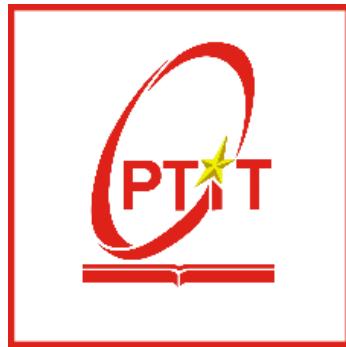


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN



**MÔN: IOT VÀ ỨNG DỤNG
BÁO CÁO GIỮA KỲ**

Đề tài: Hệ thống cảnh báo rò rỉ khí gas

Giảng viên hướng dẫn: Kim Ngọc Bách

Nhóm học phần: 05

Nhóm bài tập lớn: 14

Thành viên:

Ngô Đắc Tuấn Kiệt B22DCCN435

Ngô Tiến Hưng B22DCCN411

Ma Văn Hùng B22DCCN363

Vi Văn Hoàng B22DCCN351

HÀ NỘI

MỤC LỤC

I.	Giới thiệu	4
1.	Mục tiêu đề tài	4
a.	Mục tiêu tổng quát:.....	4
b.	Mục tiêu cụ thể:.....	4
2.	Lý do chọn đề tài	4
a.	Tầm quan trọng của đề tài	4
b.	Tính ứng dụng của đề tài	5
II.	Mô tả tổng quan	6
1.	Môi trường hoạt động	6
2.	Các ràng buộc	6
3.	Các giả định	7
III.	Xác định Yêu cầu chức năng (Functional Requirements)	8
IV.	Xác định Yêu cầu phi chức năng (Non-functional Requirements)	9
1.	Hiệu năng (Performance)	9
2.	Bảo mật (Security)	10
3.	Dộ tin cậy (Reliability)	10
4.	Khả năng mở rộng và bảo trì (Scalability & Maintainability)	10
V.	Các thiết bị sử dụng trong hệ thống	11
1.	Phần cứng	11
a.	ESP32	11
b.	Cảm biến khí gas MQ2.....	12
c.	Cảm biến phát hiện lửa.....	14
2.	Phần mềm	16
a.	Arduino IDE	16
b.	Frontend (ReactJS)	16
c.	Backend (Java Spring Boot).....	17
d.	Firebase Realtime Database	17

VI. Use Case Diagram	17
1. Use Case: Cấu hình ngưỡng cảnh báo.....	17
2. Use Case: Xem lịch sử cảnh báo	18
3. Use Case: Giải cảnh báo (SMS/web)	18
4. Use Case: Xem nồng độ khí gas hiện tại.....	18
5. Use Case: Cấu hình ESP32.....	18
6. Use Case: Quản lý người dùng.....	18
7. Use Case: Cập nhật firmware ESP32 (OTA)	18
8. Use Case: Báo cáo / Thông kê.....	19

I. Giới thiệu

1. Mục tiêu đề tài

Đề tài “**Hệ thống cảnh báo rò rỉ khí Gas sử dụng công nghệ IoT**” được xây dựng với mục tiêu tổng quát là thiết kế và triển khai một hệ thống thông minh có khả năng phát hiện, cảnh báo và xử lý tự động khi xảy ra hiện tượng rò rỉ khí gas, đảm bảo an toàn cho người sử dụng trong hộ gia đình hoặc các khu vực có nguy cơ cháy nổ cao.

a. Mục tiêu tổng quát:

Tạo ra một mô hình hệ thống IoT hoàn chỉnh, có khả năng:

- Giám sát liên tục nồng độ khí gas trong môi trường theo thời gian thực.
- Phát hiện sớm rò rỉ khí gas, cảnh báo người dùng kịp thời bằng âm thanh, tín hiệu đèn, và thông báo qua Internet.
- Điều khiển tự động các thiết bị an toàn như quạt thông gió, cửa tự động hoặc máy bơm nhằm giảm thiểu rủi ro cháy nổ.
- Cho phép người dùng giám sát và điều khiển từ xa thông qua ứng dụng web hoặc điện thoại di động nhờ tích hợp cơ sở dữ liệu đám mây (Firebase).

b. Mục tiêu cụ thể:

- Phát hiện rò rỉ khí gas bằng cảm biến MQ-2;
- Xử lý dữ liệu và cảnh báo tức thời bằng vi điều khiển ESP32;
- Lưu trữ và đồng bộ dữ liệu qua nền tảng đám mây Firebase;
- Xây dựng giao diện người dùng (Frontend);
- Phát triển backend trung gian (NodeJS);
- Đảm bảo khả năng mở rộng và tính thực tiễn;
- Nâng cao tính an toàn và ổn định của hệ thống;

2. Lý do chọn đề tài

a. Tầm quan trọng của đề tài

Trong đời sống hiện đại, khí Gas (LPG, methane, propane, butane) là nguồn năng lượng thiết yếu, được sử dụng rộng rãi trong sinh hoạt gia đình, nhà hàng, ký túc xá, khu công nghiệp... Tuy nhiên, bản chất dễ cháy và độc hại của khí gas khiến

nó trở thành một trong những nguyên nhân hàng đầu gây cháy nổ và ngộ độc nếu xảy ra rò rỉ mà không được phát hiện kịp thời.

Theo thống kê của cơ quan phòng cháy chữa cháy, mỗi năm tại Việt Nam có hàng trăm vụ cháy nổ liên quan đến khí gas, gây thiệt hại nghiêm trọng về người và tài sản. Nguyên nhân chủ yếu là thiếu hệ thống cảnh báo sớm, thiết bị kém chất lượng, hoặc con người phát hiện chậm. Trong bối cảnh đó, việc phát triển một hệ thống cảnh báo rò rỉ khí gas tự động là hết sức cần thiết và có ý nghĩa thực tiễn sâu sắc.

Sự ra đời của công nghệ Internet of Things (IoT) đã mở ra hướng đi mới trong công tác giám sát và cảnh báo an toàn. Với IoT, các cảm biến có thể kết nối Internet, gửi dữ liệu liên tục đến hệ thống trung tâm, phân tích và phản hồi tự động trong thời gian thực. Điều này giúp rút ngắn thời gian phát hiện rò rỉ, tăng khả năng phản ứng nhanh, và giảm thiểu nguy cơ cháy nổ đến mức thấp nhất.

Do đó, việc nghiên cứu và triển khai hệ thống cảnh báo rò rỉ khí gas ứng dụng IoT không chỉ có ý nghĩa về mặt kỹ thuật mà còn mang giá trị nhân văn và xã hội lớn, góp phần bảo vệ an toàn cho người dân, đặc biệt trong các khu dân cư đông đúc hoặc khu vực có nguy cơ cháy nổ cao.

b.Tính ứng dụng của đề tài

Đề tài “**Hệ thống cảnh báo rò rỉ khí Gas sử dụng công nghệ IoT**” có tính ứng dụng thực tế cao, có thể triển khai trong nhiều môi trường và quy mô khác nhau — từ hộ gia đình nhỏ đến nhà hàng, xưởng sản xuất, ký túc xá hoặc chung cư.

Cụ thể, hệ thống mang lại các ứng dụng sau:

- Giám sát an toàn theo thời gian thực
- Cảnh báo và phản ứng tự động
- Điều khiển và quản lý từ xa
- Tính mở rộng và tích hợp
- Chi phí thấp – hiệu quả cao

II. Mô tả tổng quan

1. Môi trường hoạt động

Hệ thống cảnh báo rò rỉ khí Gas được thiết kế để hoạt động trong môi trường sinh hoạt và sản xuất có sử dụng khí Gas, nơi có nguy cơ rò rỉ hoặc cháy nổ.

Các môi trường hoạt động điển hình gồm:

- Căn hộ, nhà riêng, ký túc xá, phòng trọ: nơi sử dụng bình gas mini hoặc bếp gas dân dụng.
- Nhà hàng, quán ăn, khu bếp tập thể: có hệ thống bếp gas công suất lớn cần giám sát liên tục.
- Chung cư hoặc khu dân cư cao tầng: nơi việc phát hiện rò rỉ bằng cảm biến là khó khăn.
- Xưởng sản xuất nhỏ hoặc khu công nghiệp: có sử dụng khí đốt trong dây chuyền hoặc kho chứa nhiên liệu.

Hệ thống hoạt động tốt trong các điều kiện sau:

- Nhiệt độ môi trường: 0 – 50°C.
- Độ ẩm tương đối: dưới 80%.
- Khu vực có kết nối Wi-Fi ổn định, cho phép truyền dữ liệu thời gian thực lên nền tảng Firebase.
- Không gian thông thoáng, tránh đặt cảm biến gần nguồn lửa, khói hoặc luồng gió mạnh làm sai lệch kết quả đo.

2. Các ràng buộc

Trong quá trình thiết kế và triển khai, hệ thống chịu ảnh hưởng bởi một số ràng buộc kỹ thuật và điều kiện thực tế như sau:

a. Nguồn điện:

- Hệ thống yêu cầu nguồn cấp ổn định 5V–2A để đảm bảo ESP32, cảm biến MQ-2, LCD và các thiết bị đầu ra hoạt động chính xác.
- Mất điện sẽ khiến hệ thống ngừng hoạt động, vì vậy cần có phương án nguồn dự phòng (UPS hoặc pin sạc).

b. Kết nối Internet:

- Việc đồng bộ dữ liệu và cảnh báo thời gian thực phụ thuộc hoàn toàn vào kết nối Wi-Fi.

- Nếu mạng chập chờn hoặc mất kết nối, dữ liệu tạm thời chỉ được lưu cục bộ trên ESP32, không hiển thị lên web.

c. Thời gian phản hồi:

- Hệ thống yêu cầu độ trễ giữa thời điểm phát hiện rò rỉ và thời điểm cảnh báo < 1 giây để đảm bảo an toàn.
- Mọi xử lý từ cảm biến → ESP32 → Firebase → giao diện web phải được tối ưu để phản hồi nhanh nhất.

d. Chi phí phần cứng:

- Dự án hướng đến sản phẩm giá thành thấp, dễ chế tạo, nên linh kiện được lựa chọn dựa trên tiêu chí tiết kiệm và sẵn có (ESP32, MQ-2, relay module, còi báo, servo).
- Việc giới hạn chi phí cũng đồng nghĩa với việc chưa thể sử dụng các cảm biến công nghiệp cao cấp.

e. Ràng buộc phần mềm:

- Phiên bản miễn phí của Firebase giới hạn tần suất cập nhật dữ liệu và dung lượng lưu trữ.
- Giao diện web phải đảm bảo tương thích với các trình duyệt phổ biến.

f. Ràng buộc an toàn:

- Người lắp đặt và sử dụng phải tuân thủ quy định an toàn điện và phòng chống cháy nổ.
- Hệ thống chỉ có chức năng cảnh báo, không thay thế hoàn toàn cho biện pháp an toàn cơ học như van khóa gas, cảm biến khói hoặc hệ thống chữa cháy chuyên dụng.

3. Các giả định

Khi triển khai hệ thống, nhóm đưa ra một số giả định ban đầu để đảm bảo hệ thống vận hành ổn định trong môi trường tiêu chuẩn:

- **Người dùng có sẵn kết nối Wi-Fi:** Hệ thống giả định rằng người sử dụng đã có mạng Wi-Fi ổn định để ESP32 kết nối Internet và truyền dữ liệu lên Firebase.

- **Cảm biến hoạt động ổn định:** Giả định cảm biến MQ-2 được hiệu chỉnh đúng kỹ thuật, không bị bám bụi, hư hỏng hay nhiễu điện. Việc hiệu chuẩn cảm biến được thực hiện trước khi sử dụng để tránh sai số lớn.
- **Người dùng có kiến thức cơ bản về thiết bị điện tử:** Người dùng hiểu cách cắm nguồn, bật/tắt thiết bị và thao tác cơ bản trên giao diện web. Không tự ý thay đổi phần cứng hoặc dây nối khi hệ thống đang hoạt động.
- **Nguồn điện ổn định và an toàn:** Giả định rằng môi trường có nguồn điện ổn định, không bị sụt áp hay chập điện gây lỗi hệ thống.
- **Môi trường thử nghiệm đạt tiêu chuẩn:** Hệ thống được kiểm tra trong môi trường có nồng độ khí gas kiểm soát được, không quá cao để gây cháy nổ.
- **Thiết bị kết nối đồng bộ:** Giả định rằng tất cả các thành phần (ESP32, Firebase, giao diện web) được cấu hình đúng và hoạt động đồng bộ với nhau trước khi vận hành.

III. Xác định Yêu cầu chức năng (Functional Requirements)

Phần này mô tả các hành vi và chức năng cụ thể mà hệ thống phải thực hiện, trả lời cho câu hỏi "Hệ thống sẽ làm được gì?".

Các chức năng cơ bản của hệ thống

- a. Thu thập dữ liệu nồng độ khí gas:
 - Cảm biến gas MQ-2 liên tục đo lường nồng độ khí dễ cháy (LPG, Methane, CO...) trong không khí.
 - Mẫu dữ liệu thu thập bao gồm: giá trị nồng độ, ID thiết bị, vị trí lắp đặt, và thời gian đo.
- b. Phân tích và xử lý sự kiện:
 - ESP32 (tại biên) hoặc hệ thống trên Cloud so sánh nồng độ gas nhận được với một ngưỡng an toàn đã được định cấu hình trước.
 - Nếu Nồng độ gas > Ngưỡng, hệ thống xác định đây là một sự kiện "Rò rỉ khí gas" và kích hoạt chuỗi hành động phản ứng.
- c. Gửi dữ liệu và cảnh báo từ xa:

- Dữ liệu được truyền ngay lập tức qua mạng Wi-Fi về Cloud sử dụng giao thức MQTT hoặc API của nền tảng (ví dụ: Firebase).
- Khi có sự cố, hệ thống gửi thông báo đẩy (push notification) khẩn cấp tới ứng dụng di động của người dùng.

d. Điều khiển thiết bị an toàn (Actuators):

- Đây là chức năng phản ứng tự động quan trọng nhất. Khi phát hiện rò rỉ, hệ thống sẽ đồng thời thực hiện các hành động sau:
 - Kích hoạt báo động tại chỗ: Bật còi báo động và đèn cảnh báo để thu hút sự chú ý ngay lập tức.
 - Kích hoạt hệ thống thông gió: Bật quạt hút thông qua module relay để làm loãng nồng độ gas nguy hiểm.
 - Mở lối thoát hiểm: Gửi lệnh đến các cơ cấu chấp hành như xi lanh điện hoặc khóa chốt điện tử để tự động mở cửa sổ/cửa chính.

e. Tương tác với người dùng:

- Hiển thị: Giao diện dashboard hiển thị trạng thái trực quan ("BÌNH THƯỜNG" hoặc "CẢNH BÁO") và biểu đồ nồng độ gas thời gian thực.
- Lưu trữ: Dữ liệu lịch sử được lưu trữ trên cloud để người dùng có thể xem lại và phân tích.
- Điều khiển thủ công: Người dùng có thẩm quyền có thể tắt chuông báo hoặc thay đổi ngưỡng cảnh báo từ xa qua ứng dụng.

IV. Xác định Yêu cầu phi chức năng (Non-functional Requirements)

1. Hiệu năng (Performance)

- Độ trễ đầu-cuối tại chỗ: Mục tiêu là ≤ 1 giây cho chu trình an toàn (từ lúc phát hiện đến khi tất cả các cơ cấu chấp hành tại chỗ được kích hoạt).
- Toàn bộ logic cảnh báo cốt lõi (đọc cảm biến → so sánh ngưỡng → kích hoạt còi/quạt/cửa) phải được xử lý trực tiếp trên firmware của ESP32 (Edge Computing).

- Độ trễ cảnh báo từ xa: Mục tiêu là ≤ 5 giây (từ lúc phát hiện đến khi người dùng nhận được thông báo đầy trên điện thoại). → Sử dụng Firebase Realtime Database để đồng bộ trạng thái gần như tức thời, kết hợp Firebase Cloud Functions để kích hoạt dịch vụ Firebase Cloud Messaging (FCM).

2. Bảo mật (Security)

- Mã hóa: Toàn bộ dữ liệu trên đường truyền và khi lưu trữ phải được mã hóa. → Mọi kết nối từ ESP32 đến các dịch vụ Cloud phải sử dụng thư viện hỗ trợ TLS (HTTPS/WSS). Dữ liệu trên Cloud được mã hóa tại nơi lưu trữ (at rest) theo mặc định của nhà cung cấp.
- Xác thực và phân quyền: Chỉ người dùng và thiết bị hợp lệ mới được truy cập hệ thống và chỉ được phép thao tác trong phạm vi quyền hạn. → Sử dụng Firebase Authentication để quản lý đăng nhập người dùng. Áp dụng Firebase Security Rules để đảm bảo người dùng chỉ có thể đọc/ghi dữ liệu của các thiết bị mà họ sở hữu.

3. Độ tin cậy (Reliability)

- Hoạt động ngoại tuyến: Chức năng an toàn tại chỗ BẮT BUỘC phải hoạt động 100% ngay cả khi mất kết nối Wi-Fi. → Logic an toàn cốt lõi phải được lập trình và chạy độc lập hoàn toàn trên firmware của ESP32.
- Tự phục hồi lỗi phần mềm: Thiết bị phải có khả năng tự khởi động lại nếu firmware bị treo. → Tích hợp cơ chế Watchdog Timer (WDT) vào firmware để tự động reset vi điều khiển nếu vòng lặp chính bị treo.
- Nguồn dự phòng: Hệ thống phải duy trì hoạt động trong ít nhất 2 giờ sau khi mất điện lưới. → Các thành phần quan trọng phải được kết nối với bộ lưu điện (UPS) hoặc có pin sạc dự phòng tích hợp.

4. Khả năng mở rộng và bảo trì (Scalability & Maintainability)

- Quản lý nhiều thiết bị: Một tài khoản người dùng phải có khả năng quản lý nhiều thiết bị cảnh báo. → Quyết định thiết kế: Thiết kế cấu trúc dữ liệu trên Cloud theo hướng users/{uid}/devices/{deviceId}.
- Cập nhật firmware từ xa: Hệ thống phải cho phép nâng cấp phần mềm cho thiết bị mà không cần can thiệp vật lý. → Quyết định thiết kế: Tích hợp tính năng OTA (Over-The-Air) Update cho ESP32.

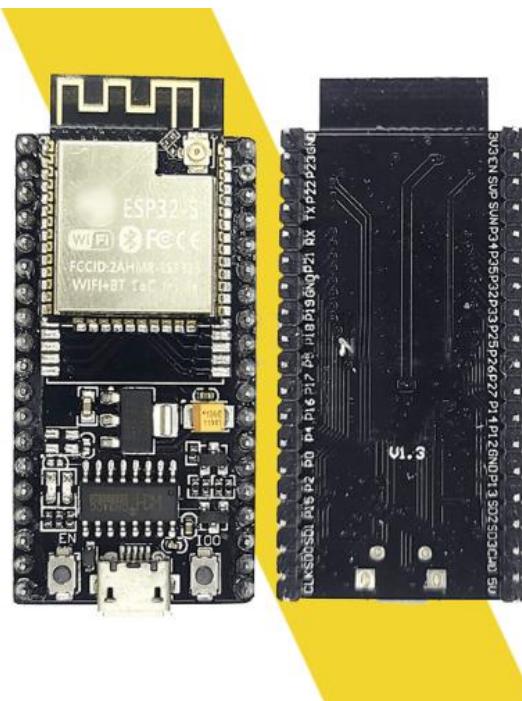
V. Các thiết bị sử dụng trong hệ thống

1. Phần cứng

a. ESP32

ESP32 là một vi điều khiển tích hợp Wi-Fi và Bluetooth được phát triển bởi Espressif Systems. Đây là phiên bản nâng cấp của ESP8266, có hiệu năng cao hơn, tiêu thụ điện năng thấp hơn và hỗ trợ nhiều giao tiếp ngoại vi hơn.

ESP32 được thiết kế dành cho các ứng dụng Internet of Things (IoT), hệ thống tự động hóa, nhà thông minh, đo lường cảm biến và các hệ thống nhúng cần khả năng kết nối không dây.



Chức năng:

- Kết nối Wi-Fi: Cho phép giao tiếp với mạng Internet hoặc mạng nội bộ để gửi và nhận dữ liệu từ máy chủ hoặc nền tảng đám mây.
- Kết nối Bluetooth (Classic và BLE): Giao tiếp không dây với các thiết bị khác như điện thoại thông minh, module hoặc cảm biến BLE.
- Thu thập dữ liệu cảm biến: Nhận tín hiệu từ các cảm biến như cảm biến khí gas, nhiệt độ, độ ẩm, ánh sáng,... thông qua các chân analog hoặc digital.
- Xử lý và truyền dữ liệu: Thực hiện các phép tính, xử lý dữ liệu cảm biến và truyền thông tin lên hệ thống giám sát.

- Điều khiển thiết bị ngoại vi: Gửi tín hiệu điều khiển đến các thiết bị như relay, động cơ, LED, màn hình LCD, servo,...
- Hỗ trợ lập trình linh hoạt: Có thể lập trình bằng nhiều môi trường như Arduino IDE, ESP-IDF hoặc MicroPython.
- Giao tiếp với các module khác: Thông qua các chuẩn giao tiếp phổ biến như UART, SPI, I2C, PWM, ADC, DAC, GPIO,...

Thành phần phần cứng:

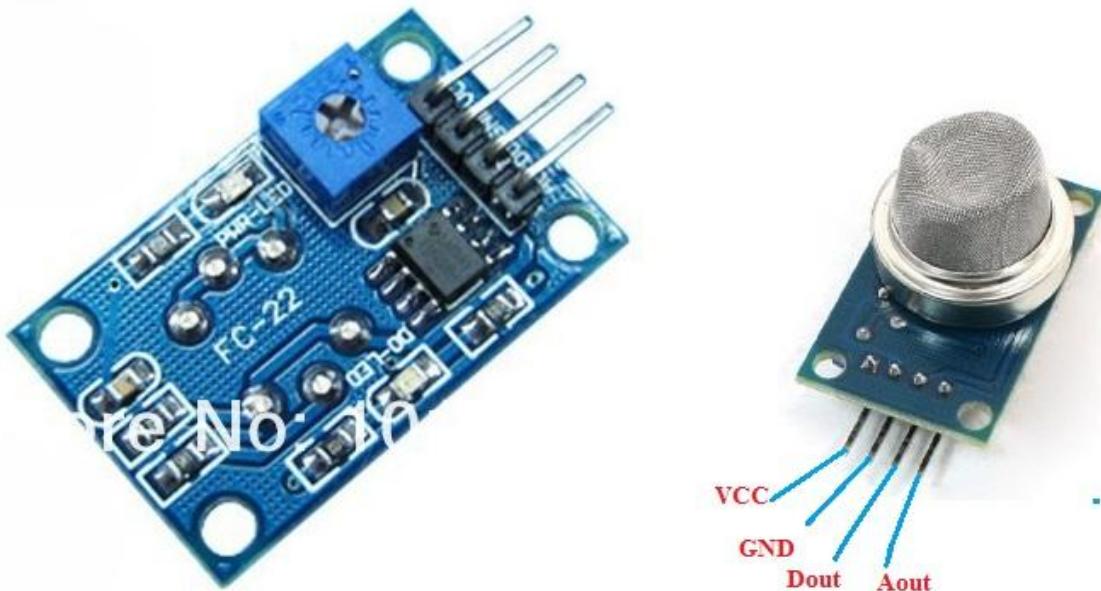
Một bo mạch ESP32 thông thường được cấu tạo từ các thành phần phần cứng chính sau:

- Bộ vi xử lý trung tâm (CPU): Sử dụng lõi kép Tensilica Xtensa LX6, xử lý dữ liệu và điều khiển toàn bộ hoạt động của module.
- Bộ nhớ Flash: Dùng để lưu trữ chương trình (firmware) và dữ liệu người dùng. Dung lượng thường từ 2MB đến 4MB.
- Bộ nhớ RAM: Lưu trữ tạm thời dữ liệu trong quá trình xử lý.
- Mạch nguồn (Power Supply): Cung cấp điện áp ổn định 3.3V cho toàn bộ hệ thống.
- Ăng-ten Wi-Fi/Bluetooth: Tích hợp sẵn hoặc gắn ngoài, giúp truyền nhận tín hiệu không dây.
- Các chân GPIO (General Purpose Input/Output): Dùng để giao tiếp với cảm biến, module hoặc thiết bị ngoại vi.
- Các bộ chuyển đổi tín hiệu:
 - ADC (Analog to Digital Converter): Chuyển tín hiệu analog thành digital để đọc dữ liệu cảm biến.
 - DAC (Digital to Analog Converter): Xuất tín hiệu analog từ dữ liệu digital.
- Công giao tiếp: Bao gồm UART, SPI, I2C, I2S, phục vụ kết nối giữa ESP32 và các module khác.
- Đèn LED trạng thái: Hiển thị tình trạng hoạt động hoặc kết nối của bo mạch.
- Mạch USB-to-Serial: Giúp nạp chương trình và giao tiếp giữa ESP32 với máy tính.

b.Cảm biến khí gas MQ2

Cảm biến MQ-2 là loại **cảm biến khí bán dẫn (gas sensor)** dùng để phát hiện khí dễ cháy hoặc khói trong không khí. Nó hoạt động dựa trên sự **thay đổi điện trở** của vật liệu nhạy khí **SnO₂ (thiếc oxit)** khi có khí gas xuất hiện.MQ-2

thường được sử dụng trong các **hệ thống cảnh báo rò rỉ gas, báo cháy, hoặc các ứng dụng IoT giám sát môi trường**.



Chức năng:

- Có khả năng phát hiện sự có mặt của các loại khí dễ cháy như: khí gas tự nhiên (methane, CH₄), khí LPG (propane, C₃H₈), khói và hơi từ các chất dễ cháy, một số khí độc như CO (carbon monoxide).
- Đo lường nồng độ khí trong không khí: Cảm biến có thể ước lượng được mức nồng độ tương đối của khí trong không khí.
- Cảnh báo rò rỉ khí gas: Trên module MQ-2 có chiết áp (biến trở điều chỉnh) và mạch so sánh điện áp, giúp thiết lập ngưỡng phát hiện (threshold). Khi nồng độ khí vượt quá ngưỡng đã cài đặt, module sẽ xuất ra tín hiệu mức logic thấp (LOW) tại chân D0 (Digital Output). Tín hiệu này có thể được dùng để kích hoạt còi báo, đèn LED, relay hoặc gửi cảnh báo tới bộ điều khiển trung tâm.

Nguyên lý hoạt động:

Cảm biến MQ-2 hoạt động dựa trên sự thay đổi điện trở của vật liệu bán dẫn khi tiếp xúc với các loại khí khác nhau. Thành phần nhạy của cảm biến là thiếc oxit

(SnO₂), một vật liệu có điện trở cao trong không khí sạch. Cảm biến hoạt động theo các bước sau đây:

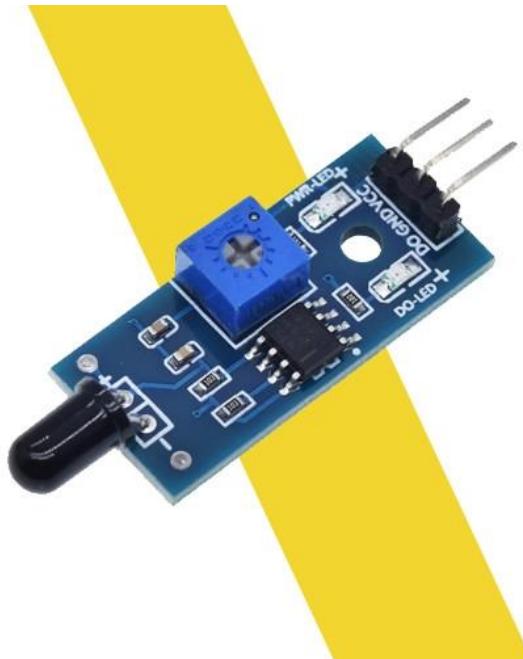
- Ở điều kiện **không khí sạch**, oxy trong không khí được hấp phụ lên bề mặt SnO₂, tạo ra một lớp điện tích làm **điện trở cảm biến rất cao**.
- Khi có các khí dễ cháy hoặc khí khử như LPG, methane, hydrogen hay khói xuất hiện, chúng sẽ phản ứng với oxy hấp phụ trên bề mặt SnO₂. Quá trình này làm giải phóng electron, khiến điện trở của cảm biến giảm xuống. Mức giảm điện trở tỉ lệ thuận với nồng độ của khí trong môi trường.
- Tín hiệu điện áp thay đổi do biến thiên điện trở được đưa ra ở chân analog (A0) để vi điều khiển đọc và xử lý. Ngoài ra, module MQ-2 còn có mạch so sánh ngưỡng, cho phép xuất tín hiệu digital (D0) khi nồng độ khí vượt mức đã cài đặt, giúp hệ thống đưa ra cảnh báo kịp thời.

Thông số kỹ thuật:

- Nguồn hoạt động: 5V DC
- Loại dữ liệu: Analog
- Phạm vi phát hiện: khoảng 300 – 10.000 ppm (tùy loại khí)
- Tốc độ phản hồi nhanh và độ nhạy cao với các khí dễ cháy
- Mạch hoạt động đơn giản, dễ tích hợp với vi điều khiển
- Ồn định trong việc sử dụng lâu dài

c.Cảm biến phát hiện lửa

Cảm biến phát hiện lửa là một loại cảm biến quang điện dùng để nhận biết sự xuất hiện của ngọn lửa trong phạm vi nhất định. Cảm biến này hoạt động dựa trên việc phát hiện tia hồng ngoại (IR) phát ra từ ngọn lửa. Khi có lửa, tín hiệu ánh sáng hồng ngoại sẽ được cảm biến ghi nhận và chuyển đổi thành tín hiệu điện để gửi về bộ xử lý trung tâm như vi điều khiển hoặc ESP32.



Chức năng:

- Phát hiện sự xuất hiện của ngọn lửa trong môi trường giám sát.
- Cảnh báo khi có cháy hoặc nhiệt độ tăng bất thường thông qua tín hiệu xuất ra.
- Gửi tín hiệu đến bộ vi điều khiển để kích hoạt còi báo động, đèn LED, hoặc các thiết bị bảo vệ khác.
- Ứng dụng trong các hệ thống cảnh báo cháy, giám sát an toàn, hoặc điều khiển tự động khi có cháy xảy ra.

Nguyên lý hoạt động:

Cảm biến phát hiện lửa hoạt động dựa trên việc nhận biết tia hồng ngoại (IR) phát ra từ ngọn lửa. Thành phần chính của cảm biến là một diode quang nhạy với bức xạ hồng ngoại có bước sóng khoảng 760 nm – 1100 nm.

Khi có lửa xuất hiện, cảm biến sẽ thu nhận ánh sáng hồng ngoại, làm thay đổi điện áp đầu ra của diode quang. Tín hiệu này được khuếch đại và xử lý bởi mạch điện tích hợp, sau đó xuất ra dạng tín hiệu số (digital) hoặc tương tự (analog) tùy theo module cảm biến.

Trong điều kiện bình thường (không có lửa), đầu ra của cảm biến ở mức logic cao. Khi phát hiện lửa, đầu ra chuyển về mức logic thấp (hoặc ngược lại tùy thiết kế). Bộ vi điều khiển sẽ đọc tín hiệu này để đưa ra phản ứng phù hợp, chẳng hạn như kích hoạt còi báo hoặc gửi cảnh báo qua mạng.

Thông số kỹ thuật:

- Nguồn cấp: 3.3V – 5VDC
- Dòng tiêu thụ: khoảng 15mA
- Tín hiệu đầu ra: Digital 3.3–5VDC (tùy theo nguồn cấp) hoặc Analog
- Khoảng cách phát hiện: tối đa khoảng 80 cm
- Góc quét: khoảng 60 độ
- Bước sóng cảm biến: 760 nm – 1100 nm (tia hồng ngoại)
- Tốc độ phản hồi: nhanh, gần như tức thời khi có ngọn lửa
- Kích thước module: 3.2 x 1.4 cm
- Có đèn LED báo nguồn và LED chỉ thị khi phát hiện lửa
- Hoạt động ổn định trong nhiều điều kiện môi trường

Ngoài ra còn các thành phần khác như:

- Màn hình LCD 1602 để hiển thị thông số.
- Module 2 relay để điều khiển bật tắt quạt, máy bơm.
- Servo để đóng mở cửa.
- Kit để kết nối các linh kiện.
- Dây dẫn.

2. Phần mềm

a. Arduino IDE

Arduino IDE được sử dụng để viết mã điều khiển cho vi điều khiển ESP32. Sau khi chương trình được viết và kiểm tra, mã nguồn sẽ được nạp trực tiếp vào ESP32 thông qua cổng giao tiếp USB. Arduino IDE cung cấp các công cụ hỗ trợ kiểm tra lỗi cú pháp, biên dịch và nạp chương trình, giúp đảm bảo quá trình lập trình diễn ra chính xác và ổn định.

b. Frontend (ReactJS)

ReactJS được sử dụng để xây dựng giao diện người dùng của hệ thống. Giao diện hiển thị nồng độ khí gas, trạng thái cảnh báo cháy, tình trạng của các thiết bị điều khiển (relay, servo), và cho phép người dùng thao tác điều khiển trực tiếp. ReactJS giúp tạo nên trải nghiệm tương tác nhanh, mượt mà và thân thiện với người dùng.

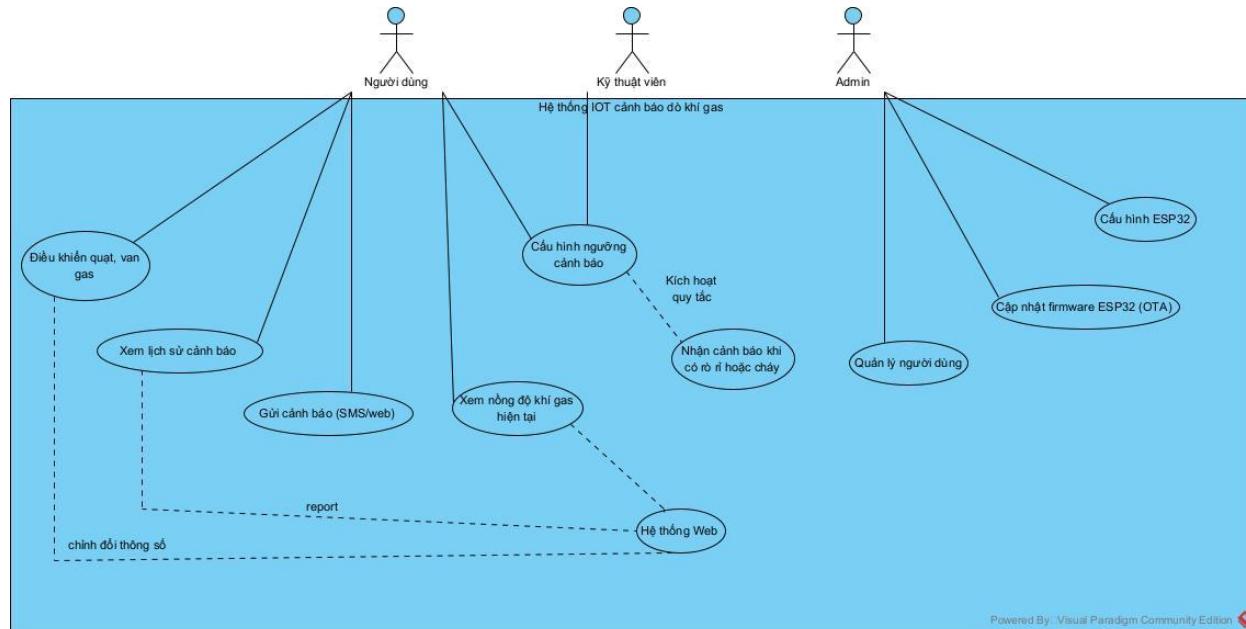
c. Backend (Java Spring Boot)

Spring Boot được sử dụng để xây dựng backend của hệ thống, chịu trách nhiệm xử lý logic nghiệp vụ và giao tiếp giữa frontend với Firebase Realtime Database. Ứng dụng backend cung cấp các API RESTful để nhận và gửi dữ liệu, đồng thời đảm bảo tính bảo mật, ổn định và khả năng mở rộng cho toàn hệ thống.

d. Firebase Realtime Database

Firebase Realtime Database là cơ sở dữ liệu thời gian thực được lưu trữ trên nền tảng đám mây của Google. Cơ sở dữ liệu này được sử dụng để lưu trữ các giá trị đo được từ cảm biến khí gas, cảm biến lửa, ngưỡng giá trị chuẩn, cũng như trạng thái hoạt động của relay và servo. Dữ liệu được đồng bộ hóa theo thời gian thực giữa ESP32 và ứng dụng web, giúp người dùng có thể giám sát và điều khiển thiết bị mọi lúc, mọi nơi.

VI. Use Case Diagram



1. Use Case: Cấu hình ngưỡng cảnh báo

- Mô tả:** Cho phép người dùng (có thể là kỹ thuật viên) thiết lập các ngưỡng giá trị cho cảm biến khí gas. Khi vượt quá ngưỡng này, hệ thống sẽ kích hoạt cảnh báo.
- Luồng kích hoạt:** Người dùng thực hiện thiết lập thông qua giao diện web hoặc ứng dụng.
- Điều kiện kích hoạt:**

- Nhận cảnh báo khi có rò rỉ hoặc cháy.
 - **Tác nhân chính:** Kỹ thuật viên
- 2. Use Case: Xem lịch sử cảnh báo**
- **Mô tả:** Người dùng có thể xem lại lịch sử các cảnh báo đã xảy ra trong quá khứ.
 - **Luồng kích hoạt:** Truy cập vào mục lịch sử cảnh báo trên web hoặc ứng dụng.
 - **Tác nhân chính:** Người dùng

3. Use Case: Giải cảnh báo (SMS/web)

- **Mô tả:** Hệ thống gửi cảnh báo đến người dùng qua SMS hoặc giao diện web khi phát hiện sự cố. Người dùng có thể xác nhận đã nhận và xử lý cảnh báo.
- **Luồng kích hoạt:** Cảnh biến phát hiện vượt ngưỡng cho phép.
- **Tác nhân chính:** Hệ thống

4. Use Case: Xem nồng độ khí gas hiện tại

- **Mô tả:** Người dùng có thể theo dõi nồng độ khí gas theo thời gian thực thông qua giao diện web hoặc ứng dụng.
- **Luồng kích hoạt:** Truy cập vào màn hình giám sát.
- **Tác nhân chính:** Người dùng

5. Use Case: Cấu hình ESP32

- **Mô tả:** Cho phép kỹ thuật viên cấu hình thiết bị phần cứng ESP32, bao gồm cài đặt thông số kết nối, ngưỡng cảm biến, v.v.
- **Tác nhân chính:** Admin

6. Use Case: Quản lý người dùng

- **Mô tả:** Cho phép thêm, xóa, sửa thông tin người dùng trong hệ thống.
- **Tác nhân chính:** Admin

7. Use Case: Cập nhật firmware ESP32 (OTA)

- **Mô tả:** Hệ thống hỗ trợ cập nhật firmware từ xa (Over-The-Air) cho thiết bị ESP32 mà không cần can thiệp vật lý.

- **Tác nhân chính:** Admin

8. Use Case: Báo cáo / Thống kê

- **Mô tả:** Hệ thống tổng hợp và hiển thị báo cáo các thông số quan trọng như số lần cảnh báo, mức độ rò rỉ, trạng thái thiết bị,,
- **Giao diện:** Hệ thống Web
- **Tác nhân chính:** Người dùng