A white rectangular frame with black border

Description automatically generated

**TRƯỜNG ĐẠI HỌC BÌNH DƯƠNG**

**KHOA CÔNG NGHỆ THÔNG TIN, ROBOT**

**VÀ TRÍ TUỆ NHÂN TẠO**

** ---oOo---**

**TIỂU LUẬN MÔN HỌC**

**LẬP TRÌNH HỆ THỐNG**

**ĐỀ TÀI:**

**CẢNH BÁO CHÁY SỚM**

**Giáo viên hướng dẫn:** **ThS.Lê Duy Hùng**

**Sinh viên thực hiện : Lê Hải Đông – 22050084**

**Nguyễn Việt Dững – 22050082**

**Bình Dương, 08/2025**

# LỜI CẢM ƠN

Trước hết, nhóm chúng em xin gửi lời cảm ơn chân thành đến Ban Giám hiệu Trường Đại học Bình Dương, Khoa Công nghệ Thông tin, Robot và Trí tuệ nhân tạo đã tạo điều kiện thuận lợi cho chúng em được học tập và nghiên cứu trong suốt thời gian qua.

Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến Thầy **ThS. Lê Duy Hùng**, giảng viên hướng dẫn, đã tận tình chỉ bảo, định hướng và hỗ trợ chúng em trong quá trình thực hiện đề tài. Những ý kiến đóng góp và sự hướng dẫn chi tiết của Thầy đã giúp chúng em hoàn thiện đề tài một cách khoa học và thực tiễn hơn.

Chúng em cũng xin cảm ơn gia đình, bạn bè đã luôn động viên, hỗ trợ về tinh thần và góp ý cho chúng em trong suốt thời gian triển khai tiểu luận.

Mặc dù đã cố gắng hoàn thiện, nhưng do kiến thức và kinh nghiệm thực tế còn hạn chế, báo cáo chắc chắn không tránh khỏi những thiếu sót. Nhóm rất mong nhận được sự góp ý từ Quý Thầy Cô để đề tài ngày càng hoàn thiện hơn. Xin chân thành cảm ơn!

Bình Dương, tháng 8 năm 2025  
Nhóm sinh viên thực hiện

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

**TIỂU LUẬN MÔN: LẬP TRÌNH HỆ THỐNG**

1. **Nhận xét**:

***Những kết quả đạt được:***

***Những hạn chế:***

1. **Điểm đánh giá (Ghi rõ thông tin sinh viên và điểm chữ, điểm số**)

|  |  |
| --- | --- |
|  | *TPHCM, ngày 15 tháng 07 năm 2025*  **Giảng viên chấm thi**  *(Ký và ghi rõ họ tên)*  ………………….. |

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_Toc207126869)

[MỤC LỤC BẢNG VI](#_Toc207126870)

[LỜI MỞ ĐẦU VII](#_Toc207126871)

[CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN 1](#_Toc207126872)

[**1.1.** **Lý do thực hiện đề tài:** 1](#_Toc207126873)

[**1.2.** **Các hệ thống tương tự:** 2](#_Toc207126874)

[a. Hệ thống báo cháy truyền thống 3](#_Toc207126875)

[b. Hệ thống báo cháy chuyên nghiệp cho công trình lớn 3](#_Toc207126876)

[c. Các nghiên cứu, sản phẩm ứng dụng IoT 4](#_Toc207126877)

[d. Nhận xét và tính mới của đề tài 4](#_Toc207126878)

[**1.3.** **Phát biểu bài toán:** 5](#_Toc207126879)

[1.3.1. Mục tiêu: 5](#_Toc207126880)

[1.3.2. Phạm vi: 5](#_Toc207126881)

[1.3.3. Sản phẩm dự kiến: 6](#_Toc207126882)

[1.3.4. Ràng buộc: 6](#_Toc207126883)

[1.3.5. Giả định và phụ thuộc: 6](#_Toc207126884)

[**1.4.** **Kết quả cần đạt:** 7](#_Toc207126885)

[1.4.1. Mục tiêu tổng quát 7](#_Toc207126886)

[1.4.2. Kết quả cụ thể 7](#_Toc207126887)

[1.4.3. Tiêu chí đánh giá 8](#_Toc207126888)

[1.4.4. Tính ứng dụng của đề tài 9](#_Toc207126889)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 10](#_Toc207126890)

[2.1. Cơ sở lý thuyết: 10](#_Toc207126891)

[2.1.1. Khái niệm về cháy và cơ chế phát hiện cháy: 10](#_Toc207126892)

[2.1.2. Nguyên lý hoạt động của các thành phần trong hệ thống: 11](#_Toc207126893)

[2.1.3. Mô hình IoT trong hệ thống cảnh báo cháy 15](#_Toc207126894)

[2.2. Công nghệ sử dụng: 16](#_Toc207126895)

[2.2.1. Vi điều khiển ESP32: 16](#_Toc207126896)

[2.2.2. Cảm biến khí gas MQ-2: 17](#_Toc207126897)

[2.2.3. Cảm biến lửa Flame Sensor 17](#_Toc207126898)

[2.2.4. Còi Buzzer 18](#_Toc207126899)

[2.2.5. Ngôn ngữ lập trình Arduino C/C++ 18](#_Toc207126900)

[2.2.6. Arduino IDE 18](#_Toc207126901)

[2.2.7. Firebase và MQTT 19](#_Toc207126902)

[2.2.8. Nền tảng Blynk (Platform Blynk) 19](#_Toc207126903)

[2.3. Cách tiếp cận, giải quyết vấn đề: 20](#_Toc207126904)

[2.3.1. Mô hình tiếp cận 20](#_Toc207126905)

[2.3.2. Phương pháp phát triển hệ thống: 20](#_Toc207126906)

[2.3.3. Quy trình thực hiện 21](#_Toc207126907)

[CHƯƠNG 3 :PHÂN TÍCH-THIẾT KẾ 23](#_Toc207126908)

[3.1. Các yêu cầu chức năng: 23](#_Toc207126909)

[3.1.1. Ngữ cảnh sử dụng: 23](#_Toc207126910)

[3.2. Các yêu cầu phi chức năng: 24](#_Toc207126911)

[3.3. Mô hình hệ thống: 25](#_Toc207126912)

[3.3.1. Sơ đồ đối tượng /lớp thực thể (entities object/class diagram): 25](#_Toc207126913)

[3.3.2. Kiến trúc phần mềm, sơ đồ thành phần: 26](#_Toc207126914)

[3.3.3. Sơ đồ bố trí, triên khai trên hệ thống phần cứng, mạng: 27](#_Toc207126915)

[3.4. Mô hình xử lý /tương tác: 29](#_Toc207126916)

[3.4.1. Use case chi tiết: 29](#_Toc207126917)

[3.4.2. Sơ đồ tuần tự (sequence diagram): 30](#_Toc207126918)

[3.4.3. Sơ đồ hoạt động (activity diagram): 30](#_Toc207126919)

[3.5. Thiết kế nguyên mẫu giao diện người dùng: 31](#_Toc207126920)

[3.5.1. Hệ thống màn hình: 31](#_Toc207126921)

[3.6. Thiết kế chi tiết: 35](#_Toc207126922)

[3.6.1. Triển khai mô hình tổng quát: 35](#_Toc207126923)

[3.6.2. Thuật toán xử lý: 37](#_Toc207126924)

[CHƯƠNG 4:KẾT QUẢ VÀ THỰC NGHIỆM 38](#_Toc207126925)

[4.1. Các kịch bản thử nghiệm: 38](#_Toc207126926)

[4.2. Kết quả thử nghiệm các kịch bản: 39](#_Toc207126927)

[4.3. Xử lý các trường hợp ngoại lệ: 40](#_Toc207126928)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 41](#_Toc207126929)

[5.1. Kết quả đối chiếu với mục tiêu: 41](#_Toc207126930)

[5.2. Các hạn chế của đồ án: 42](#_Toc207126931)

[5.3. Hướng phát triển: 42](#_Toc207126932)

[TÀI LIỆU THAM KHẢO 44](#_Toc207126933)

**MỤC LỤC ẢNH**

[Hình 1.1. Sơ đồ quy trình nghiệp vụ báo cháy sớm 2](#_Toc207110949)

[Hình 2.1. Cảm biến khí gas MQ-2 11](#_Toc207110950)

[Hình 2.2. Cảm biến lửa 12](#_Toc207110951)

[Hình 2.3. Vi điều khiển ESP32 13](#_Toc207110952)

[Hình 2.4. Còi Buzzer 14](#_Toc207110953)

[Hình 2.5. Sơ đồ khối IoT 3 lớp 15](#_Toc207110954)

[Hình 3.1. Sơ đồ ngữ cảnh hệ thống cảnh báo cháy sớm 23](#_Toc207110955)

[Hình 3.2. Hệ thống cảnh báo cháy sớm 24](#_Toc207110956)

[Hình 3.3. Sơ đồ đối tượng 25](#_Toc207110957)

[Hình 3.4. Sơ đồ lớp (Class diagram) 25](#_Toc207110958)

[Hình 3.5. . Sơ đồ thành phần hệ thống 26](#_Toc207110959)

[Hình 3.6. Sơ đồ bố trí cảm biến 27](#_Toc207110960)

[Hình 3.7. Sơ đồ triển khai phần cứng 27](#_Toc207110961)

[Hình 3.8. Sơ đồ triển khai mạng hệ thống 28](#_Toc207110962)

[Hình 3.9. Sơ đồ tuần tự 30](#_Toc207110963)

[Hình 3.10. Sơ đồ hoạt động 30](#_Toc207110964)

[Hình 3.11. Giao diện chính của ứng dụng cảnh báo cháy sớm 31](#_Toc207110965)

[Hình 3.12. Điều khiển còi 32](#_Toc207110966)

[Hình 3.13. Đổi màu khi phát hiện có lửa 32](#_Toc207110967)

[Hình 3.14. Gửi cảnh báo về gmail khi phát hiện nồng độ khí ga cao bất ngờ 33](#_Toc207110968)

[Hình 3.15. gửi cảnh báo về gmail khi phát hiện có lửa 34](#_Toc207110969)

[Hình 3.16. Sơ đồ khối phần cứng 35](#_Toc207110970)

[Hình 3.17. Sơ đồ nối dây giữa ESP32 và các cảm biến trong hệ thống 36](#_Toc207110971)

[Hình 3.18. Lưu đồ thuật toán: 37](#_Toc207110972)

# MỤC LỤC BẢNG

[Bảng 1.1. Tiêu chí đánh giá 8](#_Toc207112212)

[Bảng 2.1. So sánh ESP32 và Arduino Uno 16](#_Toc207112213)

[Bảng 2.2. So sánh MQ-2 và MQ-7 17](#_Toc207112214)

[Bảng 2.3. So sánh Buzzer và LED 18](#_Toc207112215)

[Bảng 2.4. Firebase và MQTT 19](#_Toc207112216)

[Bảng 3.1. Đặc tả chi tiết các use case chính 29](#_Toc207112217)

[Bảng 3.2. Bảng kết nối ESP32 36](#_Toc207112218)

[Bảng 4.1. Kết quả thử nghiệm các kịch bản 39](#_Toc207112219)

[Bảng 5.1. Kết quả đối chiếu với mục tiêu 41](#_Toc207112220)

# LỜI MỞ ĐẦU

Trong bối cảnh xã hội hiện đại, vấn đề phòng cháy chữa cháy (PCCC) ngày càng được quan tâm đặc biệt do số lượng các vụ hỏa hoạn gia tăng, gây thiệt hại nghiêm trọng về người và tài sản. Các hệ thống báo cháy truyền thống thường chỉ phát tín hiệu cảnh báo tại chỗ, trong khi các hệ thống chuyên nghiệp lại có chi phí quá cao, khó áp dụng đại trà cho hộ gia đình và cơ sở nhỏ.

Với sự phát triển mạnh mẽ của công nghệ IoT (Internet of Things), việc xây dựng các hệ thống cảnh báo cháy thông minh, chi phí thấp, dễ dàng triển khai đã trở thành hướng đi tiềm năng. Xuất phát từ nhu cầu thực tế đó, nhóm sinh viên lựa chọn thực hiện đề tài:

“**Cảnh báo cháy sớm**”

Mục tiêu của đề tài là thiết kế và triển khai một mô hình cảnh báo cháy thông minh sử dụng vi điều khiển ESP32 kết hợp với các cảm biến khói, khí gas (MQ-2), cảm biến lửa, còi báo động và kết nối Internet để gửi cảnh báo từ xa. Hệ thống hướng đến các tiêu chí: phát hiện nhanh – cảnh báo đa dạng – chi phí thấp – dễ dàng áp dụng trong thực tế.

# CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

* 1. **Lý do thực hiện đề tài:**

Trong xã hội hiện đại, vấn đề an toàn phòng cháy chữa cháy (PCCC) ngày càng trở nên quan trọng và cấp thiết. Hỏa hoạn không chỉ là thảm họa gây thiệt hại to lớn về tài sản mà còn trực tiếp đe dọa tính mạng con người. Ở Việt Nam cũng như trên thế giới, những vụ cháy lớn xảy ra liên tục trong những năm gần đây đã gióng lên hồi chuông cảnh báo về sự thiếu hụt các giải pháp cảnh báo cháy sớm, hiệu quả và chi phí phù hợp.

Theo số liệu của **Cục Cảnh sát PCCC và CNCH – Bộ Công an Việt Nam**, chỉ tính riêng năm 2023, cả nước đã xảy ra hơn **3.200 vụ cháy**, làm chết **152 người**, bị thương **201 người**, gây thiệt hại tài sản hàng ngàn tỷ đồng. Một số vụ điển hình như:

* Vụ cháy **chung cư mini ở Hà Nội** (9/2023) khiến hơn 50 người tử vong, hàng chục người bị thương.
* Vụ cháy **nhà xưởng tại Bình Dương** (2022) gây thiệt hại hàng trăm tỷ đồng.
* Vụ cháy **quán karaoke ở Bình Dương** (2022) làm 32 người thiệt mạng.

Những con số này cho thấy hỏa hoạn vẫn luôn là mối nguy cơ thường trực trong đời sống hàng ngày.S

Hiện nay, trong công tác PCCC, các nghiệp vụ cơ bản thường bao gồm:

1. **Phòng ngừa cháy**: trang bị bình chữa cháy, tập huấn, xây dựng lối thoát hiểm.
2. **Phát hiện cháy**: dựa vào hệ thống báo cháy hoặc do con người phát hiện.
3. **Cảnh báo và báo động**: kích hoạt còi, chuông, đèn để mọi người sơ tán.
4. **Xử lý sự cố**: sử dụng bình chữa cháy, hệ thống phun nước, gọi lực lượng PCCC.

Tuy nhiên, điểm yếu lớn nhất hiện nay nằm ở khâu phát hiện và cảnh báo sớm. Phần lớn hộ gia đình và cơ sở nhỏ lẻ chỉ dựa vào khả năng phát hiện trực tiếp bằng mắt thường hoặc khứu giác, dẫn đến việc phát hiện cháy quá muộn, khi ngọn lửa đã lan rộng.

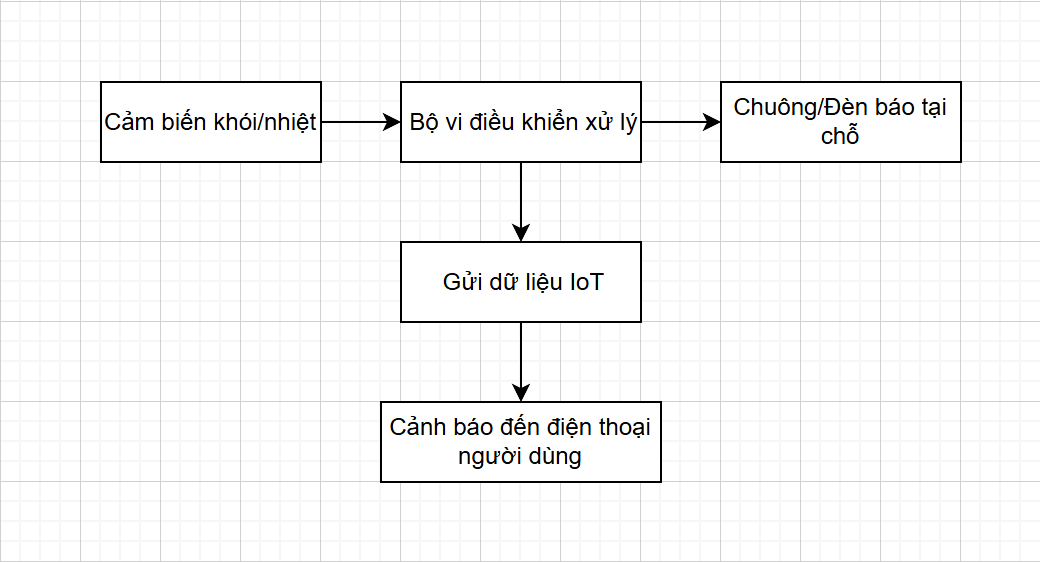
Ngoài ra, các hệ thống báo cháy hiện có cũng còn nhiều hạn chế:

* **Hệ thống truyền thống**: chỉ có chuông báo tại chỗ, người đi vắng sẽ không biết.
* **Hệ thống hiện đại trong tòa nhà lớn**: chi phí lắp đặt và bảo trì cao, khó áp dụng đại trà.
* **Giải pháp tự chế**: nhiều nghiên cứu sinh viên còn ở mức thử nghiệm, thiếu tính ổn định khi đưa vào thực tế.

Chính vì vậy, đề tài “Cảnh báo cháy sớm” được thực hiện với mong muốn xây dựng một hệ thống báo cháy tích hợp ưu điểm của các mô hình hiện tại:

* **Phát hiện nhanh** nhờ cảm biến khói, nhiệt độ.
* **Cảnh báo đa dạng**: chuông, đèn và đặc biệt là cảnh báo từ xa qua Internet.
* **Chi phí thấp**, phù hợp cho hộ gia đình, văn phòng, kho hàng nhỏ.
* **Dễ mở rộng**: có thể tích hợp thêm camera, cảm biến khí độc, hệ thống chữa cháy tự động trong tương lai.

**Sơ đồ quy trình nghiệp vụ báo cháy sớm (mô tả):**



**Hình 1.1. Sơ đồ quy trình nghiệp vụ báo cháy sớm**

Như vậy, lý do thực hiện đề tài xuất phát từ:

* Thực trạng hỏa hoạn gia tăng, gây hậu quả nghiêm trọng.
* Sự thiếu hụt giải pháp báo cháy sớm ở hộ gia đình và doanh nghiệp nhỏ.
* Nhu cầu thực tiễn về một hệ thống đơn giản, chi phí thấp nhưng hiệu quả cao.
  1. **Các hệ thống tương tự:**

Để xây dựng một hệ thống cảnh báo cháy sớm hiệu quả, cần nghiên cứu và tham khảo các công trình, sản phẩm, dự án đã và đang được ứng dụng trong thực tế.

### Hệ thống báo cháy truyền thống

Hệ thống báo cháy truyền thống thường được triển khai ở hộ gia đình, văn phòng nhỏ, nhà xưởng quy mô vừa. Mô hình cơ bản gồm:

* **Cảm biến khói/nhiệt**: phát hiện sự gia tăng nhiệt độ hoặc có khói trong không khí.
* **Thiết bị báo động**: chuông, còi, đèn báo.

**Nguyên lý hoạt động:** khi cảm biến phát hiện dấu hiệu cháy, tín hiệu được truyền trực tiếp đến chuông báo động. Người trong khu vực nghe thấy chuông sẽ sơ tán hoặc dùng bình chữa cháy để xử lý sự cố.

**Ưu điểm:**

* Cấu tạo đơn giản, giá thành rẻ.
* Dễ dàng lắp đặt và sử dụng.
* Phù hợp cho hộ gia đình và văn phòng nhỏ.

**Nhược điểm:**

* Chỉ báo động tại chỗ, không gửi được cảnh báo từ xa.
* Phụ thuộc vào việc có người nghe thấy tín hiệu hay không.
* Dễ gây báo động giả khi có khói từ bếp nấu ăn.

### Hệ thống báo cháy chuyên nghiệp cho công trình lớn

Đây là hệ thống được triển khai bắt buộc ở các công trình lớn như khách sạn, trung tâm thương mại, bệnh viện, trường học, nhà máy. Cấu trúc hệ thống thường gồm:

* Hàng loạt cảm biến (khói, nhiệt, khí CO).
* **Trung tâm điều khiển** (fire alarm control panel).
* **Thiết bị ngoại vi**: chuông, còi, đèn, loa phát thanh.
* Hệ thống kết nối với lực lượng PCCC chuyên nghiệp.

Khi phát hiện cháy, hệ thống sẽ đồng thời kích hoạt chuông/đèn báo, hiển thị khu vực có sự cố trên bảng điều khiển, đồng thời có thể khởi động hệ thống chữa cháy tự động (sprinkler).

**Ưu điểm:**

* Độ chính xác và tính an toàn cao.
* Khả năng giám sát diện tích lớn.
* Có thể tự động kích hoạt hệ thống chữa cháy.

**Nhược điểm:**

* **Chi phí rất cao**, từ vài trăm triệu đến hàng tỷ đồng.
* Đòi hỏi đội ngũ kỹ thuật vận hành và bảo trì thường xuyên.
* Không phù hợp để áp dụng cho hộ gia đình hoặc cơ sở nhỏ.

### Các nghiên cứu, sản phẩm ứng dụng IoT

Trong những năm gần đây, với sự phát triển mạnh mẽ của công nghệ **Internet of Things (IoT)**, nhiều công trình nghiên cứu và sản phẩm báo cháy thông minh đã được đề xuất và thử nghiệm.

**Ví dụ trong nước:**

* Một số đề tài của sinh viên các trường đại học tại Việt Nam đã ứng dụng **Arduino** hoặc **ESP8266/ESP32** kết hợp cảm biến khói, cảm biến nhiệt để phát hiện cháy, sau đó gửi cảnh báo qua ứng dụng điện thoại (Zalo, Telegram, Email).
* Có đề tài tích hợp thêm camera giám sát để chụp ảnh hiện trường và gửi cho người dùng khi có cháy.

**Ví dụ quốc tế:**

* Công trình nghiên cứu “IoT-Based Fire Detection System” (Ấn Độ, 2021) đã chứng minh tính khả thi của việc kết hợp cảm biến nhiệt, khói với module ESP8266 để cảnh báo qua mạng không dây.
* Một số startup tại Mỹ và Hàn Quốc đã phát triển thiết bị báo cháy thông minh có thể kết nối với Smart Home System.

**Ưu điểm:**

* Giá thành rẻ hơn nhiều so với hệ thống chuyên nghiệp.
* Có khả năng cảnh báo từ xa qua Internet.
* Tích hợp với các thiết bị thông minh khác (camera, smartphone).

**Nhược điểm:**

* Độ ổn định và độ bền chưa cao.
* Phụ thuộc vào kết nối Internet.
* Một số sản phẩm mới dừng lại ở mức thử nghiệm, chưa phổ biến trên thị trường.

### Nhận xét và tính mới của đề tài

Qua phân tích, có thể nhận thấy:

* **Hệ thống truyền thống**: đơn giản, rẻ nhưng hạn chế về phạm vi cảnh báo.
* **Hệ thống chuyên nghiệp**: an toàn cao nhưng chi phí lớn, khó ứng dụng đại trà.
* **Hệ thống IoT thử nghiệm**: có triển vọng nhưng chưa ổn định.

Tính mới của đề tài “Cảnh báo cháy sớm” chính là:

* Đề xuất giải pháp trung hòa giữa chi phí thấp và tính năng thông minh.
* Tích hợp cảnh báo đa phương thức: chuông, đèn, và cảnh báo từ xa.
* Thiết kế linh hoạt – mở rộng dễ dàng (bổ sung cảm biến khí độc, camera).
* Phù hợp cho hộ gia đình, văn phòng, kho nhỏ – những nơi hiện nay còn thiếu hệ thống báo cháy.
  1. **Phát biểu bài toán:**

Để thực hiện đề tài *“Cảnh báo cháy sớm”*, nhóm nghiên cứu cần xác định rõ mục tiêu, phạm vi, sản phẩm, ràng buộc và giả định của hệ thống. Đây là cơ sở để định hướng toàn bộ quá trình phân tích – thiết kế – triển khai mô hình.

### 1.3.1. Mục tiêu:

* Xây dựng một hệ thống cảnh báo cháy sớm có khả năng phát hiện kịp thời các dấu hiệu hỏa hoạn ngay từ giai đoạn đầu (khói, nhiệt độ bất thường, khí độc).
* Thiết kế hệ thống kết hợp cảnh báo tại chỗ (chuông, đèn, còi) và cảnh báo từ xa (qua Internet đến điện thoại thông minh).
* Hệ thống phải đơn giản – chi phí thấp – dễ triển khai để bất kỳ hộ gia đình, văn phòng nhỏ hay kho chứa nào cũng có thể ứng dụng.
* Đảm bảo tính ổn định, độ chính xác cao, hạn chế tối đa báo động giả.

### 1.3.2. Phạm vi:

* Phạm vi ứng dụng:
  + Hộ gia đình, ký túc xá, nhà trọ, văn phòng nhỏ.
  + Nhà kho hoặc xưởng sản xuất quy mô nhỏ, diện tích dưới 200 m².
* Không thuộc phạm vi:
  + Công trình lớn như trung tâm thương mại, cao ốc văn phòng, bệnh viện (yêu cầu hệ thống PCCC chuyên nghiệp).
* Phạm vi công nghệ:
  + Sử dụng vi điều khiển (ESP32, Arduino) kết hợp cảm biến khói/nhiệt.
  + Sử dụng module WiFi để gửi cảnh báo qua Internet.
  + Xây dựng ứng dụng/giải pháp cảnh báo trên điện thoại.

### 1.3.3. Sản phẩm dự kiến:

* Phần cứng:
  + Bộ cảm biến khói (MQ-2, MQ-135) và cảm biến nhiệt (DHT11, DS18B20).
  + Bộ vi điều khiển (Arduino/ESP32) làm trung tâm xử lý.
  + Chuông, còi, đèn LED cảnh báo.
  + Module kết nối WiFi (ESP8266/ESP32 tích hợp).
* Phần mềm:
  + Chương trình điều khiển vi điều khiển: xử lý tín hiệu cảm biến, phát hiện sự cố.
  + Hệ thống cảnh báo: điều khiển chuông/đèn tại chỗ.
  + Ứng dụng gửi cảnh báo qua Internet: tin nhắn điện thoại, thông báo trên app.
* Tài liệu đi kèm:
  + Hướng dẫn cài đặt và sử dụng.
  + Tài liệu phân tích – thiết kế và báo cáo thực nghiệm.

### 1.3.4. Ràng buộc:

* Hệ thống phụ thuộc vào kết nối Internet ổn định để gửi cảnh báo từ xa.
* Độ chính xác của cảm biến có thể bị ảnh hưởng bởi môi trường (bụi, khói bếp, hơi nước).
* Hệ thống phù hợp cho không gian nhỏ và trung bình, chưa tối ưu cho không gian lớn.
* Cần nguồn điện liên tục để đảm bảo hệ thống hoạt động 24/7.

### 1.3.5. Giả định và phụ thuộc:

* Người dùng có điện thoại thông minh và cài ứng dụng nhận cảnh báo.
* Mạng WiFi hoạt động ổn định, không bị ngắt kết nối quá lâu.
* Người dùng có kiến thức cơ bản để lắp đặt và vận hành hệ thống.
* Thiết bị được triển khai trong điều kiện bình thường, không chịu tác động khắc nghiệt (mưa, nắng trực tiếp, độ ẩm quá cao).
  1. **Kết quả cần đạt:**

Để đánh giá mức độ hoàn thành và tính hiệu quả của đề tài *“Cảnh báo cháy sớm”*, cần xác định rõ những kết quả mong muốn đạt được sau khi triển khai. Các kết quả này không chỉ bao gồm sản phẩm cuối cùng, mà còn liên quan đến tiêu chí đánh giá, mức độ áp dụng thực tế và khả năng mở rộng.

### 1.4.1. Mục tiêu tổng quát

* Xây dựng được một mô hình hệ thống cảnh báo cháy sớm có khả năng phát hiện kịp thời và đưa ra cảnh báo đa dạng (chuông, đèn, tin nhắn điện thoại).
* Hệ thống hoạt động ổn định, chính xác, hạn chế báo động giả.
* Đảm bảo chi phí thấp, dễ sử dụng, dễ triển khai trong thực tế.
* Có khả năng mở rộng, tích hợp thêm các cảm biến và thiết bị thông minh trong tương lai.

### 1.4.2. Kết quả cụ thể

1. **Mô hình phần cứng**: Hoàn thiện hệ thống gồm cảm biến khói, cảm biến nhiệt, vi điều khiển, chuông, đèn báo, module WiFi.
2. **Phần mềm điều khiển**: Xây dựng chương trình điều khiển vi điều khiển, xử lý dữ liệu cảm biến, phát hiện dấu hiệu cháy, đưa ra cảnh báo.
3. **Chức năng cảnh báo đa phương thức**:
   * Cảnh báo tại chỗ: chuông, còi, đèn LED.
   * Cảnh báo từ xa: gửi thông báo qua Internet đến điện thoại.
4. **Tài liệu hướng dẫn**: Cung cấp tài liệu chi tiết về cài đặt, sử dụng và bảo trì hệ thống.
5. **Khả năng mở rộng**: Thiết kế theo hướng mở, dễ dàng bổ sung cảm biến khác (khí độc CO, camera giám sát, hệ thống chữa cháy tự động).

### 1.4.3. Tiêu chí đánh giá

Để đảm bảo kết quả đạt yêu cầu, hệ thống sẽ được đánh giá dựa trên các tiêu chí sau:

**Bảng 1.1. Tiêu chí đánh giá**

| **Kết quả cần đạt** | **Tiêu chí đánh giá cụ thể** | **Ý nghĩa/Ứng dụng thực tế** |
| --- | --- | --- |
| Hoàn thiện mô hình hệ thống | Hệ thống phần cứng và phần mềm được triển khai đầy đủ, hoạt động liên tục trong 7 ngày thử nghiệm | Có thể triển khai thực tế tại hộ gia đình, văn phòng nhỏ |
| Phát hiện cháy kịp thời | Thời gian phát hiện và kích hoạt cảnh báo < 5 giây sau khi cảm biến nhận tín hiệu | Người trong khu vực có thêm thời gian để sơ tán, giảm thiểu thương vong |
| Cảnh báo tại chỗ | Chuông kêu, đèn nhấp nháy ngay khi phát hiện cháy | Cảnh báo trực tiếp cho mọi người có mặt tại hiện trường |
| Cảnh báo từ xa | Thông báo được gửi đến điện thoại trong vòng 5 giây | Chủ nhà, quản lý vẫn biết sự cố ngay cả khi đi vắng |
| Tính ổn định và độ chính xác | Hệ thống hoạt động liên tục, sai số cảm biến < 10%, hạn chế báo động giả | Đảm bảo hệ thống đáng tin cậy trong thực tế |
| Giao diện đơn giản | Người dùng không cần kiến thức kỹ thuật vẫn lắp đặt và vận hành được | Dễ dàng triển khai cho hộ gia đình, người cao tuổi |
| Chi phí thấp | Tổng chi phí linh kiện và triển khai < 1,5 triệu đồng | Phù hợp với khả năng chi trả của nhiều đối tượng |
| Khả năng mở rộng | Có thể bổ sung thêm cảm biến, kết nối với ứng dụng Smart Home | Hệ thống có tiềm năng phát triển trong tương lai |

### 1.4.4. Tính ứng dụng của đề tài

* **Trong hộ gia đình**: Hệ thống giúp phát hiện sớm sự cố cháy, đặc biệt khi gia chủ đi vắng, giảm thiểu nguy cơ thiệt hại lớn.
* **Trong nhà trọ, ký túc xá**: Giúp quản lý và người ở trọ được cảnh báo kịp thời, hạn chế nguy cơ cháy lan.
* **Trong kho hàng, xưởng nhỏ**: Hỗ trợ bảo vệ tài sản có giá trị lớn, cảnh báo sớm để xử lý kịp thời.
* **Trong tương lai**: Có thể tích hợp vào hệ thống **nhà thông minh (Smart Home)**, kết nối với hệ thống camera giám sát và hệ thống chữa cháy tự động.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Cơ sở lý thuyết:

### 2.1.1. Khái niệm về cháy và cơ chế phát hiện cháy:

Hỏa hoạn luôn là một trong những rủi ro tiềm ẩn, có thể xảy ra bất cứ lúc nào và gây ra hậu quả nghiêm trọng về người và tài sản. Theo số liệu thống kê của Cục Phòng cháy chữa cháy, mỗi năm cả nước xảy ra hàng nghìn vụ cháy, trong đó nhiều vụ có nguyên nhân từ rò rỉ khí gas, chập điện, bất cẩn khi sử dụng lửa. Đặc điểm chung của các vụ cháy này là thường phát hiện muộn, khi ngọn lửa đã lan rộng và gây khó khăn cho việc xử lý.

Trong thực tế, con người thường phát hiện cháy dựa vào khói, mùi khí gas hoặc ánh lửa. Tuy nhiên, việc phát hiện thủ công tiềm ẩn nhiều hạn chế:

* Người trực tiếp giám sát có thể không có mặt tại chỗ.
* Khói và khí gas thường lan tỏa chậm, khó nhận biết ở giai đoạn đầu.
* Khi phát hiện, đám cháy đã bùng phát lớn, gây nguy hiểm.

Để giải quyết vấn đề này, các hệ thống báo cháy tự động đã ra đời. Hệ thống truyền thống thường chỉ gồm cảm biến khói/nhiệt kết nối với còi báo động. Dù đơn giản, nhưng loại này chỉ cảnh báo tại chỗ và dễ bị nhiễu bởi yếu tố môi trường như khói bếp, hơi nước, bụi.

Với sự phát triển của Internet of Things (IoT), hệ thống cảnh báo cháy đã được nâng cấp lên mức thông minh hơn. IoT mang lại những đặc điểm nổi bật:

* Kết nối từ xa: người dùng có thể giám sát trên điện thoại thông minh dù ở bất kỳ đâu.
* Phân tích dữ liệu: hệ thống không chỉ cảnh báo tức thời mà còn lưu trữ dữ liệu để theo dõi và dự đoán nguy cơ.
* Tích hợp đa cảm biến: có thể kết hợp nhiều loại cảm biến (khí gas, khói, nhiệt độ, lửa) để giảm báo giả và tăng độ chính xác.
* Tự động hóa: khi phát hiện cháy, ngoài việc kích hoạt còi báo, hệ thống còn có thể điều khiển quạt, mở van nước, kích hoạt hệ thống chữa cháy.

Từ những lợi ích này, việc xây dựng một hệ thống cảnh báo cháy sớm dựa trên ESP32 kết hợp các cảm biến là hướng đi phù hợp, vừa có chi phí hợp lý, vừa mang lại hiệu quả cao.

### 2.1.2. Nguyên lý hoạt động của các thành phần trong hệ thống:

Để hệ thống hoạt động chính xác, cần hiểu rõ cơ chế hoạt động của từng thành phần chính.

**a) Cảm biến khí gas MQ-2:**



**Hình 2.1. Cảm biến khí gas MQ-2**

MQ-2 là cảm biến bán dẫn dựa trên vật liệu SnO₂ (thiếc dioxide). Ở điều kiện bình thường, khi không có khói hay khí gas, điện trở của SnO₂ khá cao, khiến dòng điện đi qua cảm biến yếu. Khi trong môi trường xuất hiện các loại khí dễ cháy như LPG, methane (CH₄), carbon monoxide (CO), các phân tử khí này tác động lên bề mặt, làm giảm điện trở. Kết quả là dòng điện qua cảm biến thay đổi, tạo ra sự chênh lệch điện áp.

ESP32 sẽ đọc giá trị điện áp này thông qua chân ADC (analogRead() trong Arduino C/C++). Giá trị càng cao thì nồng độ khí càng lớn.

**Ví dụ:**

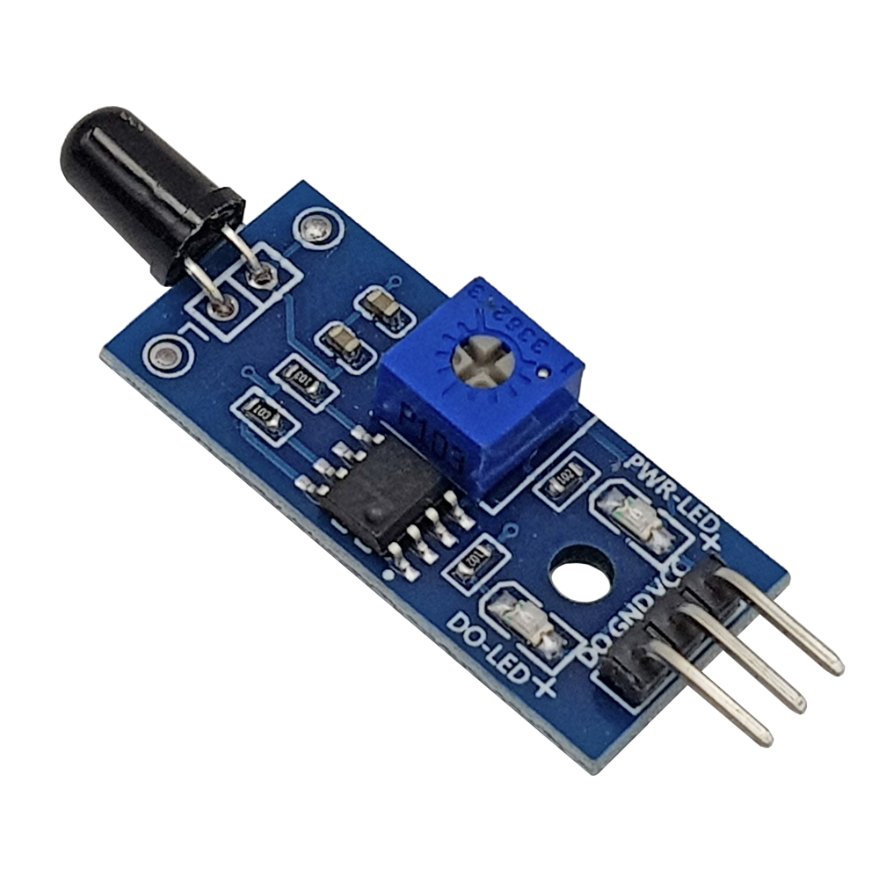
* Khi MQ-2 đọc giá trị ~100–200 → môi trường bình thường.
* Khi MQ-2 đọc >400 → có thể có khí gas rò rỉ.
* Khi MQ-2 đọc >800 → nguy cơ cháy cao, cần cảnh báo ngay.

**Ứng dụng MQ-2:**

* Phát hiện rò rỉ khí gas trong bếp gas, bình gas mini.
* Cảnh báo khói trong nhà trọ, ký túc xá.
* Giám sát môi trường trong nhà xưởng có nhiều hóa chất dễ cháy.

Hạn chế: MQ-2 dễ bị ảnh hưởng bởi khói thuốc lá, hơi cồn, khói bếp, có thể gây báo giả. Vì vậy, thường kết hợp với Flame Sensor để tăng độ chính xác.

**b) Cảm biến lửa (Flame Sensor):**



**Hình 2.2. Cảm biến lửa**

Flame Sensor hoạt động dựa trên nguyên lý thu nhận tia hồng ngoại (IR) phát ra từ ngọn lửa. Trong dải bước sóng 760–1100 nm, cảm biến có độ nhạy cao, có thể phản ứng trong vài mili giây.

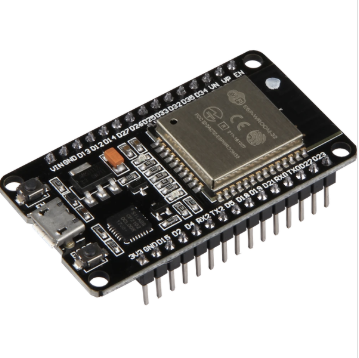
Khi có ngọn lửa, diode quang trong cảm biến sẽ nhận tia IR và chuyển thành tín hiệu điện áp. ESP32 đọc tín hiệu này thông qua digitalRead(). Nếu cảm biến trả về mức HIGH (hoặc LOW tùy thiết kế), nghĩa là đã phát hiện lửa.

Ứng dụng Flame Sensor:

* Phát hiện ngọn lửa trực tiếp trong phòng kín.
* Bổ sung cho MQ-2 để phân biệt giữa “khói bếp” và “cháy thật”.
* Sử dụng trong xưởng sản xuất, kho chứa hàng.

Hạn chế: Flame Sensor có khoảng cách phát hiện giới hạn (20–100 cm) và dễ bị nhiễu trong môi trường ánh sáng mạnh.

**c) Vi điều khiển ESP32:**



**Hình 2.3. Vi điều khiển ESP32**

ESP32 là thành phần quan trọng nhất – “bộ não” của hệ thống. Nó có CPU 32-bit, tốc độ 240 MHz, bộ nhớ lớn hơn Arduino Uno, và đặc biệt là tích hợp WiFi/Bluetooth.

Trong chương trình, ESP32 đóng vai trò:

* Đọc dữ liệu cảm biến: sử dụng analogRead() để lấy giá trị MQ-2, digitalRead() cho Flame Sensor.
* Xử lý dữ liệu: so sánh giá trị với ngưỡng định sẵn (if…else).
* Điều khiển đầu ra: kích hoạt Buzzer bằng digitalWrite().
* Kết nối IoT: gửi dữ liệu qua WiFi đến Firebase hoặc MQTT.
* **Ví dụ logic xử lý đơn giản trong Arduino C/C++:**

int mq2Value = analogRead(pinMQ2);

int flameValue = digitalRead(pinFlame);

if (mq2Value > 400 || flameValue == HIGH) {

digitalWrite(pinBuzzer, HIGH); // bật còi cảnh báo

}

else {

digitalWrite(pinBuzzer, LOW); // tắt còi

}

ESP32 giúp hệ thống trở nên thông minh hơn, thay vì chỉ bật còi khi có tín hiệu, nó có thể xử lý đa điều kiện, kết hợp cảm biến và gửi cảnh báo từ xa.

**d) Còi Buzzer**



**Hình 2.4. Còi Buzzer**

Buzzer là thiết bị phát âm thanh cảnh báo. Nguyên lý hoạt động dựa trên dao động cơ học của màng kim loại khi có dòng điện đi qua. Khi ESP32 xuất tín hiệu mức HIGH ra chân kết nối, Buzzer phát âm thanh với cường độ từ 70–90 dB, đủ để cảnh báo trong phạm vi phòng hoặc nhà xưởng nhỏ.

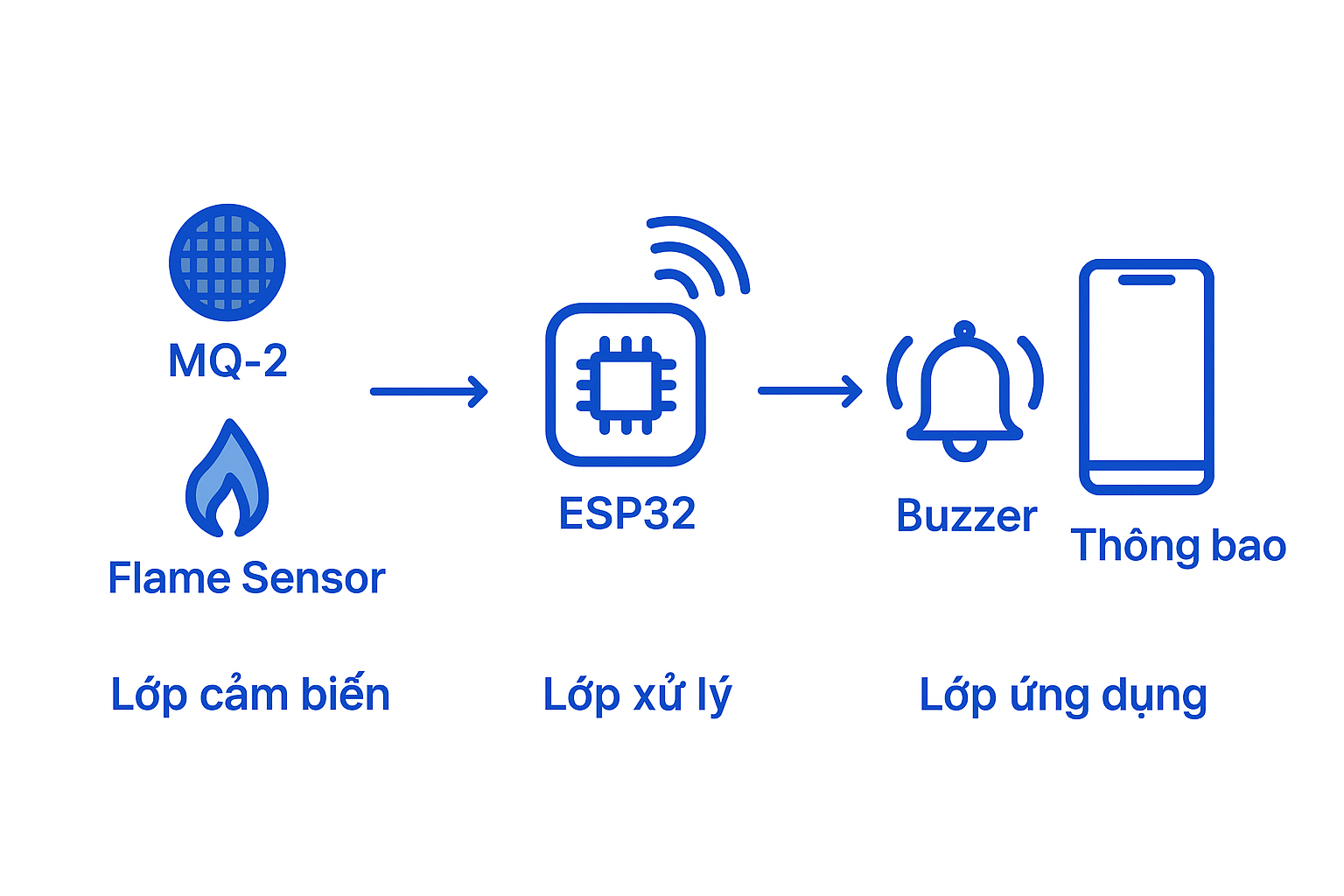
Trong Arduino C/C++, điều khiển Buzzer thường rất đơn giản:

* digitalWrite(pinBuzzer, HIGH) → bật còi.
* digitalWrite(pinBuzzer, LOW) → tắt còi.
* tone(pinBuzzer, 1000) → phát âm thanh tần số 1 kHz.

Ứng dụng:

* Cảnh báo cháy tại chỗ.
* Kết hợp với đèn LED để tăng hiệu quả cảnh báo.

### 2.1.3. Mô hình IoT trong hệ thống cảnh báo cháy



**Hình 2.5. Sơ đồ khối IoT 3 lớp**

**Mô tả:**

* **Lớp cảm biến**: MQ-2 thu thập nồng độ khí gas/khói, Flame Sensor phát hiện ngọn lửa. Đây là “giác quan” của hệ thống.
* **Lớp xử lý**: ESP32 nhận tín hiệu từ cảm biến, phân tích dữ liệu, so sánh ngưỡng và quyết định hành động. Ngoài ra, ESP32 có thể kết nối WiFi để gửi dữ liệu lên Internet (Firebase/MQTT).
* **Lớp ứng dụng**: Buzzer phát âm thanh cảnh báo tại chỗ, đồng thời người dùng có thể nhận thông báo trên smartphone.

**Hệ thống được xây dựng theo mô hình IoT ba lớp:**

1. **Lớp cảm biến**: MQ-2 và Flame Sensor liên tục đọc dữ liệu từ môi trường.
2. **Lớp xử lý**: ESP32 phân tích dữ liệu, xử lý ngưỡng, kết hợp đa cảm biến để giảm báo giả.
3. **Lớp ứng dụng**: cảnh báo bằng Buzzer và gửi dữ liệu đến điện thoại qua Internet.

**So với hệ thống truyền thống, mô hình IoT này có nhiều ưu điểm:**

* Cảnh báo **tại chỗ và từ xa**.
* Dữ liệu có thể được lưu trữ, phân tích.
* Dễ dàng mở rộng bằng cách thêm cảm biến hoặc thiết bị IoT khác.

## 2.2. Công nghệ sử dụng:

Trong quá trình xây dựng hệ thống báo cháy sớm, nhóm cần lựa chọn công nghệ phù hợp để vừa đảm bảo chi phí thấp, vừa có tính ổn định và khả năng mở rộng. Các thành phần được lựa chọn bao gồm: ESP32, cảm biến MQ-2, Flame Sensor, Buzzer, ngôn ngữ Arduino C/C++, Arduino IDE, Firebase và MQTT.

### 2.2.1. Vi điều khiển ESP32:

ESP32 được lựa chọn thay vì các vi điều khiển phổ biến khác (Arduino Uno, NodeMCU ESP8266) vì:

* **Ưu điểm**:
  + CPU 32-bit, tốc độ 240 MHz, xử lý nhanh hơn nhiều so với Arduino Uno (8-bit, 16 MHz).
  + Tích hợp WiFi và Bluetooth, không cần module rời như Arduino Uno.
  + Bộ nhớ lớn hơn, thích hợp cho lập trình IoT.
  + Giá thành rẻ (chỉ khoảng 150.000–200.000 VNĐ).
* **Nhược điểm**:
  + Lập trình phức tạp hơn so với Arduino Uno.
  + Tiêu thụ năng lượng cao hơn.

**So sánh ESP32 và Arduino Uno:**

**Bảng 2.1. So sánh ESP32 và Arduino Uno**

| **Đặc điểm** | **ESP32** | **Arduino Uno** |
| --- | --- | --- |
| CPU | 32-bit, 240 MHz | 8-bit, 16 MHz |
| Kết nối không dây | WiFi + Bluetooth tích hợp | Không có |
| Bộ nhớ | Flash 4 MB, RAM 520 KB | Flash 32 KB, RAM 2 KB |
| Giá | Rẻ (~150k – 200k) | Khoảng 200k – 250k |
| Ứng dụng | IoT, xử lý dữ liệu, AI nhỏ | Học tập, mô phỏng đơn giản |

### 2.2.2. Cảm biến khí gas MQ-2:

MQ-2 được sử dụng thay vì MQ-7, MQ-135 vì:

* **Ưu điểm**:
  + Phát hiện nhiều loại khí dễ cháy (LPG, CH₄, CO), đa dụng hơn MQ-7 (chỉ nhạy với CO).
  + Giá rẻ (40.000–60.000 VNĐ).
  + Dễ mua, tài liệu phong phú.
* **Nhược điểm**:
  + Dễ gây báo giả khi có khói bếp, hơi cồn.
  + Cần thời gian làm nóng trước khi hoạt động ổn định.

**So sánh MQ-2 và MQ-7**:

**Bảng 2.2. So sánh MQ-2 và MQ-7**

| **Đặc điểm** | **MQ-2** | **MQ-7** |
| --- | --- | --- |
| Loại khí phát hiện | LPG, CH₄, CO, khói | CO chủ yếu |
| Ứng dụng | Báo cháy, rò rỉ gas đa dụng | Báo độc hại CO chuyên dụng |
| Giá thành | Rẻ hơn | Cao hơn |

### 2.2.3. Cảm biến lửa Flame Sensor

Flame Sensor được lựa chọn vì:

* **Ưu điểm**:
  + Giá thành rẻ (~20.000–30.000 VNĐ).
  + Phản ứng nhanh với ngọn lửa.
  + Dễ tích hợp với ESP32.
* **Nhược điểm**:
  + Khoảng cách phát hiện ngắn (20–100 cm).
  + Bị nhiễu ánh sáng mạnh.

So với camera hồng ngoại:

* Camera đắt hơn (vài trăm nghìn đến vài triệu).
* Cần xử lý hình ảnh phức tạp.
* Tốn năng lượng hơn nhiều.

### 2.2.4. Còi Buzzer

Buzzer được chọn thay vì chỉ dùng LED vì:

* **Ưu điểm:**
  + Âm thanh to (70–90 dB), dễ gây chú ý.
  + Rẻ (chỉ 10.000–20.000 VNĐ).
  + Phản ứng tức thì khi nhận tín hiệu.
* **Nhược điểm:**
  + Chỉ cảnh báo cục bộ, dễ gây phiền.

**So sánh Buzzer và LED**:

**Bảng 2.3. So sánh Buzzer và LED**

| Đặc điểm | Buzzer | LED cảnh báo |
| --- | --- | --- |
| Dạng cảnh báo | Âm thanh | Ánh sáng |
| Khả năng chú ý | Rộng, dễ nhận ra | Bị hạn chế nếu không nhìn |
| Giá thành | Rất rẻ | Rẻ |

### 2.2.5. Ngôn ngữ lập trình Arduino C/C++

* **Ưu điểm**:
  + Ngôn ngữ phổ biến, cú pháp đơn giản hơn C++ thuần.
  + Có nhiều thư viện hỗ trợ sẵn cho cảm biến, IoT.
  + Tích hợp trực tiếp với Arduino IDE.
* **Nhược điểm**:
  + Không tối ưu như C++ gốc.
  + Hạn chế khi phát triển ứng dụng phức tạp.

### 2.2.6. Arduino IDE

Arduino IDE được chọn vì:

* Giao diện thân thiện, dễ học.
* Tích hợp sẵn trình biên dịch và công cụ nạp code.
* Cộng đồng người dùng rất lớn, dễ tìm tài liệu.

So với PlatformIO: Arduino IDE đơn giản hơn, phù hợp cho sinh viên và dự án nhỏ.

### 2.2.7. Firebase và MQTT

Hai công nghệ được xem xét để mở rộng hệ thống IoT:

* **Firebase**: lưu dữ liệu thời gian thực, dễ kết nối với ứng dụng smartphone.
* **MQTT**: giao thức truyền dữ liệu nhẹ, nhanh, phù hợp IoT.

**Firebase và MQTT:**

**Bảng 2.4. Firebase và MQTT**

| **Đặc điểm** | **Firebase** | **MQTT** |
| --- | --- | --- |
| Mục tiêu | Lưu trữ dữ liệu + cảnh báo | Truyền tin nhắn IoT |
| Ưu điểm | Kết nối dễ, có giao diện | Nhẹ, tốc độ nhanh |
| Nhược điểm | Cần Internet ổn định | Cần cấu hình broker |
| Ứng dụng | App giám sát, quản lý dữ liệu | IoT công nghiệp, smart home |

### 2.2.8. Nền tảng Blynk (Platform Blynk)

Blynk là một nền tảng IoT (Internet of Things) cho phép người dùng dễ dàng kết nối các vi điều khiển như ESP32, ESP8266, Arduino, Raspberry Pi… với ứng dụng di động thông qua Internet. Điểm nổi bật của Blynk là cung cấp sẵn ứng dụng trên iOS và Android với giao diện trực quan, trong đó người dùng có thể kéo – thả các widget (nút nhấn, biểu đồ, hiển thị giá trị cảm biến, thông báo, v.v.) để tạo thành ứng dụng mà không cần lập trình phức tạp.

Cách hoạt động của Blynk bao gồm 3 thành phần chính:

* **Blynk App**: ứng dụng trên điện thoại, nơi người dùng theo dõi dữ liệu cảm biến và điều khiển thiết bị.
* **Blynk Server**: đóng vai trò trung gian, lưu trữ và xử lý dữ liệu giữa thiết bị phần cứng và ứng dụng. Người dùng có thể dùng server miễn phí của Blynk hoặc tự cài đặt server riêng.
* **Blynk Library**: thư viện cài trên vi điều khiển (ESP32, Arduino…) giúp thiết bị giao tiếp với server qua WiFi, Ethernet hoặc 4G.

**Ưu điểm của Blynk:**

* Dễ sử dụng, triển khai nhanh.
* Có sẵn nhiều widget hỗ trợ giám sát và điều khiển.
* Hỗ trợ đa nền tảng, nhiều loại vi điều khiển khác nhau.

**Hạn chế**:

* Phụ thuộc vào máy chủ Blynk Cloud nếu không tự triển khai server riêng.
* Giao diện ứng dụng ít tùy biến, khó mở rộng các chức năng đặc thù (ví dụ: tích hợp gửi email cảnh báo theo kịch bản riêng).

Trong phạm vi đề tài này, nhóm không sử dụng Blynk mà lựa chọn MIT App Inventor và Firebase/MQTT để xây dựng ứng dụng tùy chỉnh, đảm bảo linh hoạt hơn trong việc hiển thị dữ liệu và gửi cảnh báo Gmail.

## 2.3. Cách tiếp cận, giải quyết vấn đề:

### 2.3.1. Mô hình tiếp cận

Hệ thống được xây dựng dựa trên kiến trúc IoT 3lớp:

1. **Lớp cảm biến**: MQ-2 và Flame Sensor thu thập dữ liệu môi trường.
2. **Lớp xử lý**: ESP32 so sánh ngưỡng, xử lý đa điều kiện.
3. **Lớp ứng dụng**: Buzzer cảnh báo tức thì, Firebase/MQTT gửi dữ liệu từ xa.

Mô hình này cho phép mở rộng linh hoạt, phù hợp với định hướng phát triển hệ thống IoT hiện đại.

### 2.3.2. Phương pháp phát triển hệ thống:

Nhóm áp dụng Prototype + Iterative Development:

* Giai đoạn 1 – Nguyên mẫu cơ bản:
  + ESP32 + MQ-2 + Buzzer.
  + Viết chương trình đọc MQ-2, so sánh ngưỡng, bật Buzzer.
* Giai đoạn 2 – Kiểm thử:
  + Tạo môi trường thử với khói nhang, bật lửa, khí gas.
  + Điều chỉnh ngưỡng MQ-2 để giảm báo giả.
* Giai đoạn 3 – Mở rộng cảm biến:
  + Thêm Flame Sensor.
  + Kết hợp dữ liệu 2 cảm biến → nếu khói + lửa thì báo động mạnh, nếu chỉ có khói thì cảnh báo nhẹ.
* Giai đoạn 4 – IoT:
  + Kết nối WiFi.
  + Gửi dữ liệu cảm biến lên Firebase hoặc MQTT.
  + Tích hợp smartphone để nhận cảnh báo.

Phương pháp này đảm bảo vừa có sản phẩm sớm, vừa dễ điều chỉnh theo thực tế.

### 2.3.3. Quy trình thực hiện

Quy trình triển khai gồm 7 bước chính:

1. **Khảo sát yêu cầu:**
   * Nghiên cứu nguyên nhân cháy phổ biến trong gia đình và nhà xưởng.
   * Xác định nhu cầu người dùng: cảnh báo tại chỗ + từ xa.
2. **Thiết kế hệ thống:**
   * Vẽ sơ đồ khối phần cứng (ESP32, MQ-2, Flame Sensor, Buzzer).
   * Thiết kế thuật toán xử lý dữ liệu (ngưỡng MQ-2, điều kiện Flame Sensor).
3. **Xây dựng phần cứng:**
   * Lắp mạch trên breadboard.
   * Nguồn cấp 5V cho ESP32 và cảm biến.
4. **Lập trình phần mềm:**
   * Viết code bằng Arduino C/C++.
   * Thử nghiệm đọc tín hiệu bằng analogRead() và digitalRead().
   * Điều khiển Buzzer bằng digitalWrite().
5. **Tích hợp IoT:**
   * Kết nối WiFi.
   * Lưu dữ liệu lên Firebase hoặc gửi MQTT đến điện thoại.
6. **Kiểm thử:**
   * Mô phỏng cháy bằng khói nhang, khí gas, bật lửa.
   * Đánh giá độ nhạy, độ chính xác, khả năng báo giả.
7. **Hoàn thiện:**
   * Tối ưu code.
   * Đóng gói mạch.
   * Viết hướng dẫn sử dụng hệ thống.

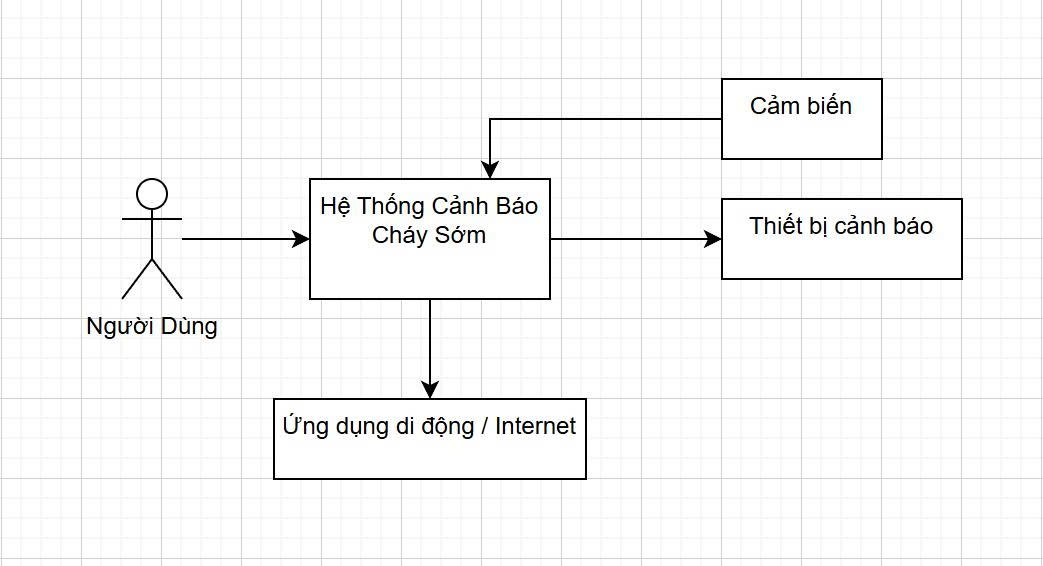
# CHƯƠNG 3 :PHÂN TÍCH-THIẾT KẾ

## 3.1. Các yêu cầu chức năng:

### 3.1.1. Ngữ cảnh sử dụng:

#### 3.1.1.1.Danh sách actor:

* **Người dùng:** Chủ hộ, quản lý ký túc xá/nhà trọ, nhân viên giám sát.
* **Cảm biến:** MQ-2 (khói, khí gas), cảm biến lửa.
* **Thiết bị cảnh báo:** Chuông/còi Buzzer, đèn LED báo động.
* **Ứng dụng di động/Internet:** Nhận dữ liệu từ hệ thống, gửi cảnh báo đến điện thoại.
* **Hệ thống cảnh báo cháy sớm:** Trung tâm xử lý (ESP32) tiếp nhận dữ liệu, ra quyết định và truyền tin.



**Hình 3.1. Sơ đồ ngữ cảnh hệ thống cảnh báo cháy sớm**

#### 3.1.1.2. Sơ đồ use case chính:



**Hình 3.2. Hệ thống cảnh báo cháy sớm**

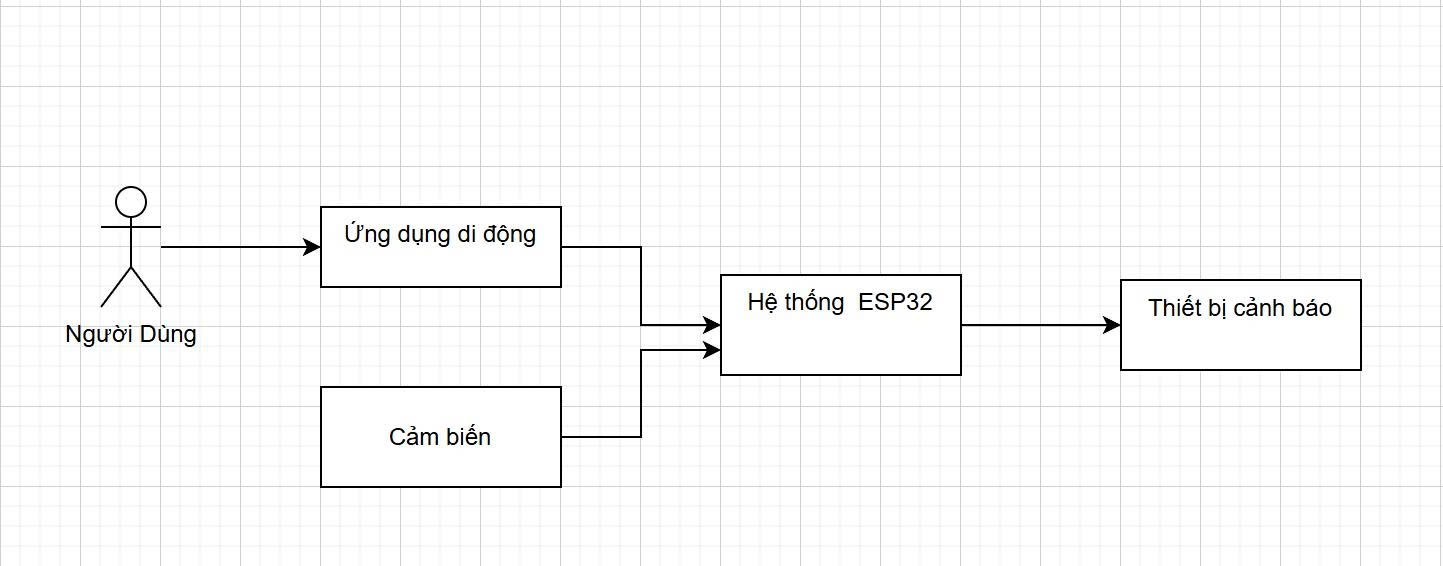
## 3.2. Các yêu cầu phi chức năng:

* **Tốc độ:** Thời gian phát hiện cháy ≤ 5 giây.
* **Độ tin cậy:** Hoạt động liên tục, ổn định 24/7.
* **An toàn:** Khi mất kết nối Internet vẫn báo động tại chỗ.
* **Dễ sử dụng:** Người không có chuyên môn vẫn thao tác được.
* **Chi phí:** Thiết bị giá rẻ, phù hợp hộ gia đình/nhà trọ.

## 3.3. Mô hình hệ thống:

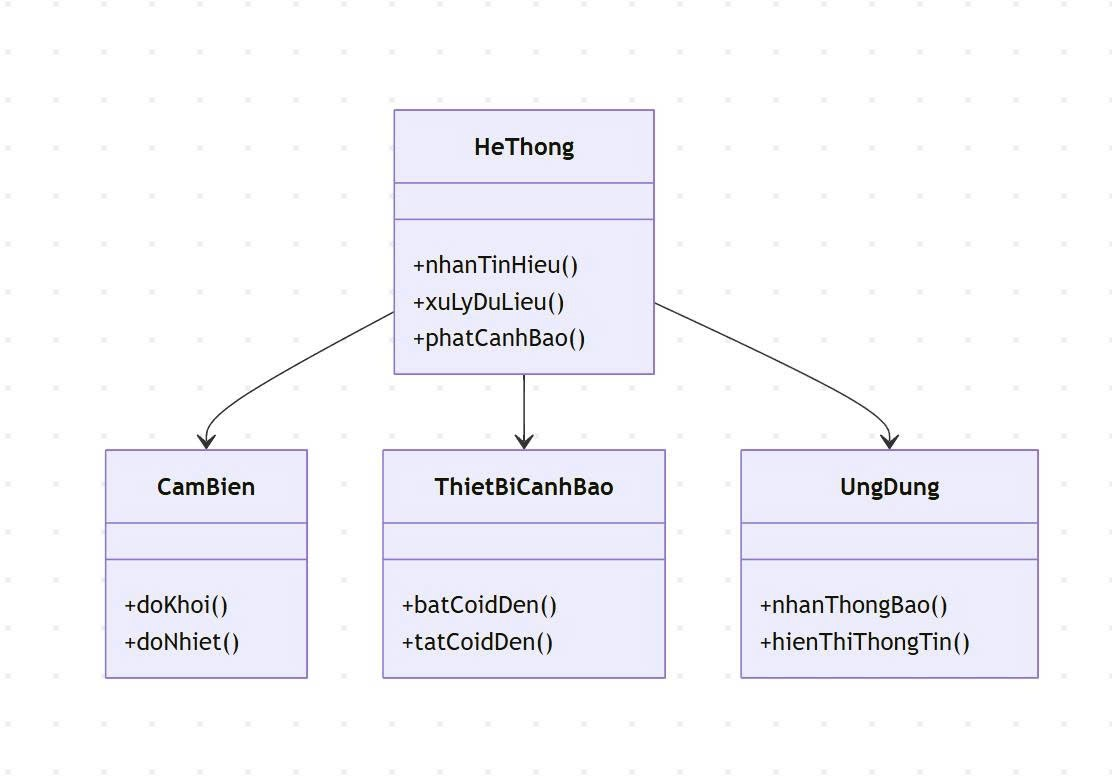
### 3.3.1. Sơ đồ đối tượng /lớp thực thể (entities object/class diagram):

1. **Sơ đồ đối tượng:**

****

**Hình 3.3. Sơ đồ đối tượng**

1. **lớp thực thể:**

****

**Hình 3.4. Sơ đồ lớp (Class diagram)**

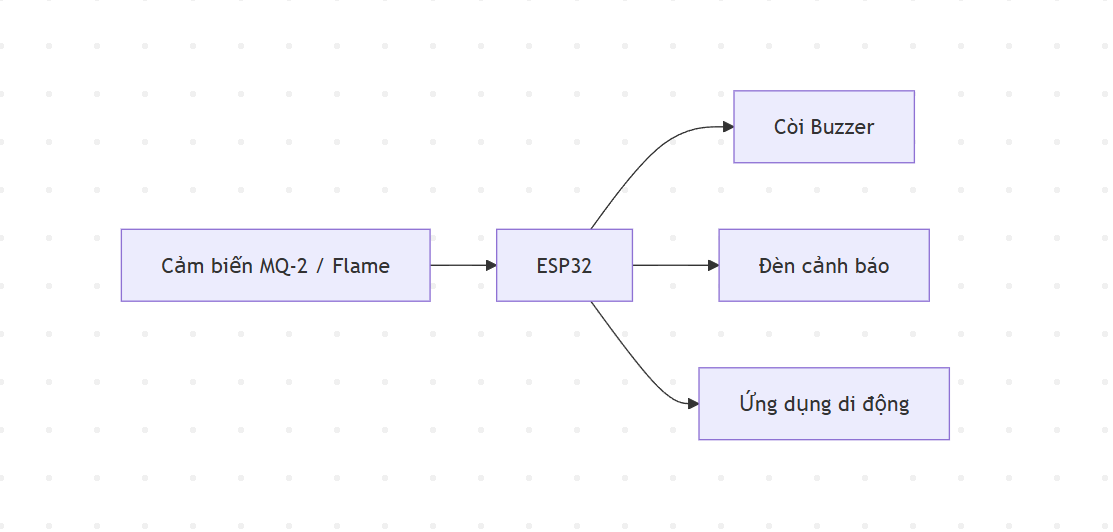
### 3.3.2. Kiến trúc phần mềm, sơ đồ thành phần:

1. **Kiến trúc phần mềm:**

Hệ thống được thiết kế thành 3 lớp chính:

* **Lớp cảm biến (Sensor Layer):** gồm MQ-2, Flame sensor → thu thập dữ liệu.
* **Lớp xử lý (Processing Layer):** vi điều khiển ESP32 → xử lý dữ liệu, xác định cháy.
* **Lớp hiển thị & cảnh báo (Application Layer):** còi, đèn, ứng dụng di động → thông báo cho người dùng

1. **sơ đồ thành phần:**

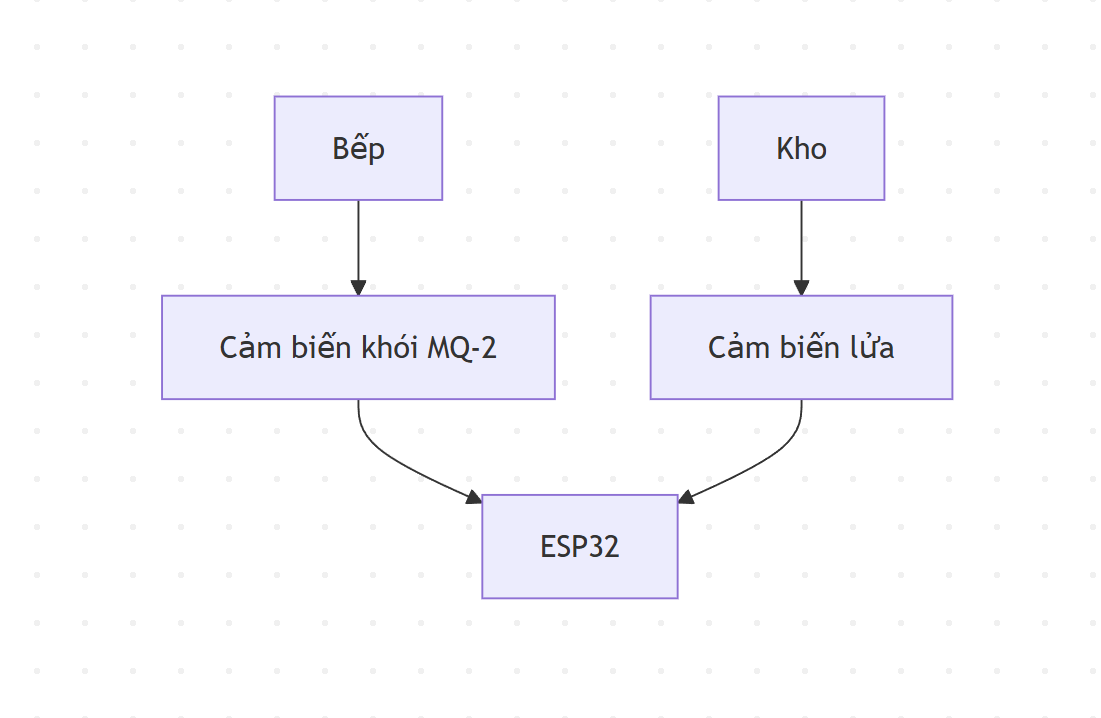


**Hình 3.5. . Sơ đồ thành phần hệ thống**

### 3.3.3. Sơ đồ bố trí, triên khai trên hệ thống phần cứng, mạng:

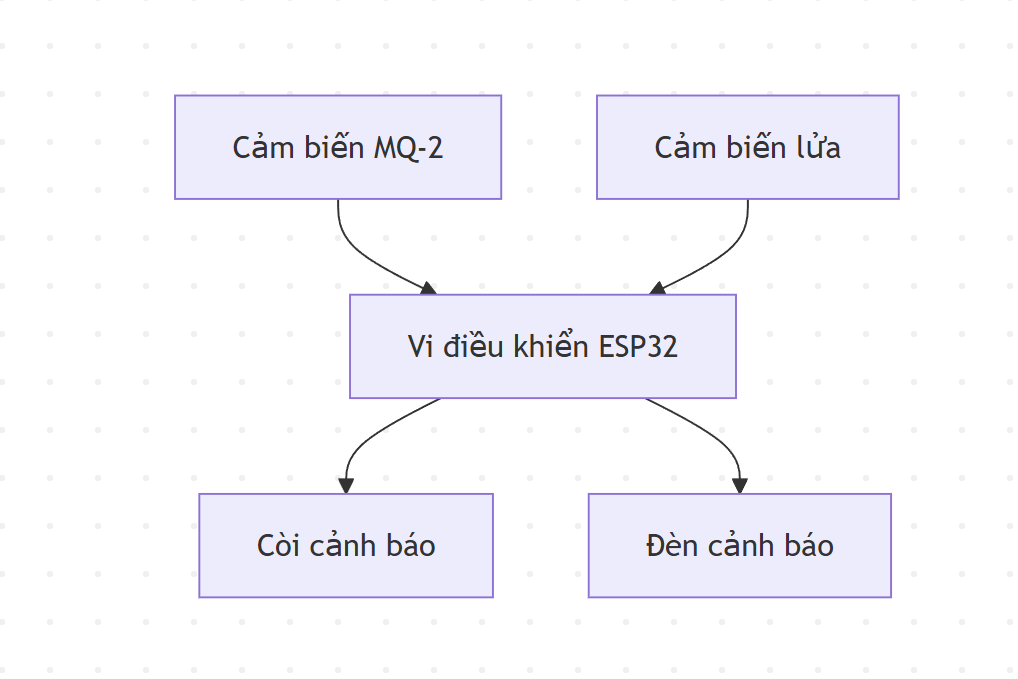
1. **Sơ đồ bố trí:**

Cảm biến được bố trí ở những vị trí dễ xảy ra cháy (nhà bếp, kho).  
ESP32 và còi/đèn được đặt tại khu vực trung tâm, dễ quan sát.



**Hình 3.6. Sơ đồ bố trí cảm biến**

1. **Triên khai trên hệ thống phần cứng:**



**Hình 3.7. Sơ đồ triển khai phần cứng**

1. **Mạng (deployment diagram):**



**Hình 3.8. Sơ đồ triển khai mạng hệ thống**

## 3.4. Mô hình xử lý /tương tác:

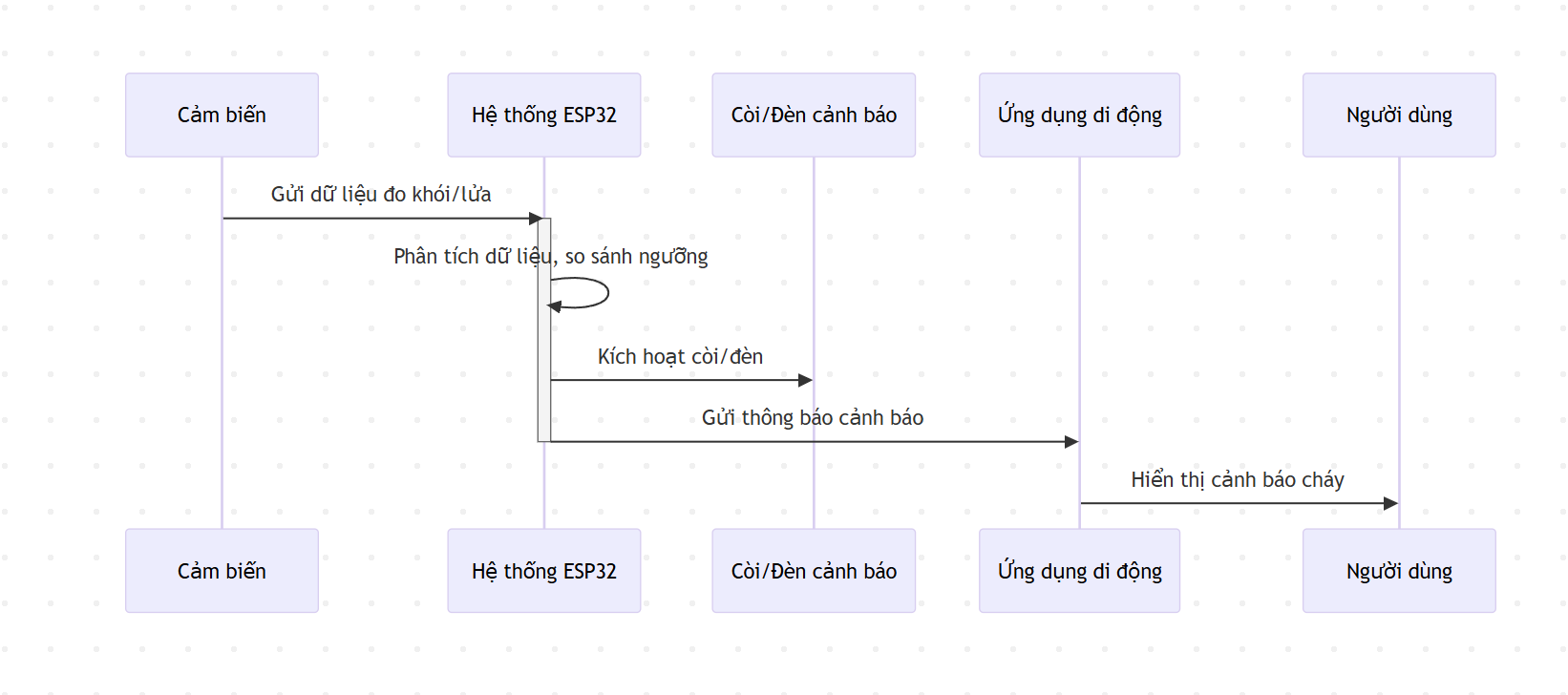
### 3.4.1. Use case chi tiết:

**Use case 1: Nhận cảnh báo cháy:**

**Bảng 3.1. Đặc tả chi tiết các use case chính**

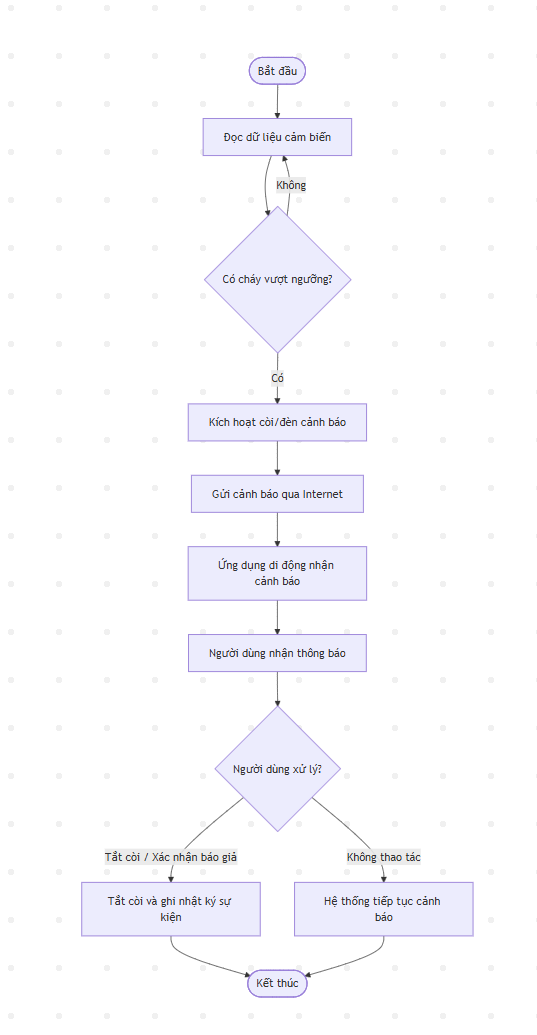
| **Tên Use case** | **Actor** | **Mô tả** | **Tiền điều kiện** | **Hậu điều kiện** | **Luồng sự kiện chính** |
| --- | --- | --- | --- | --- | --- |
| **Nhận cảnh báo cháy** | Người dùng | Khi cảm biến phát hiện cháy, hệ thống kích hoạt còi/đèn và gửi thông báo đến ứng dụng di động. | Hệ thống hoạt động, cảm biến và ESP32 kết nối ổn định. | Người dùng nhận cảnh báo kịp thời. | 1. Cảm biến phát hiện cháy  2. ESP32 xử lý dữ liệu  3. ESP32 bật còi/đèn  4. ESP32 gửi cảnh báo qua Internet  5. Ứng dụng hiển thị cảnh báo cho người dùng. |
| **Tắt còi / Xác nhận báo giả** | Người dùng | Người dùng có thể tắt còi báo động khi xác định đây là cảnh báo giả. | Hệ thống đang ở trạng thái báo động. | Còi/đèn được tắt, hệ thống ghi nhật ký sự kiện. | 1. Hệ thống kích hoạt báo động  2. Người dùng mở ứng dụng  3. Người dùng chọn “Tắt còi”  4. Ứng dụng gửi lệnh đến ESP32  5. ESP32 tắt còi/đèn và lưu nhật ký. |

### 3.4.2. Sơ đồ tuần tự (sequence diagram):



**Hình 3.9. Sơ đồ tuần tự**

### 3.4.3. Sơ đồ hoạt động (activity diagram):



**Hình 3.10. Sơ đồ hoạt động**

## 3.5. Thiết kế nguyên mẫu giao diện người dùng:

### 3.5.1. Hệ thống màn hình:

Ứng dụng di động được thiết kế để người dùng có thể dễ dàng quan sát và điều khiển hệ thống cảnh báo cháy sớm. Các màn hình chính gồm:

* **Màn hình hiển thị nồng độ khí gas:** hiển thị giá trị đo từ cảm biến MQ-2 theo thời gian thực.
* **Màn hình điều khiển còi:** cung cấp nút bật/tắt còi để người dùng thao tác khi có báo động.
* **Màn hình đổi màu khi phát hiện cháy:** khi hệ thống phát hiện khói/lửa vượt ngưỡng, giao diện đổi màu cảnh báo và hiển thị thông báo.
* **Màn hình gửi cảnh báo Gmail:** khi phát hiện cháy, hệ thống gửi email cảnh báo tự động đến người dùng.

**A screenshot of a computer

AI-generated content may be incorrect.**

**Hình 3.11. Giao diện chính của ứng dụng cảnh báo cháy sớm**

**A screenshot of a device

AI-generated content may be incorrect.**

**A white background with black text

AI-generated content may be incorrect.**

**Hình 3.12. Điều khiển còi**

**A green circle with black outline

AI-generated content may be incorrect.**

**Hình 3.13. Đổi màu khi phát hiện có lửa**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Hình 3.14. Gửi cảnh báo về gmail khi phát hiện nồng độ khí ga cao bất ngờ**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Hình 3.15. gửi cảnh báo về gmail khi phát hiện có lửa**

## 3.6. Thiết kế chi tiết:

Hệ thống được xây dựng trên nền tảng ESP32 kết hợp các cảm biến và thiết bị cảnh báo. Các thành phần chính:

* **ESP32**: vi điều khiển trung tâm, xử lý tín hiệu và kết nối WiFi.
* **Cảm biến MQ-2**: phát hiện khói và khí gas.
* **Cảm biến lửa (Flame Sensor)**: phát hiện ánh sáng ngọn lửa.
* **Buzzer & LED**: thiết bị cảnh báo trực tiếp bằng âm thanh và ánh sáng.
* **Nguồn điện 5V**: cấp nguồn cho ESP32 và cảm biến.

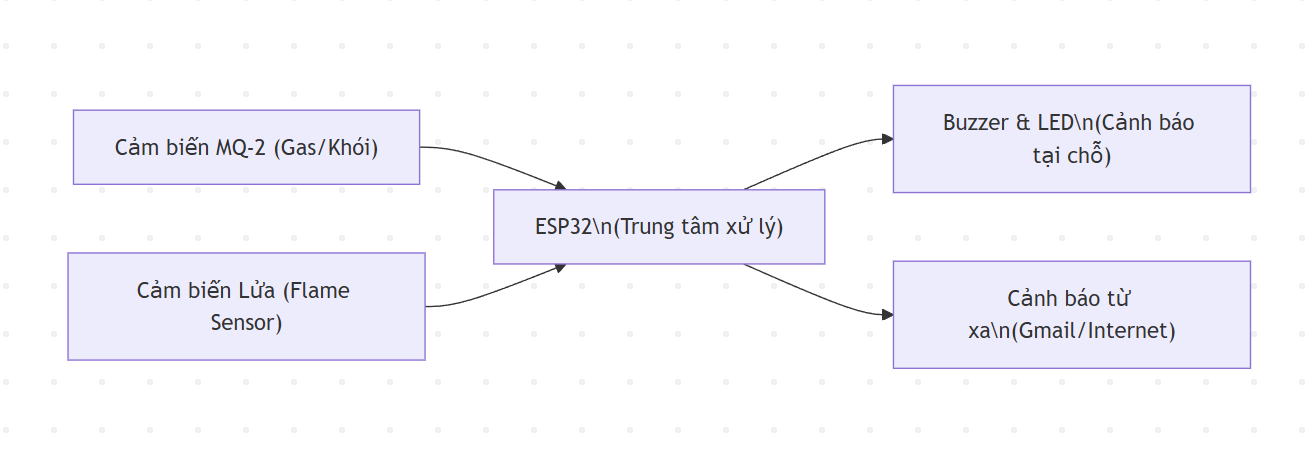
Các phần trên được kết nối với nhau qua các chân GPIO của ESP32.

### 3.6.1. Triển khai mô hình tổng quát:

Các kết nối chính:

* MQ-2 (AOUT) → GPIO34 ESP32.
* Flame Sensor (DOUT) → GPIO35 ESP32.
* Buzzer (chân tín hiệu) → GPIO26 ESP32.
* Nguồn: 5V cấp cho toàn hệ thống.

**Sơ đồ khối phần cứng**:



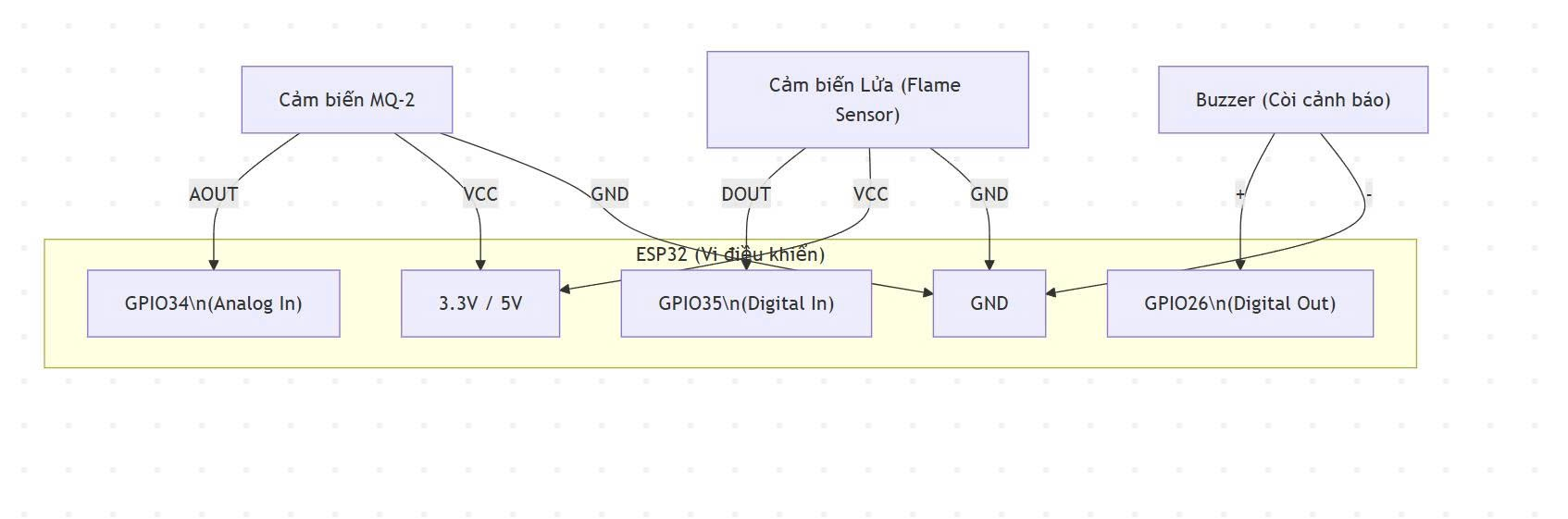
**Hình 3.16. Sơ đồ khối phần cứng**

**Bảng kết nối ESP32 – MQ-2 – Flame Sensor – Buzzer:**

**Bảng 3.2. Bảng kết nối ESP32**

| **Thiết bị** | **Chân module** | **Chân ESP32** | **Ghi chú** |
| --- | --- | --- | --- |
| **Cảm biến MQ-2** | AOUT | GPIO34 | Đọc tín hiệu analog nồng độ khí gas |
|  | VCC | 3.3V / 5V | Nếu dùng 5V, cần chia áp (R1/R2) trước khi đưa vào ADC |
|  | GND | GND | Mass chung |
| **Cảm biến Lửa** | DOUT | GPIO35 | Đọc tín hiệu digital phát hiện lửa |
|  | VCC | 3.3V | Nguồn 3.3V ổn định |
|  | GND | GND | Mass chung |
| **Buzzer (Còi cảnh báo)** | + | GPIO26 | Xuất tín hiệu điều khiển. Khuyên dùng NPN transistor + diode để bảo vệ chân GPIO |
|  | – | GND | Mass chung |

**Các phần tử trong hệ thống được kết nối với vi điều khiển ESP32 theo sơ đồ:**

****

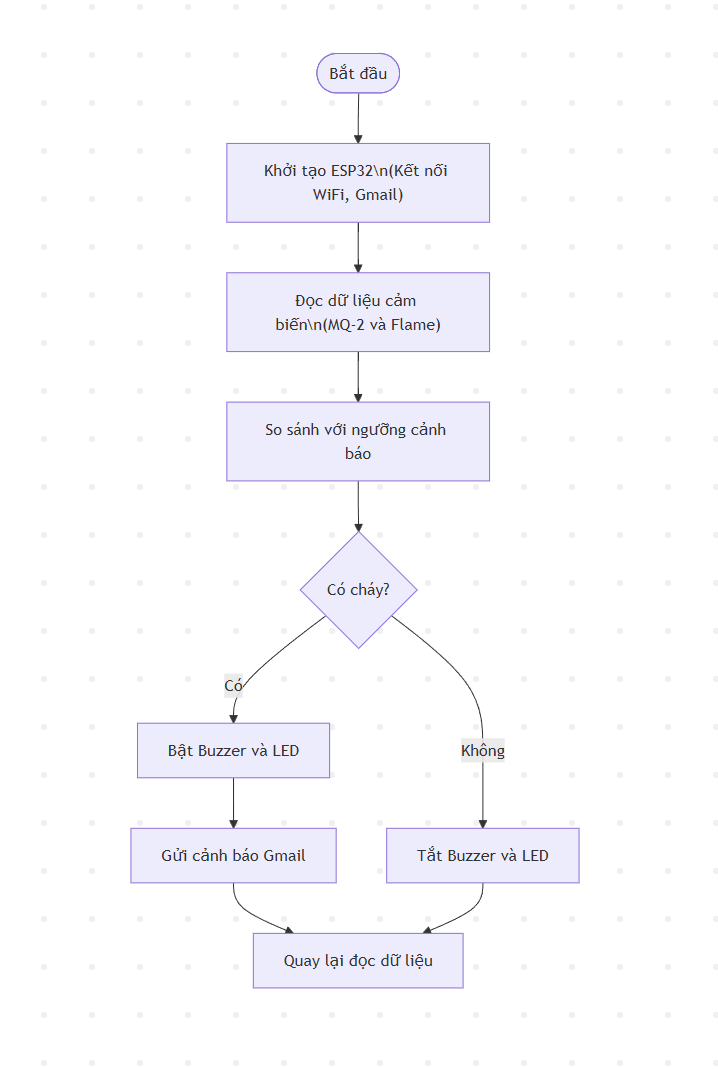
**Hình 3.17. Sơ đồ nối dây giữa ESP32 và các cảm biến trong hệ thống**

### 3.6.2. Thuật toán xử lý:

Hệ thống hoạt động theo các bước:

1. ESP32 khởi tạo kết nối WiFi và Gmail API.
2. Đọc giá trị từ cảm biến MQ-2 và Flame Sensor.
3. So sánh giá trị cảm biến với ngưỡng cảnh báo.
4. Nếu vượt ngưỡng → bật Buzzer, LED, gửi Gmail cảnh báo.
5. Nếu không vượt ngưỡng → Buzzer tắt.
6. Lặp lại vòng lặp đọc dữ liệu.

**Lưu đồ thuật toán**:



**Hình 3.18. Lưu đồ thuật toán:**

# CHƯƠNG 4:KẾT QUẢ VÀ THỰC NGHIỆM

## 4.1. Các kịch bản thử nghiệm:

Để đánh giá hệ thống cảnh báo cháy sớm, tiến hành thử nghiệm theo các kịch bản bám sát code đã lập trình và mô hình phần cứng như sau:

**Kịch bản 1: Phát hiện khí gas**

* **Mô tả:** Đặt nguồn khói/khí gas (nhang, bật lửa gas mini) gần cảm biến MQ-2.
* **Mong đợi:**
  + ESP32 đọc giá trị ADC từ MQ-2 vượt ngưỡng → bật Buzzer + LED.
  + Ứng dụng di động đổi màu sang đỏ, hiển thị cảnh báo.
  + Hệ thống gửi email cảnh báo đến người dùng.

**Kịch bản 2: Phát hiện ngọn lửa**

* **Mô tả:** Đưa nguồn lửa (bật lửa, nến) gần cảm biến Flame sensor.
* **Mong đợi:**
  + Tín hiệu DOUT từ Flame sensor ở mức LOW → ESP32 kích hoạt cảnh báo.
  + Buzzer + LED hoạt động, ứng dụng đổi màu, gửi email.

**Kịch bản 3: Trạng thái bình thường**

* **Mô tả:** Không có khói/lửa trong môi trường thử nghiệm.
* **Mong đợi:**
  + MQ-2 và Flame sensor trả giá trị trong ngưỡng an toàn.
  + Buzzer + LED tắt, ứng dụng hiển thị màu xanh (Safe), không gửi email.

**Kịch bản 4: So sánh 2 mô hình cảm biến**

* Mô tả:
  + **Mô hình A**: chỉ sử dụng MQ-2.
  + **Mô hình B**: sử dụng MQ-2 kết hợp Flame sensor.
* Mục đích: So sánh độ chính xác và hạn chế báo động giả.
* Mong đợi:
  + Mô hình A dễ bị báo động giả khi có khói loãng (ví dụ khói nhang).
  + Mô hình B giảm báo động giả nhờ yêu cầu cả khói + lửa để kích hoạt báo động.

**Kịch bản 5: Mất kết nối WiFi**

* Mô tả: Ngắt WiFi khi hệ thống đang hoạt động.
* Mong đợi:
  + ESP32 không gửi được Gmail cảnh báo.
  + Tuy nhiên**,** Buzzer + LED vẫn bật để đảm bảo an toàn tại chỗ.

## 4.2. Kết quả thử nghiệm các kịch bản:

Bảng dưới đây tổng hợp kết quả thực nghiệm của hệ thống:

**Bảng 4.1. Kết quả thử nghiệm các kịch bản**

| **Kịch bản** | **Mô tả thử nghiệm** | **Kết quả quan sát** | **Đánh giá** |
| --- | --- | --- | --- |
| **1: Phát hiện khí gas** | Đốt nhang / bật lửa gas gần MQ-2 | Sau ~3–5 giây, giá trị ADC vượt ngưỡng → Buzzer + LED bật, ứng dụng đổi màu đỏ, Gmail cảnh báo được gửi thành công | Hệ thống phản ứng đúng, thời gian phát hiện nhanh |
| **2: Phát hiện ngọn lửa** | Đưa bật lửa/nến gần Flame sensor | Sau ~1–2 giây, DOUT = LOW → Buzzer + LED bật, ứng dụng đổi màu đỏ, Gmail cảnh báo được gửi thành công | Hệ thống phát hiện chính xác ngọn lửa |
| **3: Trạng thái bình thường** | Không có khói/lửa | MQ-2 và Flame trong ngưỡng an toàn → Buzzer + LED tắt, ứng dụng màu xanh, không gửi email | Hệ thống ổn định, không báo động giả |
| **4: So sánh MQ-2 và MQ-2 + Flame** | Đốt nhang (khói loãng) | - Mô hình A (MQ-2): có cảnh báo giả.- Mô hình B (MQ-2 + Flame): không cảnh báo. | Mô hình B hiệu quả hơn, giảm báo động giả |
| **5: Mất WiFi** | Ngắt kết nối WiFi khi thử lửa | Buzzer + LED vẫn bật, nhưng Gmail không gửi được; log Serial báo “WiFi disconnected” | Hệ thống vẫn đảm bảo cảnh báo tại chỗ, có khả năng dự phòng |

## 4.3. Xử lý các trường hợp ngoại lệ:

Trong quá trình vận hành, hệ thống có thể gặp một số tình huống bất thường. Các cơ chế xử lý được thiết kế như sau:

* **Mất kết nối WiFi**: Trước khi gửi email, hệ thống luôn kiểm tra trạng thái kết nối. Nếu WiFi bị ngắt, quá trình gửi Gmail sẽ không thực hiện, nhưng Buzzer và LED vẫn hoạt động bình thường để cảnh báo tại chỗ.
* **Cảm biến MQ-2 trả giá trị ngoài ngưỡng**: MQ-2 được đọc qua chân analog. Nếu giá trị vượt ngưỡng định sẵn → hệ thống coi là có khói/khí gas và bật báo động. Nếu giá trị dưới ngưỡng → coi là an toàn.
* **Cảm biến Flame**: Được đọc qua tín hiệu digital. Nếu trả về mức LOW → coi là có lửa, kích hoạt báo động. Nếu luôn HIGH (kể cả khi có lửa) thì coi như cảm biến lỗi, hệ thống chỉ dựa vào MQ-2 để báo cháy.
* **Báo động giả**: Do code hiện tại dùng điều kiện hoặc (OR) giữa hai cảm biến, hệ thống vẫn có thể báo động giả nếu chỉ có khói loãng hoặc ánh sáng mạnh giống ngọn lửa. Đây là hạn chế của mô hình.

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết quả đối chiếu với mục tiêu:

Dựa trên các mục tiêu đã đề ra ở Mục 1.4, nhóm tiến hành so sánh giữa mục tiêu ban đầu và kết quả thực tế. Bảng sau thể hiện chi tiết mức độ hoàn thành:

**Bảng 5.1. Kết quả đối chiếu với mục tiêu**

| **Mục tiêu đề ra**  **(Mục 1.4)** | **Kết quả thực hiện** | **Đạt/Không đạt** | **Giải thích** |
| --- | --- | --- | --- |
| Xây dựng hệ thống cảnh báo cháy sớm với chi phí thấp | Hệ thống được triển khai bằng ESP32, MQ-2, Flame Sensor, Buzzer – chi phí < 300.000 VNĐ | Đạt | Sử dụng linh kiện giá rẻ, dễ tìm, phù hợp sinh viên. |
| Phát hiện khói/khí gas rò rỉ | MQ-2 hoạt động ổn định, phát hiện được khói và khí gas trong các thử nghiệm | Đạt | Tuy đôi khi vẫn có báo giả trong môi trường có khói bếp, hơi cồn. |
| Phát hiện ngọn lửa trực tiếp | Flame Sensor phát hiện lửa nhanh (khoảng cách 20–100 cm), cảnh báo tức thì | Đạt | Hoạt động tốt trong môi trường thử nghiệm, nhưng hạn chế ngoài trời. |
| Cảnh báo tức thì bằng âm thanh (Buzzer) | Khi có khói/lửa, Buzzer kích hoạt ngay, âm thanh 70–90 dB | Đạt | Thích hợp cảnh báo cục bộ trong phòng/ký túc xá. |
| Tích hợp IoT để giám sát từ xa (Firebase/MQTT) | Đã thử nghiệm kết nối WiFi và gửi dữ liệu; tuy nhiên chức năng còn đơn giản, chưa có giao diện đẹp mắt | Một phần đạt | Chỉ mới gửi dữ liệu lên server; chưa có ứng dụng di động riêng. |
| Hệ thống dễ mở rộng, tích hợp thêm cảm biến khác | Kiến trúc mở, có thể bổ sung nhiệt độ, độ ẩm, camera IR | Đạt | Chưa triển khai, nhưng phần cứng và code có khả năng mở rộng. |

## 5.2. Các hạn chế của đồ án:

Mặc dù hệ thống đã cơ bản đáp ứng được yêu cầu, nhưng vẫn còn một số hạn chế:

1. **Độ chính xác của cảm biến MQ-2**:
   * Dễ bị nhiễu bởi khói bếp, thuốc lá, hơi cồn → gây báo giả.
   * Cần hiệu chỉnh thêm để tối ưu.
2. **Khoảng cách phát hiện của Flame Sensor**:
   * Chỉ hiệu quả trong phạm vi ngắn (20–100 cm).
   * Không phù hợp cho không gian rộng hoặc ngoài trời.
3. **Nguồn điện**:
   * Hệ thống phụ thuộc nguồn điện trực tiếp.
   * Khi mất điện, hệ thống không hoạt động.
4. **Chức năng IoT**:
   * Mới dừng ở mức gửi dữ liệu cơ bản lên Firebase/MQTT.
   * Chưa có ứng dụng di động riêng, giao diện trực quan còn thiếu.
5. **Chưa tích hợp hệ thống chữa cháy tự động**:
   * Hệ thống chỉ cảnh báo, chưa có khả năng kích hoạt quạt, phun nước, đóng van gas.

## 5.3. Hướng phát triển:

Để nâng cao hiệu quả, hệ thống có thể phát triển theo các hướng sau:

1. **Nâng cấp cảm biến**
   * Thay MQ-2 bằng cảm biến đa khí (MQ-135, CCS811) để tăng độ chính xác.
   * Bổ sung cảm biến nhiệt độ – độ ẩm (DHT11/DHT22) để phát hiện tăng nhiệt bất thường.
   * Kết hợp camera hồng ngoại để xác nhận cháy, giảm báo giả.
2. **Hoàn thiện IoT và ứng dụng di động**
   * Xây dựng ứng dụng Android/iOS riêng để hiển thị dữ liệu, cảnh báo real-time.
   * Tích hợp bản đồ để xác định vị trí cháy.
   * Kết nối nhiều hệ thống thành mạng giám sát IoT tập trung.
3. **Phát triển phần cứng nâng cao**
   * Bổ sung pin Lithium + mạch sạc để hoạt động khi mất điện.
   * Tích hợp module GSM/4G để gửi SMS/cảnh báo khi không có WiFi.
   * Thêm relay điều khiển thiết bị chữa cháy (bơm nước, quạt thông gió, đóng van gas).
4. **Ứng dụng rộng rãi**
   * Áp dụng cho nhà ở, ký túc xá, nhà trọ, nhà xưởng.
   * Tích hợp vào hệ thống nhà thông minh (Smart Home).
   * Phát triển thành sản phẩm thương mại giá rẻ phục vụ cộng đồng.

# TÀI LIỆU THAM KHẢO

[1] Bahga, A., & Madisetti, V. (2014). Internet of Things: A hands-on approach. Universities Press.

[2] Patel, N., & Sharma, R. (2021). IoT-based fire detection and monitoring system. International Journal of Computer Applications, 183(28), 12–18. https://doi.org/10.5120/ijca2021921512

[3] Espressif Systems. (2022). ESP32 technical reference manual. https://www.espressif.com

[4] Blynk IoT Platform. (2023). Documentation and API reference. https://blynk.io

[5] Cục Cảnh sát PCCC và CNCH. (2023). Thống kê tình hình cháy, nổ tại Việt Nam. Bộ Công an Việt Nam.

[6] Arduino. (2022). Arduino IDE 2.0 Documentation. https://docs.arduino.cc

[7] MQ Gas Sensors. (2021). Technical data sheet for MQ-2 gas sensor. Hanwei Electronics.

[8] Firebase. (2023). Firebase Realtime Database Documentation. Google Developers. https://firebase.google.com/docs

[9] MQTT. (2023). MQTT Version 3.1.1 protocol specification. OASIS Standard. https://mqtt.org