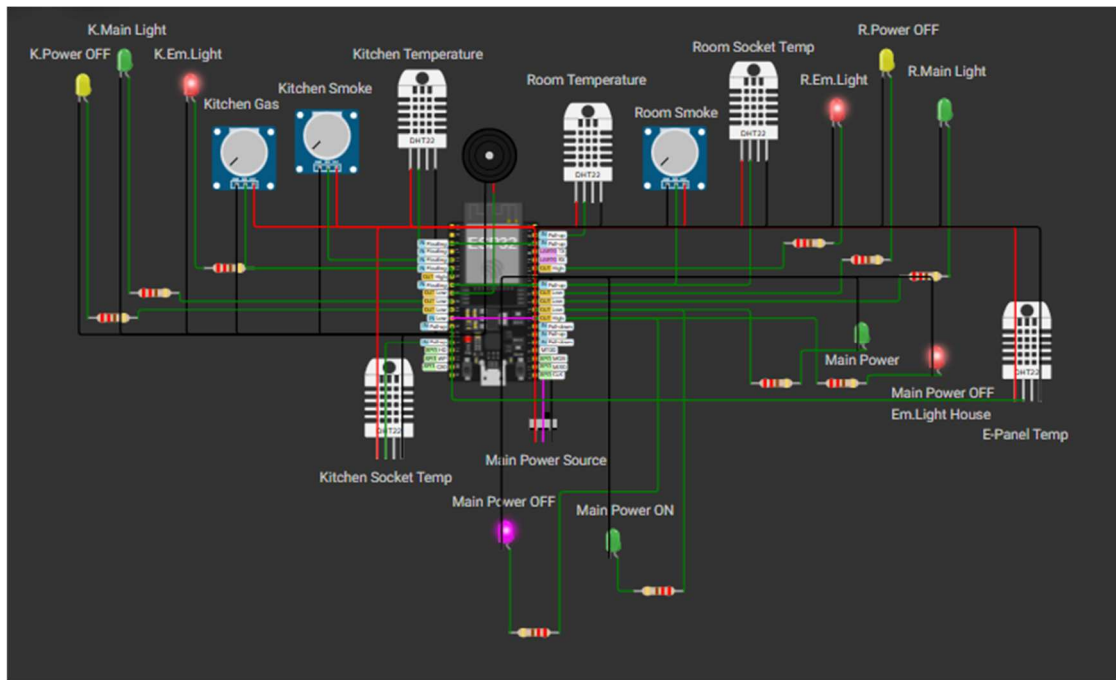


Рисунок 3.16 - Втрата основного живлення

```

===== SYSTEM STATUS =====
Main Power: LOST

Room Temp: 23.10
Room Socket Temp: 29.90
Room Smoke: 0
Room State: NORMAL

Kitchen Temp: 21.40
Kitchen Socket Temp: 16.30
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: NORMAL

Panel Temp: 27.30
Panel State: NORMAL

POWER WARNING: MAIN POWER LOST
SYSTEM CONTINUES WORKING FROM BACKUP POWER
EMERGENCY LIGHTS ON
=====

```

Рисунок 3.17 - Втрата основного живлення

Дим у кімнаті: WARNING і FIRE:

Для житлової кімнати було перевірено сценарій появи диму. При досягненні попереджувального рівня система переходить у стан WARNING і формує повідомлення про задимлення. У цьому режимі сирена ще не активується, а зональне відключення не виконується.

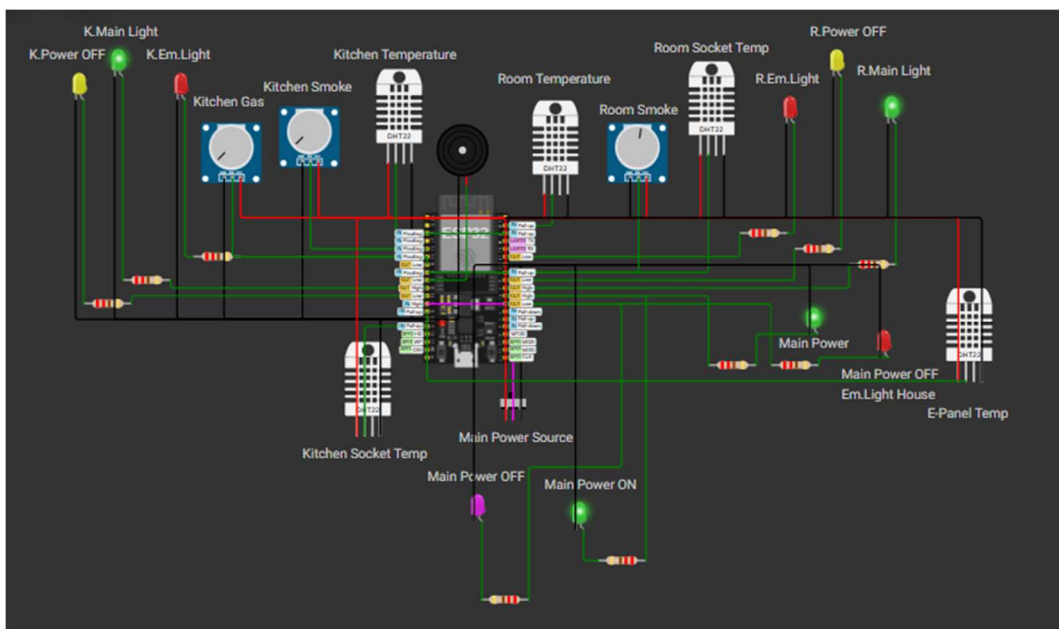


Рисунок 3.18 - Дим у кімнаті WARNING

```
===== SYSTEM STATUS =====  
Main Power: PRESENT  
  
Room Temp: 23.10  
Room Socket Temp: 29.90  
Room Smoke: 2170  
Room State: WARNING  
WARNING: SMOKE DETECTED IN ROOM  
  
Kitchen Temp: 21.40  
Kitchen Socket Temp: 16.30  
Kitchen Smoke: 0  
Kitchen Gas: 0  
Kitchen State: NORMAL  
  
Panel Temp: 27.30  
Panel State: NORMAL  
  
=====
```

Рисунок 3.19 - Дим у кімнаті WARNING

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підп.	Дата		

При подальшому збільшенні рівня диму до критичного значення система переходить у стан FIRE. У цьому випадку активується сирена, вмикається аварійне освітлення кімнати та формується сигнал на відключення кімнатної зони.

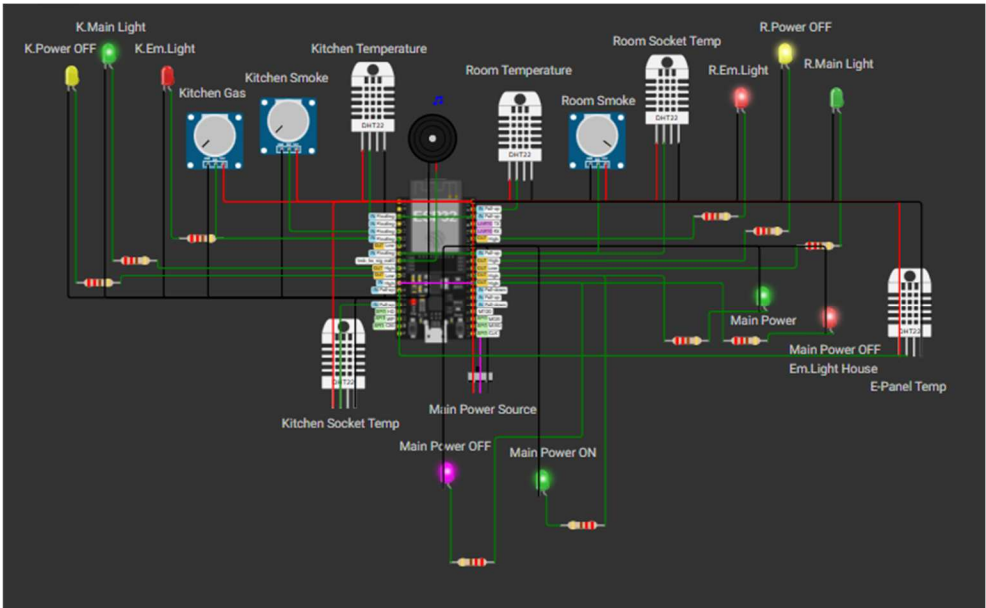


Рисунок 3.20 - Дим у кімнаті FIRE

```
===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 29.90
Room Smoke: 3987
Room State: FIRE
FIRE: SMOKE FIRE LEVEL IN ROOM
ROOM POWER DISABLED
ROOM EMERGENCY LIGHT ON

Kitchen Temp: 21.40
Kitchen Socket Temp: 16.30
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: NORMAL

Panel Temp: 27.30
Panel State: NORMAL

=====
```

Рисунок 3.21 - Дим у кімнаті FIRE

Температура кімнати: WARNING і FIRE:

У цьому сценарії перевірялася реакція системи на підвищення температури в житловій кімнаті. При перевищенні попереджувального порогу система визначає стан WARNING і виводить повідомлення про підвищену температуру.

```
===== SYSTEM STATUS =====  
Main Power: PRESENT  
  
Room Temp: 57.90  
Room Socket Temp: 23.90  
Room Smoke: 44  
Room State: WARNING  
WARNING: HIGH TEMPERATURE IN ROOM  
  
Kitchen Temp: 22.20  
Kitchen Socket Temp: 15.40  
Kitchen Smoke: 0  
Kitchen Gas: 0  
Kitchen State: NORMAL  
  
Panel Temp: 23.90  
Panel State: NORMAL  
  
=====
```

Рисунок 3.22 - Температура кімнати WARNING

У режимі FIRE система вимикає основне світло кімнати, активує індикатор відключення кімнатної зони, вмикає аварійне освітлення та запускає сирену.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						67
Зм.	Арк.	№ докум.	Підп.	Дата		

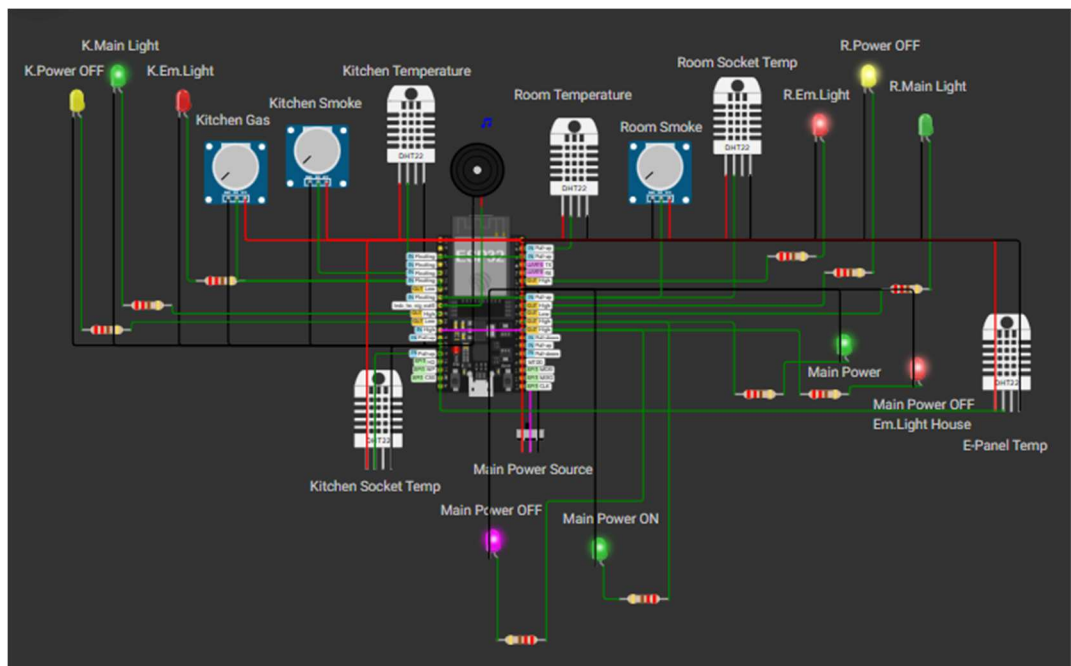


Рисунок 3.23 - Температура кімнати FIRE

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 66.40
Room Socket Temp: 23.90
Room Smoke: 44
Room State: FIRE
FIRE: HIGH TEMPERATURE IN ROOM
ROOM POWER DISABLED
ROOM EMERGENCY LIGHT ON

Kitchen Temp: 22.20
Kitchen Socket Temp: 15.40
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: NORMAL

Panel Temp: 23.90
Panel State: NORMAL

=====

```

Рисунок 3.24 - Температура кімнати FIRE

Температура + дим кімнати: WARNING і FIRE:

Окремо перевірено комбінований сценарій, коли в кімнаті одночасно підвищується температура та зростає рівень диму. Така ситуація є більш небезпечною, оскільки одночасна поява кількох ознак може свідчити про реальний розвиток пожежі.

У стані WARNING система формує попередження, а при досягненні критичних значень переходить у стан FIRE і запускає повний аварійний сценарій для кімнати.

```
===== SYSTEM STATUS =====  
Main Power: PRESENT  
  
Room Temp: 55.40  
Room Socket Temp: 23.90  
Room Smoke: 2170  
Room State: WARNING  
WARNING: HIGH TEMPERATURE IN ROOM  
WARNING: SMOKE DETECTED IN ROOM  
  
Kitchen Temp: 22.20  
Kitchen Socket Temp: 15.40  
Kitchen Smoke: 0  
Kitchen Gas: 0  
Kitchen State: NORMAL  
  
Panel Temp: 23.90  
Panel State: NORMAL
```

Рисунок 3.25 - Температура + дим кімнати WARNING

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						69
Зм.	Арк.	№ докум.	Підп.	Дата		

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 66.40
Room Socket Temp: 23.90
Room Smoke: 4095
Room State: FIRE
FIRE: HIGH TEMPERATURE IN ROOM
FIRE: SMOKE FIRE LEVEL IN ROOM
ROOM POWER DISABLED
ROOM EMERGENCY LIGHT ON

Kitchen Temp: 22.20
Kitchen Socket Temp: 15.40
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: NORMAL

Panel Temp: 23.90
Panel State: NORMAL

```

Рисунок 3.26 - Температура + дим кімнати FIRE

Перегрів розетки кімнати і кухні: WARNING і FIRE:

У цьому сценарії перевірявся контроль температури біля електричної точки кімнати і кухні окремо. Така перевірка потрібна для імітації локального перегріву розетки або місця підключення навантаження.

При досягненні рівня WARNING система формує попередження про можливий перегрів. При досягненні рівня FIRE виконується аварійне реагування для кімнатної зони: формується сигнал на відключення, вмикається аварійне освітлення та активується сирена.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підп.	Дата		

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 57.90
Room Smoke: 44
Room State: WARNING
WARNING: ROOM SOCKET OVERHEATING

Kitchen Temp: 27.30
Kitchen Socket Temp: 16.30
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: NORMAL

Panel Temp: 27.30
Panel State: NORMAL

=====

```

Рисунок 3.27 - Перегрів розетки кімнати WARNING

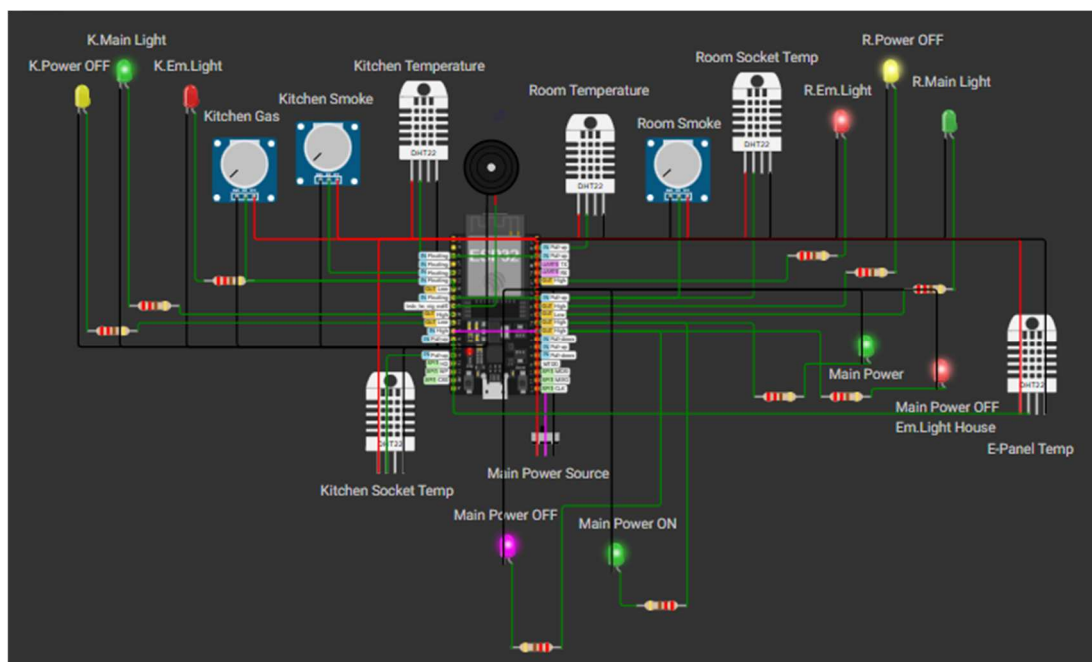


Рисунок 3.28 - Перегрів розетки кімнати FIRE

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 80.00
Room Smoke: 44
Room State: FIRE
FIRE: ROOM SOCKET OVERHEATING / POSSIBLE SHORT CIRCUIT
ROOM POWER DISABLED
ROOM EMERGENCY LIGHT ON

Kitchen Temp: 27.30
Kitchen Socket Temp: 16.30
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: NORMAL

Panel Temp: 27.30
Panel State: NORMAL

=====

```

Рисунок 3.29 - Перегрів розетки кімнати FIRE

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 26.50
Room Smoke: 44
Room State: NORMAL

Kitchen Temp: 27.30
Kitchen Socket Temp: 55.40
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: WARNING
WARNING: KITCHEN SOCKET OVERHEATING

Panel Temp: 27.30
Panel State: NORMAL

=====

```

Рисунок 3.30 - Перегрів розетки кухні WARNING

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підп.	Дата		

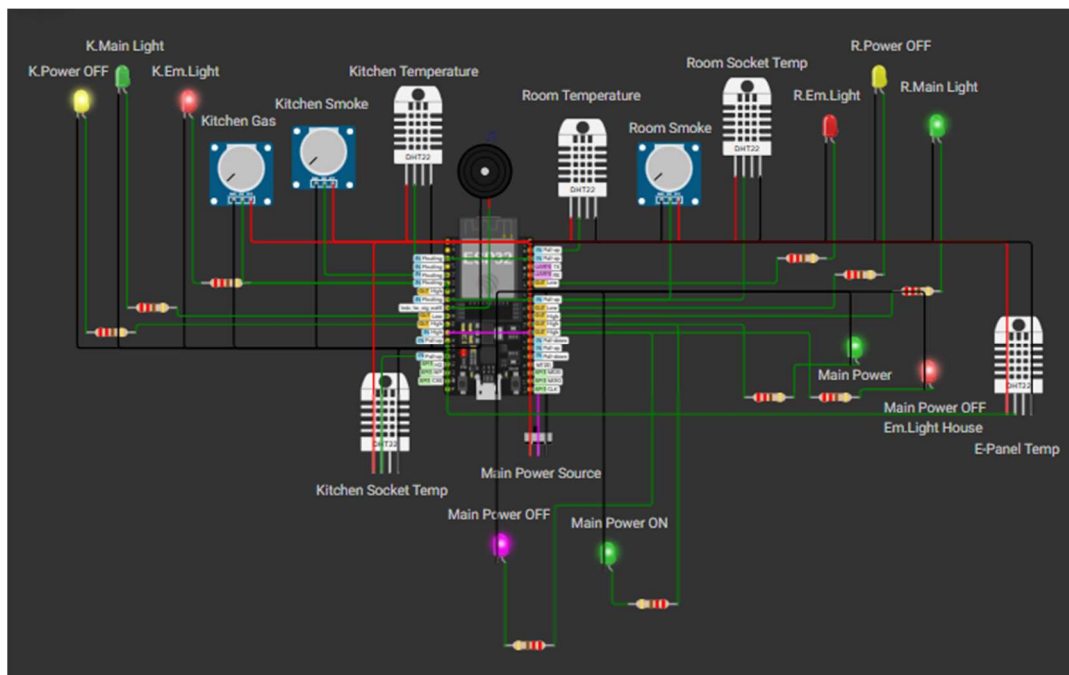


Рисунок 3.31 - Перегрів розетки кухні FIRE

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 26.50
Room Smoke: 44
Room State: NORMAL

Kitchen Temp: 27.30
Kitchen Socket Temp: 80.00
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: FIRE
FIRE: KITCHEN SOCKET OVERHEATING / POSSIBLE SHORT CIRCUIT
KITCHEN POWER DISABLED
KITCHEN EMERGENCY LIGHT ON

Panel Temp: 27.30
Panel State: NORMAL

=====
  
```

Рисунок 3.32 - Перегрів розетки кухні FIRE

Температура кухні: WARNING і FIRE:

Для кухонної зони перевірено сценарій підвищення температури. При перевищенні попереджувального порогу система переходить у стан WARNING.

У режимі FIRE система формує команду на відключення кухонної зони, активує аварійне освітлення кухні та вмикає сирену.

```
===== SYSTEM STATUS =====  
Main Power: PRESENT  
  
Room Temp: 23.10  
Room Socket Temp: 29.90  
Room Smoke: 44  
Room State: NORMAL  
  
Kitchen Temp: 58.80  
Kitchen Socket Temp: 16.30  
Kitchen Smoke: 0  
Kitchen Gas: 0  
Kitchen State: WARNING  
WARNING: HIGH TEMPERATURE IN KITCHEN  
  
Panel Temp: 27.30  
Panel State: NORMAL  
  
=====
```

Рисунок 3.33 - Температура кухні WARNING

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						74
Зм.	Арк.	№ докум.	Підп.	Дата		

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 29.90
Room Smoke: 44
Room State: NORMAL

Kitchen Temp: 80.00
Kitchen Socket Temp: 16.30
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: FIRE
FIRE: HIGH TEMPERATURE IN KITCHEN
KITCHEN POWER DISABLED
KITCHEN EMERGENCY LIGHT ON

Panel Temp: 27.30
Panel State: NORMAL

=====

```

Рисунок 3.34 - Температура кухні FIRE

Температура + дим кухні: WARNING і FIRE:

У цьому сценарії перевірялася комбінована небезпека в кухонній зоні. Підвищення температури разом із задимленням є більш серйозною ознакою, ніж одиничне короткочасне відхилення одного параметра.

При рівні WARNING система формує попередження. При досягненні рівня FIRE активується повний аварійний сценарій для кухні: вмикається основне світло кухні, формується сигнал відключення кухонної зони, вмикається аварійне освітлення та сирена

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						75
Зм.	Арк.	№ докум.	Підп.	Дата		

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 26.50
Room Smoke: 44
Room State: NORMAL

Kitchen Temp: 55.40
Kitchen Socket Temp: 15.40
Kitchen Smoke: 2382
Kitchen Gas: 0
Kitchen State: WARNING
WARNING: HIGH TEMPERATURE IN KITCHEN
WARNING: SMOKE DETECTED IN KITCHEN

Panel Temp: 23.90
Panel State: NORMAL

=====

```

Рисунок 3.35 - Температура + дим кухні WARNING

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 26.50
Room Smoke: 44
Room State: NORMAL

Kitchen Temp: 61.30
Kitchen Socket Temp: 15.40
Kitchen Smoke: 3851
Kitchen Gas: 0
Kitchen State: FIRE
FIRE: HIGH TEMPERATURE IN KITCHEN
FIRE: SMOKE FIRE LEVEL IN KITCHEN
KITCHEN POWER DISABLED
KITCHEN EMERGENCY LIGHT ON

Panel Temp: 23.90
Panel State: NORMAL

=====

```

Рисунок 3.36 - Температура + дим кухні FIRE

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						76
Зм.	Арк.	№ докум.	Підп.	Дата		

Газ у кухні: WARNING і FIRE:

Окремо перевірено сценарій витоку газу в кухонній зоні. При перевищенні попереджувального порогу система формує повідомлення про можливий витік газу, але не запускає повний аварійний сценарій.

Якщо рівень газу досягає критичного значення, система переходить у стан FIRE. У цьому випадку активується сирена, вмикається аварійне освітлення кухні та формується сигнал на відключення кухонної зони. Така логіка потрібна, оскільки витік газу є окремим небезпечним фактором, який може призвести до пожежі або вибухонебезпечної ситуації.

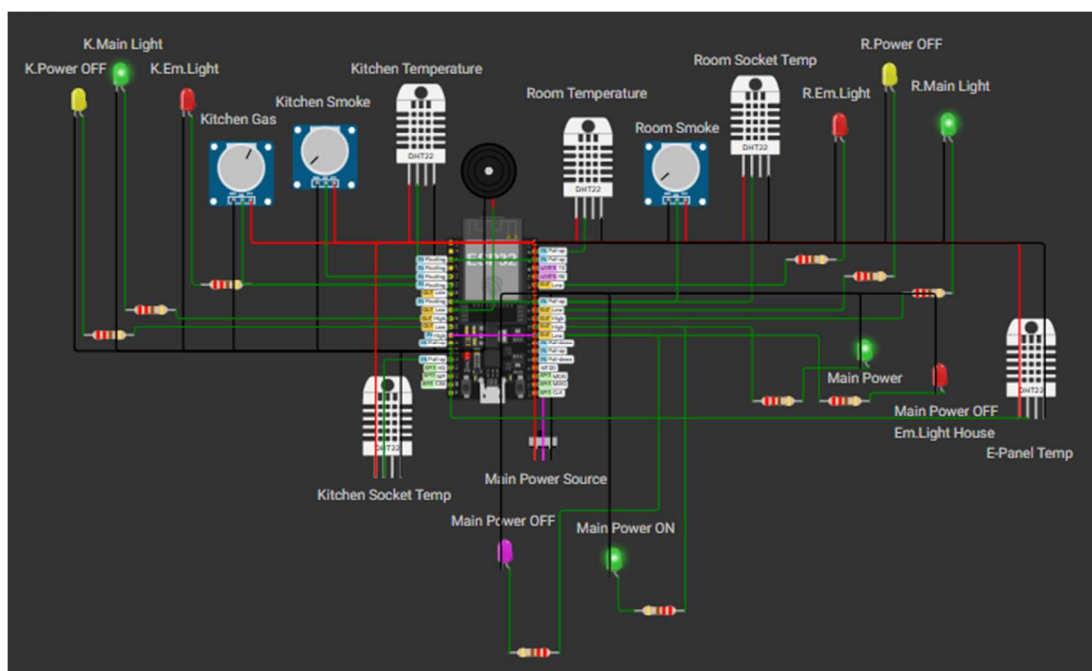


Рисунок 3.37 - Газ у кухні WARNING

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 29.90
Room Smoke: 44
Room State: NORMAL

Kitchen Temp: 27.30
Kitchen Socket Temp: 16.30
Kitchen Smoke: 0
Kitchen Gas: 2442
Kitchen State: WARNING
WARNING: GAS LEAK

Panel Temp: 27.30
Panel State: NORMAL
=====

```

Рисунок 3.38 - Газ у кухні WARNING

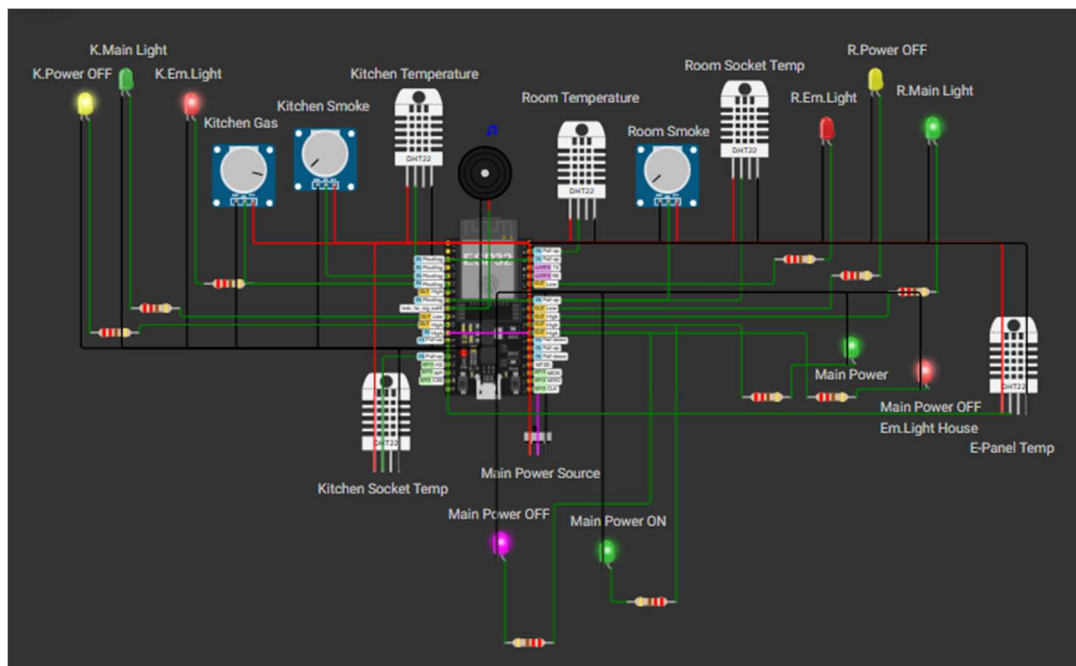


Рисунок 3.39 - Газ у кухні FIRE

```

Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 29.90
Room Smoke: 44
Room State: NORMAL

Kitchen Temp: 27.30
Kitchen Socket Temp: 16.30
Kitchen Smoke: 0
Kitchen Gas: 3627
Kitchen State: FIRE
FIRE: CRITICAL GAS LEAK
KITCHEN POWER DISABLED
SIREN ON
KITCHEN EMERGENCY LIGHT ON
KITCHEN POWER DISABLED
KITCHEN EMERGENCY LIGHT ON

Panel Temp: 27.30
Panel State: NORMAL

```

Рисунок 3.40 - Газ у кухні FIRE

Перегрів щитової: WARNING і FIRE:

Останнім було перевірено сценарій перегріву електрощитової. При досягненні попереджувального порогу система формує повідомлення про підвищену температуру в щитовій.

При переході щитової у стан FIRE система виконує глобальне аварійне реагування. У цьому випадку відключаються основні навантаження кімнати та кухні, вмикається загальне аварійне освітлення, активуються аварійні індикатори зон і запускається сирена. Такий сценарій пояснюється тим, що електрощитова є центральною частиною електроживлення об'єкта, тому її перегрів може становити небезпеку для всієї системи.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						79
Зм.	Арк.	№ докум.	Підп.	Дата		

```

===== SYSTEM STATUS =====
Main Power: PRESENT

Room Temp: 23.10
Room Socket Temp: 26.50
Room Smoke: 44
Room State: NORMAL

Kitchen Temp: 27.30
Kitchen Socket Temp: 15.40
Kitchen Smoke: 0
Kitchen Gas: 0
Kitchen State: NORMAL

Panel Temp: 57.90
Panel State: WARNING
WARNING: HIGH TEMPERATURE IN ELECTRICAL PANEL

=====

```

Рисунок 3.41 - Перегрів щитової WARNING

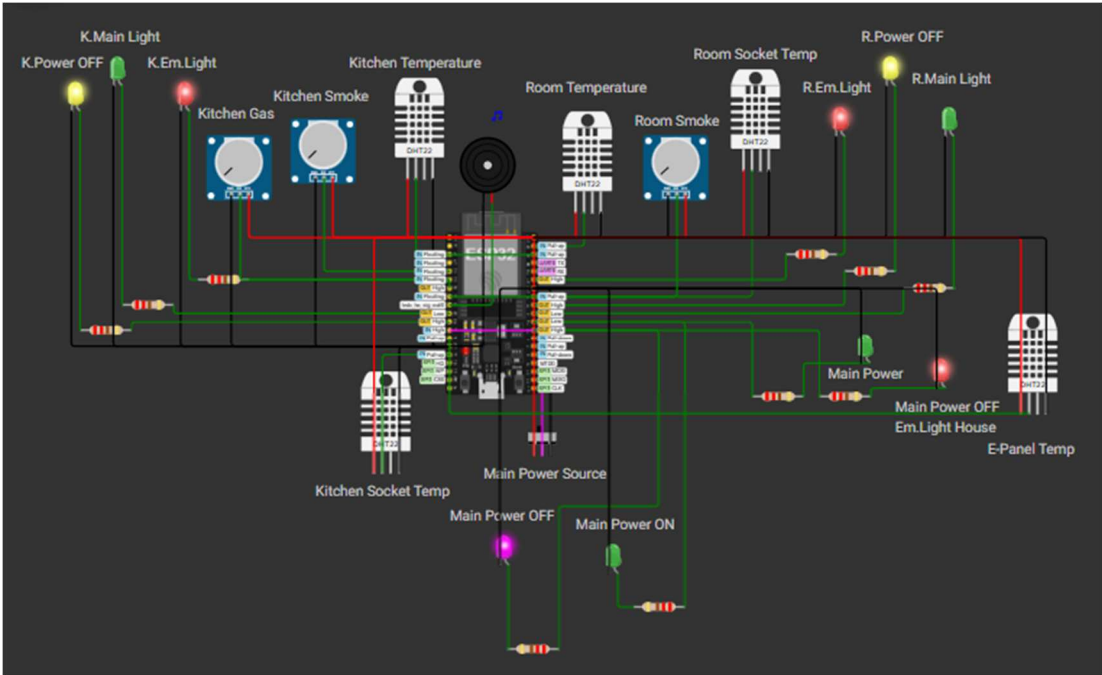


Рисунок 3.42 - Перегрів щитової FIRE

```

Panel Temp: 80.00
Panel State: FIRE
FIRE: ELECTRICAL PANEL OVERHEATING
GLOBAL POWER SHUTDOWN ACTIVE
ROOM POWER DISABLED
KITCHEN POWER DISABLED
ALL MAIN LIGHTS OFF
EMERGENCY LIGHTS ON
SIREN ON

```

Рисунок 3.43 - Перегрів щитової FIRE

Таким чином, виконані перевірки підтвердили працездатність основних сценаріїв системи. Модель правильно реагує на втрату основного живлення, підвищення температури, задимлення, витік газу, перегрів контрольованих електричних точок та перегрів електрощитової. Система коректно виконує зональне реагування для кімнати й кухні, а у випадку аварійного стану в електрощитовій переходить до загального аварійного відключення.

3.7 Інтеграція з хмарною платформою Blynk та Telegram-сповіщеннями.

Для розширення можливостей розробленої системи, реалізовано інтеграцію з хмарною платформою Blynk та месенджером Telegram. Ці засоби використовуються як додатковий рівень моніторингу та інформування користувача. Основна логіка безпеки при цьому залишається локальною: зчитування датчиків, аналіз станів, активація сирени, аварійного освітлення та команд зонального відключення виконується безпосередньо мікроконтролером ESP32. Хмарна платформа та месенджер не замінюють роботу контролера, а лише забезпечують зручне відображення інформації та оперативне отримання повідомлень.

У програмній частині для підключення до Blynk використовуються бібліотеки WiFi.h та BlynkSimpleEsp32.h. У коді задано ідентифікатор шаблону, назву проєкту та токен авторизації, після чого ESP32 підключається до Wi-Fi мережі Wokwi-GUEST і встановлює зв'язок із Blynk Cloud. Також використовується таймер BlynkTimer, який з інтервалом 2 секунди викликає функцію передавання даних до хмарної платформи.

```
#define BLYNK_TEMPLATE_ID "-----"
#define BLYNK_TEMPLATE_NAME "Fire Safety System"
#define BLYNK_AUTH_TOKEN "-----"

#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
```

У головному циклі програми виконуються команди Blynk.run() та blynkTimer.run(). Завдяки цьому підтримується з'єднання з хмарною платформою, а дані з ESP32 регулярно оновлюються на панелі користувача. Після цього система продовжує виконувати основну локальну логіку: зчитування датчиків, аналіз станів та керування виконавчими елементами.

```
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

BlynkTimer blynkTimer;

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

    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
    blynkTimer.setInterval(2000L, sendDataToBlynk);
```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						82
Зм.	Арк.	№ докум.	Підп.	Дата		

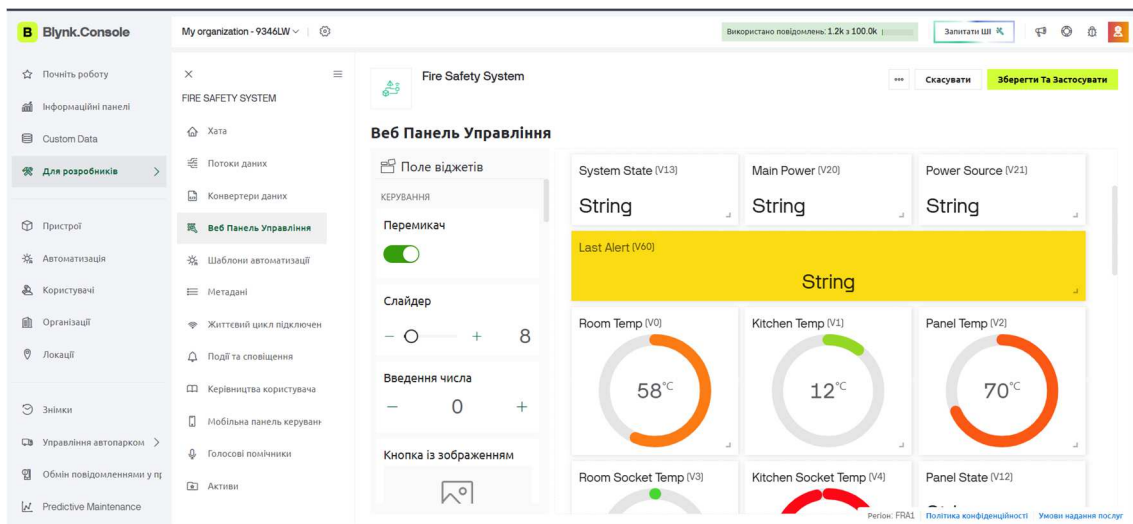


Рисунок 3.44 - середовище редагування в Blynk

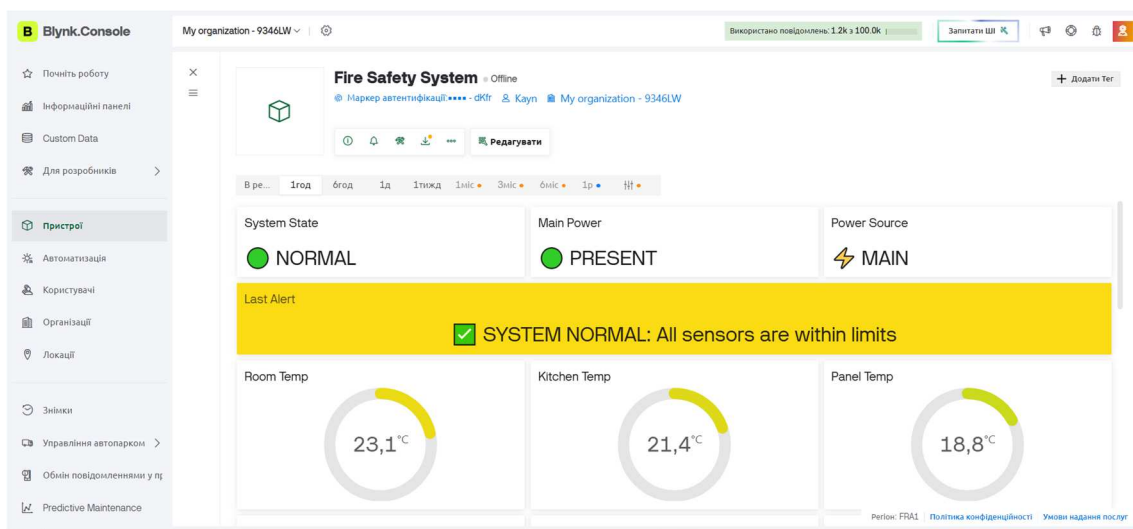


Рисунок 3.45 - середовище робочого стану в Blynk

Передавання даних до Blynk реалізовано за допомогою віртуальних пінів [12]. На панель користувача передаються значення температури в кімнаті, кухні та електрощитовій, температура біля контрольованих електричних точок, рівень диму в кімнаті та кухні, а також рівень газу в кухонній зоні. Крім цього, у Blynk відображаються стани окремих зон: кімнати, кухні, електрощитової та загальний стан системи. Для кращої наочності стани передаються у вигляді тексту з емодзі, наприклад 🟢 NORMAL, 🟡 WARNING, 🔴 FIRE.

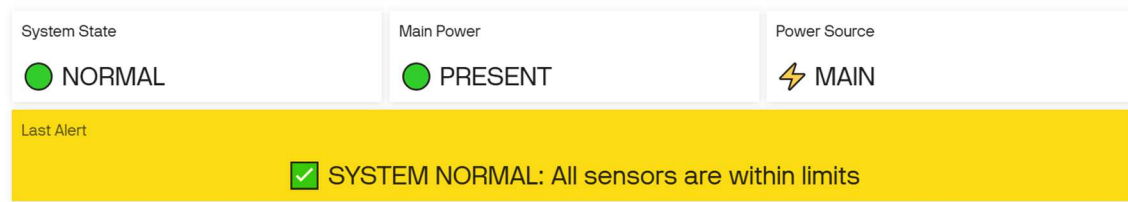


Рисунок 3.46 - Стани системи в Blynk

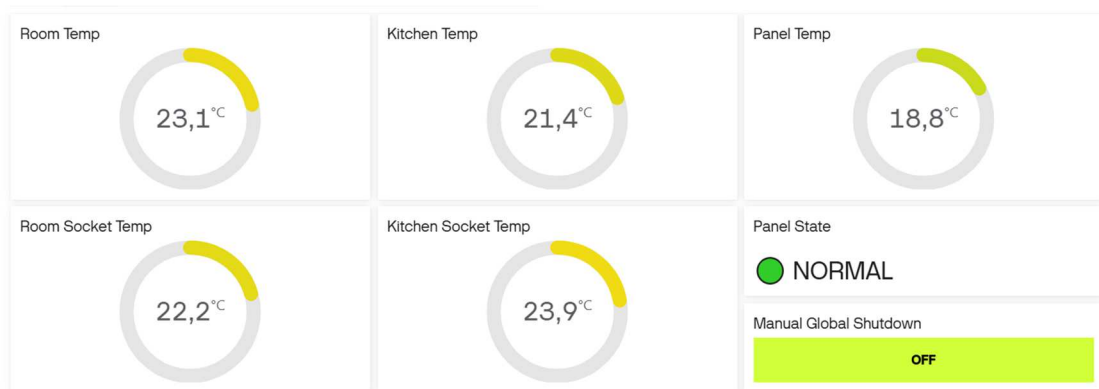


Рисунок 3.47 - Дані датчиків

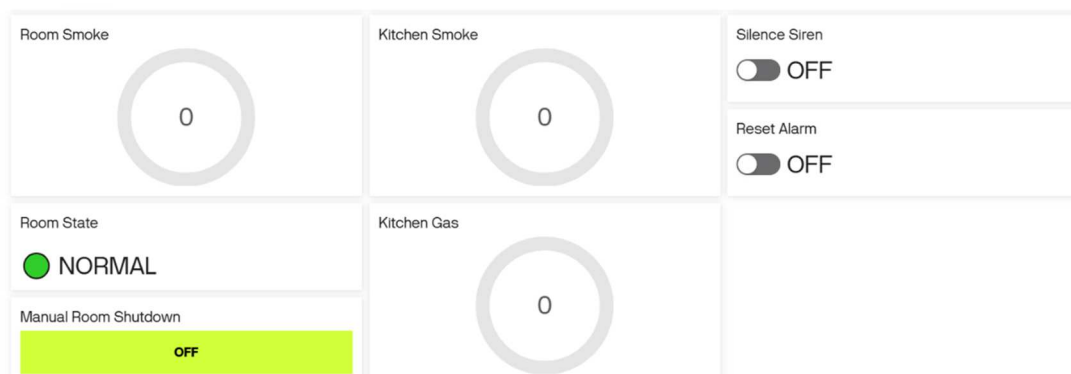


Рисунок 3.48 - Дані датчиків

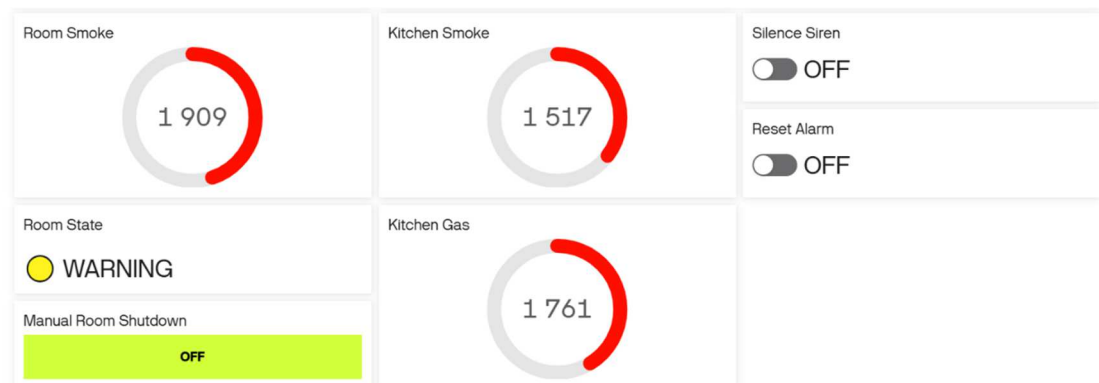


Рисунок 3.49 - Дані датчиків які отримали сигнал і передали в середовище Blynk

```

void sendDataToBlynk() {
  Blynk.virtualWrite(V0, roomTemp);
  Blynk.virtualWrite(V1, kitchenTemp);
  Blynk.virtualWrite(V2, panelTemp);

  Blynk.virtualWrite(V3, roomSocketTemp);
  Blynk.virtualWrite(V4, kitchenSocketTemp);

  Blynk.virtualWrite(V5, roomSmoke);
  Blynk.virtualWrite(V6, kitchenSmoke);
  Blynk.virtualWrite(V7, kitchenGas);

  Blynk.virtualWrite(V10, stateToEmojiText(roomState));
  Blynk.virtualWrite(V11, stateToEmojiText(kitchenState));
  Blynk.virtualWrite(V12, stateToEmojiText(panelState));

  ZoneState systemState = getSystemState();
  Blynk.virtualWrite(V13, stateToEmojiText(systemState));
}

```

Окремо в інтерфейсі Blynk передбачено відображення стану живлення. Якщо основне живлення присутнє, на панель передається статус PRESENT та джерело MAIN. Якщо основне живлення втрачено, система виводить статус LOST і позначає джерело як BACKUP. Це дозволяє користувачу дистанційно бачити, чи працює система від основної мережі, чи перейшла на резервне живлення.

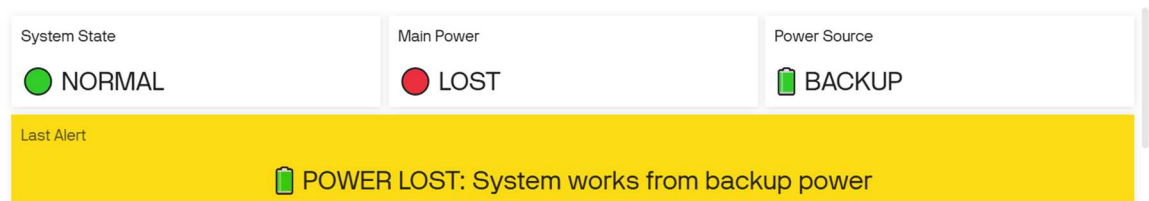


Рисунок 3.50 - Стан живлення в Blynk

```

if (mainPowerPresent) {
  Blynk.virtualWrite(V20, "● PRESENT");
  Blynk.virtualWrite(V21, "⚡ MAIN");
} else {
  Blynk.virtualWrite(V20, "● LOST");
  Blynk.virtualWrite(V21, "🔋 BACKUP");
}

```

Для відображення актуального повідомлення про стан системи використовується віртуальний пін V60. У нього передається текст

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						85
Зм.	Арк.	№ докум.	Підп.	Дата		

останнього важливого сповіщення: нормальний стан системи, втрата живлення, попередження, пожежа, витік газу або вимкнення сирени. Такий підхід зручний тим, що користувач бачить не лише окремі числові значення, а й короткий зрозумілий опис поточної ситуації.

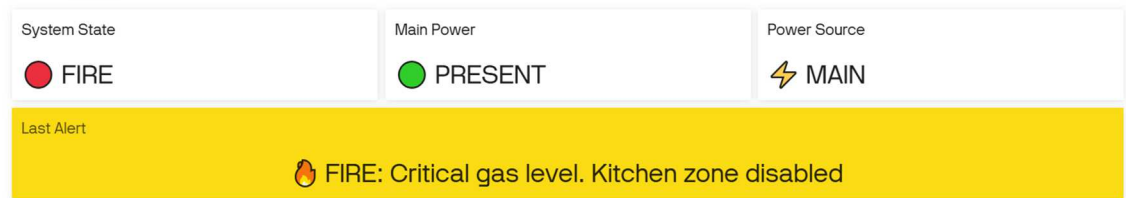


Рисунок 3.51 - Система відображення повідомлень

У програмі передбачено функцію `setCurrentAlert()`, яка оновлює повідомлення у Blynk і, за потреби, надсилає його в Telegram. Щоб уникнути дублювання однакових повідомлень, функція перевіряє, чи не збігається новий текст із попереднім. Якщо повідомлення не змінилося, повторне надсилання не виконується. Це захищає користувача від спаму однаковими сповіщеннями.

```
void setCurrentAlert(String message, bool sendToTelegram =
true) {
    if (message == lastAlertMessage) {
        return;
    }

    lastAlertMessage = message;

    Blynk.virtualWrite(V60, message);

    if (sendToTelegram) {
        sendTelegramMessage(message);
    }
}
```

Формування актуального повідомлення виконується у функції `updateCurrentAlert()`. Вона послідовно перевіряє основні стани системи: втрату живлення, пожежу в електрощитовій, критичний рівень газу, пожежу на кухні, пожежу в кімнаті, попередження про газ, попередження в щитовій, кухні або кімнаті. Якщо жодної небезпеки немає, у Blynk передається

повідомлення про нормальний стан системи. Така послідовність важлива, оскільки найбільш небезпечні стани мають вищий пріоритет.

```
void updateCurrentAlert() {
    if (!mainPowerPresent) {
        setCurrentAlert("🔌 POWER LOST: System works from backup power");
        return;
    }

    if (panelState == FIRE) {
        setCurrentAlert("🔥 FIRE: Electrical panel overheating. Global shutdown active");
        return;
    }

    if (kitchenGas >= GAS_FIRE_LEVEL) {
        setCurrentAlert("🔥 FIRE: Critical gas level. Kitchen zone disabled");
        return;
    }

    if (kitchenState == FIRE) {
        setCurrentAlert("🔥 FIRE: Kitchen zone disabled, emergency light ON");
        return;
    }

    if (roomState == FIRE) {
        setCurrentAlert("🔥 FIRE: Room zone disabled, emergency light ON");
        return;
    }

    setCurrentAlert("✅ SYSTEM NORMAL: All sensors are within limits");
}
```

Крім відображення станів, у Blynk реалізовано кнопку вимкнення сирени. Для цього використовується віртуальний пін V33. При натисканні кнопки змінна `sirenMuted` переходить у значення `true`, після чого сирена вимикається, навіть якщо система залишається у стані `FIRE`. При цьому аварійне освітлення та відключення небезпечної зони залишаються активними. Така функція потрібна для ситуацій, коли аварійна подія вже зафіксована, але тривале звучання сирени заважає подальшим діям користувача.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						87
Зм.	Арк.	№ докум.	Підп.	Дата		

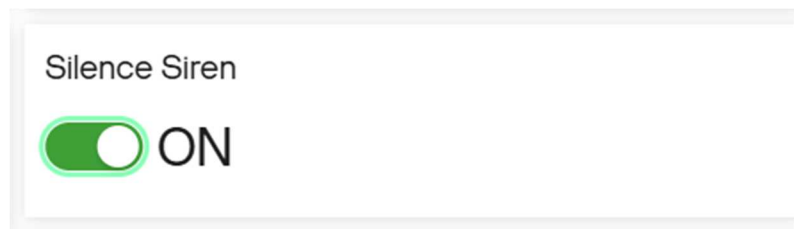


Рисунок 3.52 - Кнопка вимкнення сирени

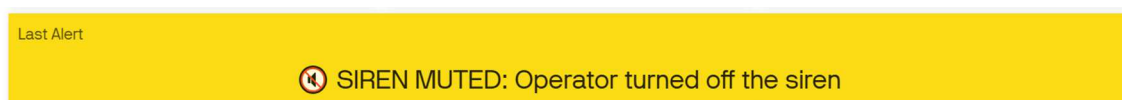


Рисунок 3.53 - Сповіщення про вимкнену сирену

```
BLYNK_WRITE(V33) {
  int value = param.asInt();

  if (value == 1) {
    sirenMuted = true;
  }
}
```

Telegram у розробленій системі використовується як додатковий канал оперативного сповіщення. Для цього було створено Telegram-бота через BotFather, отримано токен доступу та Chat ID користувача. У програмі для надсилення повідомлень використовуються бібліотеки HTTPClient.h та WiFiClientSecure.h, які дозволяють ESP32 сформувати HTTPS-запит до Telegram Bot API.

```
#include <HTTPClient.h>
#include <WiFiClientSecure.h>

const char* TELEGRAM_BOT_TOKEN = "-----";
const char* TELEGRAM_CHAT_ID = "-----";
```

Надсилення повідомлення виконується функцією sendTelegramMessage(). Вона перевіряє, чи не надсилалося таке саме повідомлення раніше, після чого формує URL-запит до Telegram API та виконує його за допомогою HTTPClient. У Serial Monitor додатково виводиться HTTP-код відповіді, що дозволяє перевірити успішність надсилення повідомлення.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						88
Зм.	Арк.	№ докум.	Підп.	Дата		

```

void sendTelegramMessage(String message) {
    if (message == lastTelegramMessage) {
        return;
    }

    lastTelegramMessage = message;

    WiFiClientSecure client;
    client.setInsecure();

    HTTPClient https;

    String url = "https://api.telegram.org/bot";
    url += TELEGRAM_BOT_TOKEN;
    url += "/sendMessage?chat_id=";
    url += TELEGRAM_CHAT_ID;
    url += "&text=";
    url += urlEncode(message);

    if (https.begin(client, url)) {
        int httpCode = https.GET();

        Serial.print("Telegram HTTP code: ");
        Serial.println(httpCode);

        https.end();
    } else {
        Serial.println("Telegram connection failed");
    }
}

```

Під час роботи системи Telegram-повідомлення надсилаються при зміні актуального стану. Наприклад, користувач може отримати повідомлення про втрату основного живлення, пожежу в електрощитовій, критичний рівень газу, аварійну ситуацію на кухні або в кімнаті, а також про повернення системи до нормального стану. Це дозволяє контролювати стан об'єкта навіть тоді, коли користувач не перебуває біля комп'ютера або панелі Blynk.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						89
Зм.	Арк.	№ докум.	Підп.	Дата		

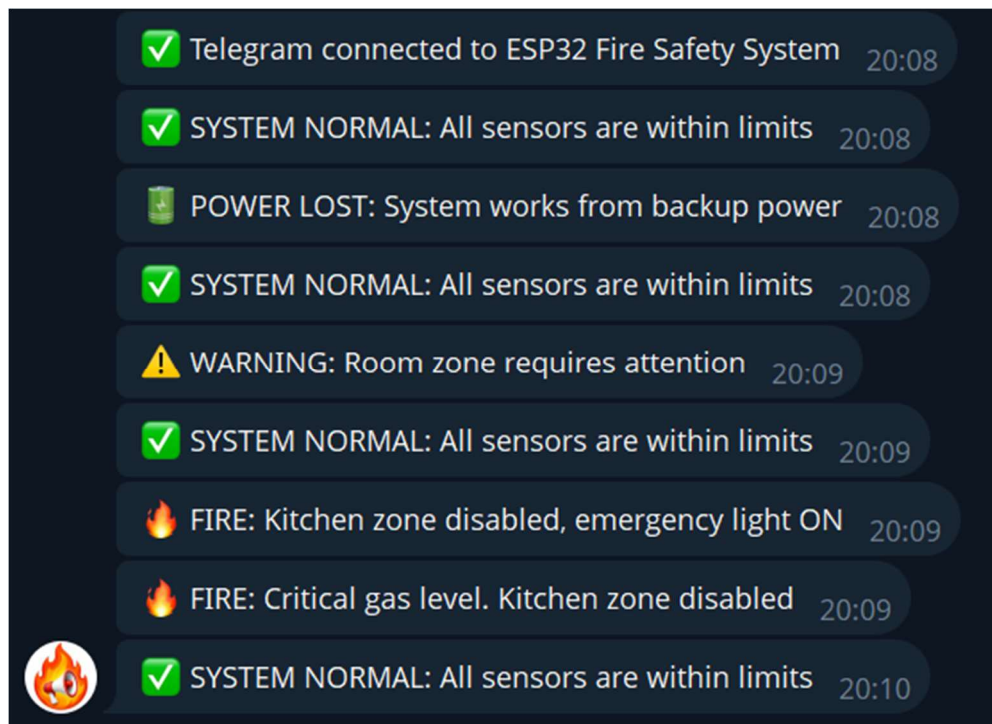


Рисунок 3.54 - Сповіщення від Telegram_bot

Важливо зазначити, що Blynk і Telegram не є основою прийняття рішень у системі. Усі основні дії - зчитування датчиків, визначення стану, увімкнення сирени, аварійного освітлення та формування сигналів зонального відключення - виконуються локально на ESP32. Тому при втраті інтернет-з'єднання система зберігає базову працездатність, хоча дистанційне відображення даних і Telegram-сповіщення можуть бути недоступними.

Таким чином, інтеграція з Blynk та Telegram дозволила зробити систему більш інформативною і зручною для користувача. Blynk забезпечує візуалізацію параметрів і станів у реальному часі, а Telegram виконує роль оперативного каналу сповіщень про важливі події. Разом вони доповнюють локальну логіку ESP32 і завершують побудову комп'ютерно-інтегрованої системи протипожежного моніторингу та реагування.

ВИСНОВКИ

У даній роботі розроблено комп'ютерно-інтегровану систему захисту від пожеж, яка поєднує елементи протипожежного моніторингу, зонального керування та дистанційного інформування користувача.

У першому розділі проаналізовано функціональне призначення сучасних систем пожежної сигналізації та протипожежного моніторингу. Встановлено, що сучасні системи повинні не лише виявляти ознаки пожежі, а й забезпечувати своєчасне оповіщення, контроль окремих зон, взаємодію з виконавчими пристроями та передачу інформації користувачу. Окрему увагу було приділено комп'ютерно-інтегрованим системам, які дозволяють поєднувати сенсорний контроль, програмну обробку даних, локальне реагування та хмарний моніторинг.

У другому розділі обґрунтовано вибір структури системи та технічних засобів для її реалізації. Визначено основні контрольовані зони об'єкта: житлова кімната, кухня, електрощитова та контрольовані електричні точки підвищеного ризику. Для кожної зони враховано характерні небезпечні фактори, зокрема підвищення температури, задимлення, витік газу та локальний перегрів електричних з'єднань. Центральним вузлом обробки даних обрано мікроконтролер ESP32, який забезпечує зчитування показників сенсорів, аналіз станів і керування виконавчими елементами.

У роботі використано цифрові сенсори температури DHT22 для моделювання теплового каналу контролю, а також аналогові сигнали для імітації каналів диму та газу. Для відображення роботи виконавчих пристроїв у середовищі моделювання застосовано LED-індикатори, які показують роботу основного та аварійного освітлення, а також факт формування команд на зональне відключення живлення. Звукове оповіщення реалізовано за допомогою бузера, який активується у разі переходу системи до аварійного стану.

У третьому розділі розроблено та змодельовано програмно-алгоритмічну частину системи. Програмна реалізація виконана мовою C++

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						91
Зм.	Арк.	№ докум.	Підп.	Дата		

з використанням Arduino Framework для ESP32. У програмі реалізовано циклічне зчитування даних із сенсорів, порівняння значень із порогами, визначення станів NORMAL, WARNING та FIRE, а також керування виконавчими пристроями залежно від рівня небезпеки.

Особливістю розробленої системи є зональне реагування. У разі аварійного стану в кімнаті або кухні система формує команду на відключення саме відповідної зони, активує аварійне освітлення та вмикає сирену. Якщо ж критичний стан виникає в електрощитовій, система переходить до глобального аварійного реагування, оскільки ця зона пов'язана з живленням усього об'єкта. Також у системі передбачено контроль втрати основного живлення з переходом до аварійного режиму освітлення.

Для перевірки працездатності системи створено модель у середовищі Wokwi. У ході моделювання були перевірені основні сценарії роботи: нормальний стан усіх датчиків, втрата основного живлення, задимлення кімнати, підвищення температури в кімнаті та кухні, перегрів контрольованих електричних точок, витік газу на кухні, комбіновані сценарії “температура + дим”, а також перегрів електрощитової. Результати перевірки показали, що система коректно переходить між станами, активує відповідні індикатори, сирену та аварійне освітлення, а також формує сигнали зонального або глобального відключення залежно від місця виникнення небезпеки.

Також реалізовано інтеграцію з хмарною платформою Blynk та Telegram-сповіщеннями. Blynk використовується для дистанційного відображення поточних значень датчиків, станів зон, стану живлення та актуального повідомлення системи. Telegram застосовується як додатковий канал оперативного інформування користувача про важливі події: попередження, пожежу, витік газу, втрату основного живлення або повернення системи до нормального стану. При цьому основна логіка безпеки залишається локальною і виконується безпосередньо на ESP32.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						92
Зм.	Арк.	№ докум.	Підп.	Дата		

У результаті виконання дипломної роботи було розроблено навчально-дослідний прототип комп'ютерно-інтегрованої системи захисту від пожеж. Запропоноване рішення поєднує моніторинг параметрів середовища, програмну оцінку стану зон, локальне аварійне реагування, зональне керування та дистанційне інформування користувача. Отримана модель не є сертифікованою промисловою системою пожежної сигналізації, однак вона демонструє принципи побудови сучасних інтелектуальних систем протипожежного моніторингу та може бути використана як основа для подальшого вдосконалення.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						93
Зм.	Арк.	№ докум.	Підп.	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДБН В.2.5-56:2014. Системи протипожежного захисту. Київ: Мінрегіон України.
2. ДСТУ EN 54-1:2014. Системи пожежної сигналізації. Частина 1. Вступ.
3. ДСТУ EN 54-2:2003. Системи пожежної сигналізації. Частина 2. Прилади приймально-контрольні пожежні.
4. ДСТУ EN 54-7:2019. Системи пожежної сигналізації. Частина 7. Сповіщувачі пожежні димові точкові розсіяного світла, пропускнуго світла чи іонізаційні.
5. Espressif Systems. ESP32 Series Datasheet.
6. Espressif Systems. ESP32 Wi-Fi & Bluetooth SoC: official product page.
7. Espressif Systems. Arduino Core for ESP32 Documentation.
8. Arduino. digitalWrite() - Arduino Documentation.
9. Aosong Electronics. DHT22 / AM2302 Digital-output relative humidity and temperature sensor/module: Datasheet.
10. Wokwi. DHT22 Sensor Reference.
11. Hanwei Electronics. MQ-2 Semiconductor Sensor for Combustible Gas: Datasheet.
12. Blynk. Virtual Pins - Blynk Documentation.
13. Blynk. Blynk IoT Platform: official website/documentation.
14. Telegram. Telegram Bot API.
15. Wokwi. ESP32, STM32, Arduino Simulator - online simulation environment.
16. Arduino ESP32 Core. Getting Started with Arduino ESP32.
17. Handson Technology. MQ-2 Gas Sensor Module Datasheet.
18. RS Components. DHT22 Temperature and Humidity Sensor Datasheet.

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						94
Зм.	Арк.	№ докум.	Підп.	Дата		

ДОДАТКИ

Додаток А: Програмний код системи

```
#define BLYNK_TEMPLATE_ID "TMPL4yUTrZC8D"
#define BLYNK_TEMPLATE_NAME "Fire Safety System"
#define BLYNK_AUTH_TOKEN "F6FgKGocbleo0mcemB4anu8pZBZNdKfr"
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <HTTPClient.h>
#include <WiFiClientSecure.h>
#include <DHT.h>

// DHT SETTINGS
#define DHT_TYPE DHT22

#define PANEL_TEMP_PIN 12
#define KITCHEN_SOCKET_TEMP_PIN 13
#define ROOM_SOCKET_TEMP_PIN 19
#define KITCHEN_TEMP_PIN 22
#define ROOM_TEMP_PIN 23

DHT panelDHT(PANEL_TEMP_PIN, DHT_TYPE);
DHT kitchenSocketDHT(KITCHEN_SOCKET_TEMP_PIN, DHT_TYPE);
DHT roomSocketDHT(ROOM_SOCKET_TEMP_PIN, DHT_TYPE);
DHT kitchenDHT(KITCHEN_TEMP_PIN, DHT_TYPE);
DHT roomDHT(ROOM_TEMP_PIN, DHT_TYPE);

// PIN MAP
// Main power input
#define MAIN_POWER_SOURCE_PIN 14

// Main power indicators
#define WHOLE_HOUSE_EMERGENCY_LIGHT_PIN 16
#define MAIN_POWER_OK_LIGHT_PIN 17

// Room outputs
#define ROOM_MAIN_LIGHT_PIN 5
#define ROOM_POWER_OFF_PIN 18
#define ROOM_EMERGENCY_LIGHT_PIN 21

// Kitchen outputs
#define KITCHEN_MAIN_LIGHT_PIN 26
#define KITCHEN_POWER_OFF_PIN 27
#define KITCHEN_EMERGENCY_LIGHT_PIN 32

// Siren
#define SIREN_PIN 25

// Analog sensors
#define ROOM_SMOKE_PIN 33
```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						95
Зм.	Арк.	№ докум.	Підп.	Дата		

```

#define KITCHEN_SMOKE_PIN          34
#define KITCHEN_GAS_PIN            35

// CONFIG
const int MAIN_POWER_PRESENT_LEVEL = HIGH;

// Temperature thresholds
const float ROOM_TEMP_WARNING = 45.0;
const float ROOM_TEMP_FIRE    = 60.0;

const float KITCHEN_TEMP_WARNING = 45.0;
const float KITCHEN_TEMP_FIRE    = 60.0;

const float PANEL_TEMP_WARNING = 55.0;
const float PANEL_TEMP_FIRE    = 75.0;

const float ROOM_SOCKET_TEMP_WARNING = 55.0;
const float ROOM_SOCKET_TEMP_FIRE    = 70.0;

const float KITCHEN_SOCKET_TEMP_WARNING = 55.0;
const float KITCHEN_SOCKET_TEMP_FIRE    = 70.0;

// Analog thresholds: 0-4095 для ESP32
const int SMOKE_WARNING_LEVEL = 1500;
const int SMOKE_FIRE_LEVEL    = 2800;

const int GAS_WARNING_LEVEL = 1500;
const int GAS_FIRE_LEVEL    = 2800;

// Siren
const int SIREN_FREQUENCY = 1000;

// Serial update interval
const unsigned long STATUS_INTERVAL = 2000;
unsigned long lastStatusTime = 0;

// SYSTEM STATES
enum ZoneState {
    NORMAL,
    WARNING,
    FIRE
};

ZoneState roomState = NORMAL;
ZoneState kitchenState = NORMAL;
ZoneState panelState = NORMAL;
ZoneState roomSocketState = NORMAL;
ZoneState kitchenSocketState = NORMAL;

// Blynk Wi-Fi
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						96
Зм.	Арк.	№ докум.	Підп.	Дата		

```

BlynkTimer blynkTimer;

// Telegram settings
const char* TELEGRAM_BOT_TOKEN =
"8793888923:AAGw91ZT9b2eAKOynx2cwVQ7ecBtq5qUpAw";
const char* TELEGRAM_CHAT_ID = "1493945817";

String lastTelegramMessage = "";
String lastAlertMessage = "";

// Ручне керування з Blynk
bool manualRoomShutdown = false;
bool manualKitchenShutdown = false;
bool manualGlobalShutdown = false;
bool sirenMuted = false;

// Попередні стани для захисту від спаму повідомлень
ZoneState lastRoomState = NORMAL;
ZoneState lastKitchenState = NORMAL;
ZoneState lastPanelState = NORMAL;
ZoneState lastSystemState = NORMAL;

bool lastMainPowerPresent = true;

// SENSOR VALUES

float panelTemp = NAN;
float kitchenSocketTemp = NAN;
float roomSocketTemp = NAN;
float kitchenTemp = NAN;
float roomTemp = NAN;

int roomSmoke = 0;
int kitchenSmoke = 0;
int kitchenGas = 0;

bool mainPowerPresent = true;

// HELPER FUNCTIONS
const char* stateToText(ZoneState state) {
    switch (state) {
        case NORMAL:    return "NORMAL";
        case WARNING:   return "WARNING";
        case FIRE:      return "FIRE";
        default:        return "UNKNOWN";
    }
}

ZoneState getTempState(float temp, float warningLevel, float
fireLevel) {
    if (isnan(temp)) {
        return NORMAL;
    }
}

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						97
Зм.	Арк.	№ докум.	Підп.	Дата		

```

    if (temp >= fireLevel) {
        return FIRE;
    }

    if (temp >= warningLevel) {
        return WARNING;
    }

    return NORMAL;
}

ZoneState getAnalogState(int value, int warningLevel, int
fireLevel) {
    if (value >= fireLevel) {
        return FIRE;
    }

    if (value >= warningLevel) {
        return WARNING;
    }

    return NORMAL;
}

ZoneState maxState(ZoneState a, ZoneState b) {
    if (a == FIRE || b == FIRE) {
        return FIRE;
    }

    if (a == WARNING || b == WARNING) {
        return WARNING;
    }

    return NORMAL;
}

void printTempValue(const char* label, float value) {
    Serial.print(label);

    if (isnan(value)) {
        Serial.println("nan");
    } else {
        Serial.println(value);
    }
}

// SETUP
void setup() {
    Serial.begin(115200);

    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
    blynkTimer.setInterval(2000L, sendDataToBlynk);
}

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						98
Зм.	Арк.	№ докум.	Підп.	Дата		

```

    sendTelegramMessage("✅ Telegram connected to ESP32 Fire
    Safety System");

    panelDHT.begin();
    kitchenSocketDHT.begin();
    roomSocketDHT.begin();
    kitchenDHT.begin();
    roomDHT.begin();

    pinMode(MAIN_POWER_SOURCE_PIN, INPUT);

    pinMode(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, OUTPUT);
    pinMode(MAIN_POWER_OK_LIGHT_PIN, OUTPUT);

    pinMode(ROOM_MAIN_LIGHT_PIN, OUTPUT);
    pinMode(ROOM_POWER_OFF_PIN, OUTPUT);
    pinMode(ROOM_EMERGENCY_LIGHT_PIN, OUTPUT);

    pinMode(KITCHEN_MAIN_LIGHT_PIN, OUTPUT);
    pinMode(KITCHEN_POWER_OFF_PIN, OUTPUT);
    pinMode(KITCHEN_EMERGENCY_LIGHT_PIN, OUTPUT);

    pinMode(SIREN_PIN, OUTPUT);

    pinMode(ROOM_SMOKE_PIN, INPUT);
    pinMode(KITCHEN_SMOKE_PIN, INPUT);
    pinMode(KITCHEN_GAS_PIN, INPUT);

    noTone(SIREN_PIN);

    Serial.println("Fire Safety System Started");
}

// MAIN LOOP
void loop() {
    Blynk.run();
    blynkTimer.run();

    readSensors();
    analyzeSystem();
    controlOutputs();

    if (millis() - lastStatusTime >= STATUS_INTERVAL) {
        lastStatusTime = millis();
        printStatus();
    }
}

// READ SENSORS
void readSensors() {
    mainPowerPresent = digitalRead(MAIN_POWER_SOURCE_PIN) ==
    MAIN_POWER_PRESENT_LEVEL;

    panelTemp = panelDHT.readTemperature();

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						99
Зм.	Арк.	№ докум.	Підп.	Дата		

```

kitchenSocketTemp = kitchenSocketDHT.readTemperature();
roomSocketTemp = roomSocketDHT.readTemperature();
kitchenTemp = kitchenDHT.readTemperature();
roomTemp = roomDHT.readTemperature();

roomSmoke = analogRead(ROOM_SMOKE_PIN);
kitchenSmoke = analogRead(KITCHEN_SMOKE_PIN);
kitchenGas = analogRead(KITCHEN_GAS_PIN);
}

// ANALYZE SYSTEM
void analyzeSystem() {

    // ROOM
    ZoneState roomTempState = getTempState(
        roomTemp,
        ROOM_TEMP_WARNING,
        ROOM_TEMP_FIRE
    );

    roomSocketState = getTempState(
        roomSocketTemp,
        ROOM_SOCKET_TEMP_WARNING,
        ROOM_SOCKET_TEMP_FIRE
    );

    ZoneState roomSmokeState = getAnalogState(
        roomSmoke,
        SMOKE_WARNING_LEVEL,
        SMOKE_FIRE_LEVEL
    );

    roomState = NORMAL;
    roomState = maxState(roomState, roomTempState);
    roomState = maxState(roomState, roomSocketState);
    roomState = maxState(roomState, roomSmokeState);

    // KITCHEN
    ZoneState kitchenTempState = getTempState(
        kitchenTemp,
        KITCHEN_TEMP_WARNING,
        KITCHEN_TEMP_FIRE
    );

    kitchenSocketState = getTempState(
        kitchenSocketTemp,
        KITCHEN_SOCKET_TEMP_WARNING,
        KITCHEN_SOCKET_TEMP_FIRE
    );

    ZoneState kitchenSmokeState = getAnalogState(
        kitchenSmoke,
        SMOKE_WARNING_LEVEL,
        SMOKE_FIRE_LEVEL
    );

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						00
Зм.	Арк.	№ докум.	Підп.	Дата		

```

);

ZoneState kitchenGasState = getAnalogState(
    kitchenGas,
    GAS_WARNING_LEVEL,
    GAS_FIRE_LEVEL
);

kitchenState = NORMAL;
kitchenState = maxState(kitchenState, kitchenTempState);
kitchenState = maxState(kitchenState, kitchenSocketState);
kitchenState = maxState(kitchenState, kitchenSmokeState);
kitchenState = maxState(kitchenState, kitchenGasState);

// PANEL
panelState = getTempState(
    panelTemp,
    PANEL_TEMP_WARNING,
    PANEL_TEMP_FIRE
);
}

//CONTROL OUTPUTS
void controlOutputs() {
    bool roomFire = roomState == FIRE;
    bool kitchenFire = kitchenState == FIRE;
    bool panelFire = panelState == FIRE;

    bool anyFire = roomFire || kitchenFire || panelFire;

    // MAIN POWER INDICATION
    if (mainPowerPresent && !panelFire) {
        digitalWrite(MAIN_POWER_OK_LIGHT_PIN, HIGH);
    } else {
        digitalWrite(MAIN_POWER_OK_LIGHT_PIN, LOW);
    }

    if (!mainPowerPresent || anyFire) {
        digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, HIGH);
    } else {
        digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, LOW);
    }

    // PANEL FIRE GLOBAL SHUTDOWN
    if (panelFire) {
        digitalWrite(ROOM_MAIN_LIGHT_PIN, LOW);
        digitalWrite(KITCHEN_MAIN_LIGHT_PIN, LOW);

        digitalWrite(ROOM_POWER_OFF_PIN, HIGH);
        digitalWrite(KITCHEN_POWER_OFF_PIN, HIGH);

        digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, HIGH);
        digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, HIGH);
        digitalWrite(WHOLE_HOUSE_EMERGENCY_LIGHT_PIN, HIGH);
    }
}

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						01
Зм.	Арк.	№ докум.	Підп.	Дата		

```

    tone(SIREN_PIN, SIREN_FREQUENCY);
    return;
}

// ROOM OUTPUTS
if (roomFire || !mainPowerPresent) {
    digitalWrite(ROOM_MAIN_LIGHT_PIN, LOW);
} else {
    digitalWrite(ROOM_MAIN_LIGHT_PIN, HIGH);
}

if (roomFire) {
    digitalWrite(ROOM_POWER_OFF_PIN, HIGH);
    digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, HIGH);
} else {
    digitalWrite(ROOM_POWER_OFF_PIN, LOW);

    if (mainPowerPresent) {
        digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, LOW);
    } else {
        digitalWrite(ROOM_EMERGENCY_LIGHT_PIN, HIGH);
    }
}

// KITCHEN OUTPUTS
if (kitchenFire || !mainPowerPresent) {
    digitalWrite(KITCHEN_MAIN_LIGHT_PIN, LOW);
} else {
    digitalWrite(KITCHEN_MAIN_LIGHT_PIN, HIGH);
}

if (kitchenFire) {
    digitalWrite(KITCHEN_POWER_OFF_PIN, HIGH);
    digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, HIGH);
} else {
    digitalWrite(KITCHEN_POWER_OFF_PIN, LOW);

    if (mainPowerPresent) {
        digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, LOW);
    } else {
        digitalWrite(KITCHEN_EMERGENCY_LIGHT_PIN, HIGH);
    }
}

// SIREN
// Сирена НЕ вмикається від втрати основного живлення.
// Сирена вмикається тільки при FIRE.
if (anyFire && !sirenMuted) {
    tone(SIREN_PIN, SIREN_FREQUENCY);
} else {
    noTone(SIREN_PIN);
}

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						02
Зм.	Арк.	№ докум.	Підп.	Дата		

```

if (!anyFire) {
    sirenMuted = false;
}
}

// PRINT STATUS
void printStatus() {
    Serial.println();
    Serial.println("===== SYSTEM STATUS =====");

    Serial.print("Main Power: ");
    Serial.println(mainPowerPresent ? "PRESENT" : "LOST");

    Serial.println();

    // ROOM
    printTempValue("Room Temp: ", roomTemp);
    printTempValue("Room Socket Temp: ", roomSocketTemp);

    Serial.print("Room Smoke: ");
    Serial.println(roomSmoke);

    Serial.print("Room State: ");
    Serial.println(stateToText(roomState));

    if (roomState == WARNING) {
        if (!isnan(roomTemp) && roomTemp >= ROOM_TEMP_WARNING &&
roomTemp < ROOM_TEMP_FIRE) {
            Serial.println("WARNING: HIGH TEMPERATURE IN ROOM");
        }

        if (!isnan(roomSocketTemp) && roomSocketTemp >=
ROOM_SOCKET_TEMP_WARNING && roomSocketTemp <
ROOM_SOCKET_TEMP_FIRE) {
            Serial.println("WARNING: ROOM SOCKET OVERHEATING");
        }

        if (roomSmoke >= SMOKE_WARNING_LEVEL && roomSmoke <
SMOKE_FIRE_LEVEL) {
            Serial.println("WARNING: SMOKE DETECTED IN ROOM");
        }
    }

    if (roomState == FIRE) {
        if (!isnan(roomTemp) && roomTemp >= ROOM_TEMP_FIRE) {
            Serial.println("FIRE: HIGH TEMPERATURE IN ROOM");
        }

        if (!isnan(roomSocketTemp) && roomSocketTemp >=
ROOM_SOCKET_TEMP_FIRE) {
            Serial.println("FIRE: ROOM SOCKET OVERHEATING / POSSIBLE
SHORT CIRCUIT");
        }
    }
}

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						03
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        if (roomSmoke >= SMOKE_FIRE_LEVEL) {
            Serial.println("FIRE: SMOKE FIRE LEVEL IN ROOM");
        }

        Serial.println("ROOM POWER DISABLED");
        Serial.println("ROOM EMERGENCY LIGHT ON");
    }

    Serial.println();

    // KITCHEN
    printTempValue("Kitchen Temp: ", kitchenTemp);
    printTempValue("Kitchen Socket Temp: ", kitchenSocketTemp);

    Serial.print("Kitchen Smoke: ");
    Serial.println(kitchenSmoke);

    Serial.print("Kitchen Gas: ");
    Serial.println(kitchenGas);

    Serial.print("Kitchen State: ");
    Serial.println(stateToText(kitchenState));

    if (kitchenGas >= GAS_WARNING_LEVEL && kitchenGas <
        GAS_FIRE_LEVEL) {
        Serial.println("WARNING: GAS LEAK");
    }

    if (kitchenGas >= GAS_FIRE_LEVEL) {
        Serial.println("FIRE: CRITICAL GAS LEAK");
        Serial.println("KITCHEN POWER DISABLED");
        Serial.println("SIREN ON");
        Serial.println("KITCHEN EMERGENCY LIGHT ON");
    }

    if (kitchenState == WARNING) {
        if (!isnan(kitchenTemp) && kitchenTemp >=
            KITCHEN_TEMP_WARNING && kitchenTemp < KITCHEN_TEMP_FIRE) {
            Serial.println("WARNING: HIGH TEMPERATURE IN KITCHEN");
        }

        if (!isnan(kitchenSocketTemp) && kitchenSocketTemp >=
            KITCHEN_SOCKET_TEMP_WARNING && kitchenSocketTemp <
            KITCHEN_SOCKET_TEMP_FIRE) {
            Serial.println("WARNING: KITCHEN SOCKET OVERHEATING");
        }

        if (kitchenSmoke >= SMOKE_WARNING_LEVEL && kitchenSmoke <
            SMOKE_FIRE_LEVEL) {
            Serial.println("WARNING: SMOKE DETECTED IN KITCHEN");
        }
    }

    if (kitchenState == FIRE) {

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						04
Зм.	Арк.	№ докум.	Підп.	Дата		

```

        if (!isnan(kitchenTemp) && kitchenTemp >=
KITCHEN_TEMP_FIRE) {
            Serial.println("FIRE: HIGH TEMPERATURE IN KITCHEN");
        }

        if (!isnan(kitchenSocketTemp) && kitchenSocketTemp >=
KITCHEN_SOCKET_TEMP_FIRE) {
            Serial.println("FIRE: KITCHEN SOCKET OVERHEATING /
POSSIBLE SHORT CIRCUIT");
        }

        if (kitchenSmoke >= SMOKE_FIRE_LEVEL) {
            Serial.println("FIRE: SMOKE FIRE LEVEL IN KITCHEN");
        }

        Serial.println("KITCHEN POWER DISABLED");
        Serial.println("KITCHEN EMERGENCY LIGHT ON");
    }

    Serial.println();

    // PANEL
    printTempValue("Panel Temp: ", panelTemp);

    Serial.print("Panel State: ");
    Serial.println(stateToText(panelState));

    if (panelState == WARNING) {
        Serial.println("WARNING: HIGH TEMPERATURE IN ELECTRICAL
PANEL");
    }

    if (panelState == FIRE) {
        Serial.println("FIRE: ELECTRICAL PANEL OVERHEATING");
        Serial.println("GLOBAL POWER SHUTDOWN ACTIVE");
        Serial.println("ROOM POWER DISABLED");
        Serial.println("KITCHEN POWER DISABLED");
        Serial.println("ALL MAIN LIGHTS OFF");
        Serial.println("EMERGENCY LIGHTS ON");
        Serial.println("SIREN ON");
    }

    Serial.println();

    // POWER
    if (!mainPowerPresent) {
        Serial.println("POWER WARNING: MAIN POWER LOST");
        Serial.println("SYSTEM CONTINUES WORKING FROM BACKUP
POWER");
        Serial.println("EMERGENCY LIGHTS ON");
    }

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

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						05
Зм.	Арк.	№ докум.	Підп.	Дата		

```

String stateToEmojiText (ZoneState state) {
    switch (state) {
        case NORMAL:
            return "🟢 NORMAL";
        case WARNING:
            return "🟡 WARNING";
        case FIRE:
            return "🔴 FIRE";
        default:
            return "⬜ UNKNOWN";
    }
}

ZoneState getSystemState() {
    ZoneState systemState = NORMAL;
    systemState = maxState(systemState, roomState);
    systemState = maxState(systemState, kitchenState);
    systemState = maxState(systemState, panelState);
    return systemState;
}

void setCurrentAlert (String message, bool sendToTelegram =
true) {
    if (message == lastAlertMessage) {
        return;
    }

    lastAlertMessage = message;

    Blynk.virtualWrite(V60, message);

    if (sendToTelegram) {
        sendTelegramMessage(message);
    }
}

void updateCurrentAlert() {
    if (!mainPowerPresent) {
        setCurrentAlert("🔌 POWER LOST: System works from backup
power");
        return;
    }

    if (panelState == FIRE) {
        if (sirenMuted) {
            setCurrentAlert("🔊 FIRE: Electrical panel overheating.
Siren muted, emergency lights ON");
        } else {
            setCurrentAlert("🔥 FIRE: Electrical panel overheating.
Global shutdown active");
        }
        return;
    }
}

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						06
Зм.	Арк.	№ докум.	Підп.	Дата		

```

    }

    if (kitchenGas >= GAS_FIRE_LEVEL) {
        if (sirenMuted) {
            setCurrentAlert("🔊 FIRE: Critical gas level. Siren
muted, kitchen disabled");
        } else {
            setCurrentAlert("🔥 FIRE: Critical gas level. Kitchen
zone disabled");
        }
        return;
    }

    if (kitchenState == FIRE) {
        if (sirenMuted) {
            setCurrentAlert("🔊 FIRE: Kitchen emergency. Siren
muted, emergency light ON");
        } else {
            setCurrentAlert("🔥 FIRE: Kitchen zone disabled,
emergency light ON");
        }
        return;
    }

    if (roomState == FIRE) {
        if (sirenMuted) {
            setCurrentAlert("🔊 FIRE: Room emergency. Siren muted,
emergency light ON");
        } else {
            setCurrentAlert("🔥 FIRE: Room zone disabled, emergency
light ON");
        }
        return;
    }

    if (kitchenGas >= GAS_WARNING_LEVEL && kitchenGas <
GAS_FIRE_LEVEL) {
        setCurrentAlert("⚠️ WARNING: Gas leak detected in
kitchen");
        return;
    }

    if (panelState == WARNING) {
        setCurrentAlert("⚠️ WARNING: Electrical panel temperature
increased");
        return;
    }

    if (kitchenState == WARNING) {
        setCurrentAlert("⚠️ WARNING: Kitchen zone requires
attention");
        return;
    }

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						07
Зм.	Арк.	№ докум.	Підп.	Дата		

```

    }

    if (roomState == WARNING) {
        setCurrentAlert("⚠ WARNING: Room zone requires
attention");
        return;
    }

    setCurrentAlert("✅ SYSTEM NORMAL: All sensors are within
limits");
}

void sendDataToBlynk() {
    // SENSOR VALUES
    Blynk.virtualWrite(V0, roomTemp);
    Blynk.virtualWrite(V1, kitchenTemp);
    Blynk.virtualWrite(V2, panelTemp);

    Blynk.virtualWrite(V3, roomSocketTemp);
    Blynk.virtualWrite(V4, kitchenSocketTemp);

    Blynk.virtualWrite(V5, roomSmoke);
    Blynk.virtualWrite(V6, kitchenSmoke);
    Blynk.virtualWrite(V7, kitchenGas);

    // ZONE STATES
    Blynk.virtualWrite(V10, stateToEmojiText(roomState));
    Blynk.virtualWrite(V11, stateToEmojiText(kitchenState));
    Blynk.virtualWrite(V12, stateToEmojiText(panelState));

    ZoneState systemState = getSystemState();
    Blynk.virtualWrite(V13, stateToEmojiText(systemState));

    // POWER STATUS
    if (mainPowerPresent) {
        Blynk.virtualWrite(V20, "🟢 PRESENT");
        Blynk.virtualWrite(V21, "⚡ MAIN");
    } else {
        Blynk.virtualWrite(V20, "🔴 LOST");
        Blynk.virtualWrite(V21, "🔋 BACKUP");
    }

    // CURRENT ALERT
    updateCurrentAlert();
}

BLYNK_WRITE(V33) {
    int value = param.asInt();

    if (value == 1) {
        sirenMuted = true;
    }
}

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						08
Зм.	Арк.	№ докум.	Підп.	Дата		

```

String urlEncode(String text) {
    String encoded = "";

    for (int i = 0; i < text.length(); i++) {
        char c = text.charAt(i);

        if (isalnum(c)) {
            encoded += c;
        } else if (c == ' ') {
            encoded += "%20";
        } else if (c == '\\n') {
            encoded += "%0A";
        } else {
            char buf[4];
            sprintf(buf, "%%02X", (unsigned char)c);
            encoded += buf;
        }
    }

    return encoded;
}

void sendTelegramMessage(String message) {
    if (message == lastTelegramMessage) {
        return;
    }

    lastTelegramMessage = message;

    WiFiClientSecure client;
    client.setInsecure();

    HTTPClient https;

    String url = "https://api.telegram.org/bot";
    url += TELEGRAM_BOT_TOKEN;
    url += "/sendMessage?chat_id=";
    url += TELEGRAM_CHAT_ID;
    url += "&text=";
    url += urlEncode(message);

    if (https.begin(client, url)) {
        int httpCode = https.GET();

        Serial.print("Telegram HTTP code: ");
        Serial.println(httpCode);

        https.end();
    } else {
        Serial.println("Telegram connection failed");
    }
}

```

					ДП.АКІТ. 10658921.00.00.000 ПЗ	Арк.
						09
Зм.	Арк.	№ докум.	Підп.	Дата		