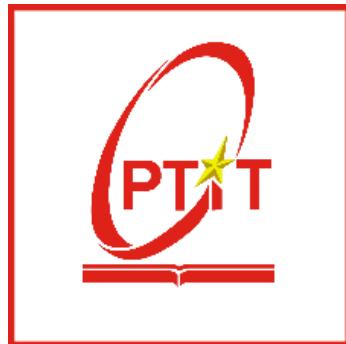


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN



**MÔN: IOT VÀ ỨNG DỤNG
BÁO CÁO CUỐI 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: 17

Thành viên:

Ngô Đắc Tuấn Kiệt B22DCCN435

Ngô Tiến Hưng B22DCCN411

Ma Văn Hùng B22DCCN363

Võ Văn Hoàng B22DCCN351

HÀ NỘI

MỤC LỤC

1. Giới thiệu chung	6
1.1 Lý do chọn đề tài	6
1.1.1 Tâm quan trọng của đề tài	6
1.1.2 Tính ứng dụng của đề tài	6
1.2 Tổng quan dự án	7
1.3 Mục đích dự án	7
1.3.1 Mục tiêu tổng quát:.....	7
1.3.2 Mục tiêu cụ thể:	7
1.4 Đối tượng và phạm vi nghiên cứu	7
1.5 Phương pháp nghiên cứu	8
1.6 Các công nghệ áp dụng.....	8
2. Nền tảng lý thuyết.....	9
2.1 Tổng quan công nghệ IoT	9
2.1.1 Kiến trúc IoT 5 tầng.....	9
2.1.2 Ứng dụng IoT trong an toàn cháy nổ.....	9
2.2 Cảm biến khí gas MQ-2	9
2.2.1 Giới thiệu tổng quan	9
2.2.2 Cấu tạo bên trong MQ-2.....	10
2.2.3 Nguyên lý hoạt động	10
2.2.4 Thông số kỹ thuật	11
2.2.5 Sơ đồ chân (Pinout)	11
2.2.6 Sơ đồ kết nối MQ-2 với ESP32.....	12
2.2.7 Hiệu chuẩn cảm biến	12
2.3 Cảm biến phát hiện lửa(Flame Sensor)	12
2.3.1 Giới thiệu tổng quan	12
2.3.2 Linh kiện cảm biến chính — Phototransistor YG1006	13
2.3.3 Cấu tạo module Flame Sensor.....	14
2.3.4 Nguyên lý hoạt động chi tiết	14
2.3.5 Phổ bức xạ hồng ngoại của ngọn lửa.....	15
2.3.6 Thông số kỹ thuật chi tiết	15

2.3.7 Sơ đồ chân (Pinout)	16
2.3.8 Sơ đồ kết nối Flame Sensor với ESP32.....	16
2.3.9 Điều chỉnh độ nhạy.....	17
2.3.10 Vai trò trong hệ thống cảnh báo rò rỉ khí gas:	17
2.4 Vi điều khiển ESP32(ESP – WROOM-32S)	17
2.4.1 Giới thiệu tổng quan	17
2.4.2 Kiến trúc phần cứng	18
2.4.3 Sơ đồ chân (Pinout) & Giao tiếp	20
2.4.4 Nguyên lý hoạt động	21
2.4.5 Lập trình thời gian thực (FreeRTOS)	21
2.4.6 Vai trò trong hệ thống	21
2.5 Modul Relay Mini 2 kênh 5V 10A.....	21
2.5.1 Nguyên lý hoạt động	22
2.5.2 Thông số kỹ thuật	24
2.5.3 Ứng dụng trong hệ thống cảnh báo rò rỉ khí gas	25
2.6 Động cơ SERVO SG90	26
2.6.1 Giới thiệu tổng quan	26
2.6.2 Nguyên lý hoạt động	26
2.6.3 Cấu tạo	27
2.6.4 Ứng dụng trong hệ thống cảnh báo rò rỉ khí gas	28
2.7 Công nghệ điện toán đám mây (Firebase).....	29
2.7.1 Giới thiệu	29
2.7.2 Các dịch vụ Firebase sử dụng trong dự án	29
2.7.3 Firebase Realtime Database	30
2.7.4 Firebase Cloud Messaging (FCM)	30
2.7.5 Authentication & Security Rules	30
2.8 Công nghệ phần mềm	30
2.8.1 Tổng quan công nghệ	30
2.8.2 Giao thức truyền thông	31
2.8.3 WiFi (IEEE 802.11)	31

2.8.4 HTTP/HTTPS (HyperText Transfer Protocol)	31
2.8.5 WebSocket	31
3. Phân tích yêu cầu chức năng	32
3.1 Thu thập dữ liệu nồng độ khí gas	32
3.2 Phân tích và xử lý sự kiện	33
3.3 Gửi dữ liệu và cảnh báo từ xa.....	33
3.4 Điều khiển thiết bị an toàn (Actuators)	34
3.5 Tương tác với người dùng	34
4. Phân tích thiết kế	34
4.1 Đặc tả mô hình miền.....	34
4.2 Đặc tả các thành phần chức năng và hoạt động.....	38
4.2.1 Thành phần chức năng.....	39
4.2.2 Thành phần hoạt động	41
4.3 Đặc tả luồng hoạt động.....	42
4.3.1 Thiết bị cảm biến và vi xử lý	42
4.3.2 Website quản lý.....	43
4.4 Tích hợp thiết bị.....	44
4.5 Thiết kế tổng thể hệ thống	45
4.5.1 Kiến trúc hệ thống	45
4.5.2 Sơ đồ khối tổng thể.....	46
4.6 Thiết kế hệ thống IoT	46
4.6.1 Usecase tổng quan	47
4.6.2 Data Flow Diagram	48
5. Kết quả đạt được	48
5.1 Về mặt lý thuyết	48
5.2 Về mặt thực tiễn.....	48
6. Kết luận và hướng mở rộng	49
6.1 Về hệ thống.....	49
6.2 Về bảo mật và sản phẩm hóa	49

DANH SÁCH BẢNG

Bảng 2.1: Sơ đồ chân module MQ-2	11
Bảng 2.2: Sơ đồ đấu nối MQ-2 với ESP32	12
Bảng 2.3: Sơ đồ chân module Flame Sensor 4 pin	16
Bảng 2.4: Sơ đồ đấu nối chi tiết Flame Sensor với ESP32.....	17
Bảng 2.5: Sơ đồ chân modul relay 2 kênh.....	25
Bảng 2.6: Sơ đồ kết nối với ESP32 của modul relay	25
Bảng 2.7: Sơ đồ chân của Servo SG90	28
Bảng 2.8: Sơ đồ kết nối Servo SG90 với ESP32	29
Bảng 4.1: Các hàm chức năng	42
Bảng 4.2: Các API sử dụng	44
Bảng 4.3: Bảng nối chân thiết bị với ESP32.....	45

DANH SÁCH HÌNH ẢNH

Hình 2.1: Cảm biến MQ-2	10
Hình 2.2: Hình modul Flame Sensor 4 chân	13
Hình 2.3: Ảnh phototransistor YG1006	13
Hình 2.4: Biểu đồ phô bức xạ hồng ngoại và vùng phát hiện lửa.....	15
Hình 2.5: Hình ảnh module ESP32-WROOM-32	18
Hình 2.6: Sơ đồ khối kiến trúc ESP32	18
Hình 2.7: Sơ đồ chân ESP32 chi tiết	20
Hình 2.8: Hình ảnh Module Relay Mini 2 Kênh 5V 10A	22
Hình 2.9: Sơ đồ pinout Module Relay 2 kênh.....	23
Hình 2.10: Hình ảnh Servo SG90	26
Hình 2.11: Hình ảnh cấu tạo bên trong Servo SG90	27
Hình 2.12: Hình ảnh kết nối Servo SG90 với ESP32.....	28
Hình 4.1: Đặc tả mô hình miền.....	35
Hình 4.2: Các thành phần chức năng và hoạt động.....	39
Hình 4.3: Sơ đồ nối chân	44
Hình 4.4: Sơ đồ khái tổng thể.....	46
Hình 4.5: Usecase tổng quan.....	47
Hình 4.6: Data Flow Diagram	48

1. Giới thiệu chung

1.1 Lý do chọn đề tài

1.1.1 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.

1.1.2 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

1.2 Tổng quan dự án

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. Hệ thống bao gồm các cảm biến đặt tại các khu vực có nguy cơ rò rỉ gas, cháy nổ cao và gửi dữ liệu về hệ thống quản cho phép người dùng theo dõi, điều khiển thông qua giao diện của hệ thống.

1.3 Mục đích dự án

1.3.1 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 và thông báo qua Internet.
- Điều khiển tự động các thiết bị an toàn như quạt thông gió, cửa tự động 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).

1.3.2 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
- Phát triển backend trung gian
- Đảm bảo khả năng mở rộng và tính thực tiễn
- Nâng cao tính an toàn và ổn định của hệ thống

1.4 Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu:
 - Các loại cảm biến khí Gas (MQ-2), cảm biến phát hiện lửa
 - Bộ vi điều khiển ESP32
 - Mạch relay, còi báo động, màn hình hiển thị
 - Hệ thống quản lý và điều khiển qua website sử dụng dịch vụ Cloud Firebase
- Phạm vi nghiên cứu:
 - Lắp đặt cảm biến tại các vị trí cần cảnh báo, thu thập và truyền dữ liệu nồng độ khí Gas về hệ thống chính.

- Xử lý dữ liệu, đối chiếu với ngưỡng an toàn để phát hiện và cảnh báo rò rỉ khí Gas.
- Kích hoạt cảnh báo đa phương tiện (âm thanh, ánh sáng, thông báo đến người dùng).
- Điều khiển thiết bị chấp hành như relay mở cửa thoát hiểm, còi báo động.
- Quản lý, giám sát từ xa qua giao diện web, lưu trữ và phân tích dữ liệu trên cloud.
- Triển khai thử nghiệm trong môi trường sử dụng thực tế: hộ gia đình, quán ăn, xưởng sản xuất,...

1.5 Phương pháp nghiên cứu

- Phương pháp lý thuyết: Tìm hiểu về các công nghệ IoT liên quan (cấu trúc vi điều khiển ESP32, cảm biến khí Gas, Cloud Firebase, giao tiếp không dây), học hỏi kinh nghiệm từ các hệ thống cảnh báo hiện có.
- Phương pháp thực nghiệm: Thiết kế, lắp ráp và cấu hình hệ thống thực tế gồm phần cứng và phần mềm. Thực hiện kiểm thử hệ thống dưới các điều kiện môi trường khác nhau để đánh giá độ chính xác, độ tin cậy trong việc phát hiện rò rỉ khí Gas và hoạt động cảnh báo, điều khiển thiết bị.
- Phương pháp phân tích: Sử dụng công cụ phân tích xử lý dữ liệu thu thập được để đo lường hiệu quả hệ thống, đối chiếu kết quả thực nghiệm với lý thuyết, điều chỉnh phần cứng/phần mềm để hoàn thiện sản phẩm.

1.6 Các công nghệ áp dụng

- Công nghệ IoT:
 - ESP32
 - Cảm biến khí gas MQ-2
 - Cảm biến phát hiện lửa (Flame Sensor)
- Các thiết bị chấp hành (Actuators) và ngoại vi:
 - Module 2 relay
 - Servo SG90
 - Màn hình LCD 1602 và module I2C
 - Còi buzzer
- Công nghệ điện toán đám mây:
 - Firebase Realtime Database
 - Firebase Cloud Messaging (FCM)
 - Firebase Authentication & Security Rules
- Các công nghệ phần mềm:
 - Arduino IDE
 - ReactJS (Frontend)
 - Java Spring Boot (Backend)
 - Giao thức truyền thông: HTTP, RestAPI, WebSocket

2. Nền tảng lý thuyết

2.1 Tổng quan công nghệ IoT

IoT là mạng lưới các thiết bị vật lý được kết nối internet, có khả năng thu thập, trao đổi và xử lý dữ liệu tự động mà không cần sự can thiệp của con người.

2.1.1 Kiến trúc IoT 5 tầng

- Tầng Perception: Thu thập dữ liệu
- Tầng Processing: Xử lý logic, ra quyết định
- Tầng Network: Truyền dữ liệu
- Tầng Cloud: Lưu trữ, đồng bộ
- Tầng Application: Hiển thị, điều khiển

2.1.2 Ứng dụng IoT trong an toàn cháy nổ

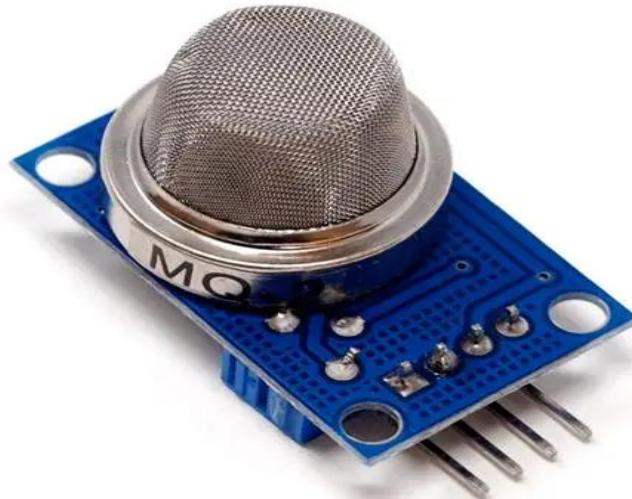
- Giám sát: Cảm biến đo nồng độ khí gas 24/7
- Phát hiện: Nhận diện khí gas vượt ngưỡng hoặc có lửa
- Cảnh báo: Gửi notification, bật còi tức thì
- Xử lý tự động: Bật quạt, mở cửa thoát hiểm

2.2 Cảm biến khí gas MQ-2

2.2.1 Giới thiệu tổng quan

Cảm biến MQ-2 là loại cảm biến bán dẫn phổ biến nhất để phát hiện khí gas cháy và khói trong các hệ thống IoT cảnh báo rò rỉ khí gas. Cảm biến có thể phát hiện nhiều loại khí như LPG, propane, methane, butane, hydrogen, alcohol và khói với dải đo từ 300–10.000 ppm.

2.2.2 Cấu tạo bên trong MQ-2



Hình 2.1: Cảm biến MQ-2

Cảm biến MQ-2 bao gồm các thành phần chính:

- Ông gómc micro Al_2O_3 : Làm nền cho lớp vật liệu cảm biến.
- Lớp cảm biến SnO_2 (Tin Dioxide): Vật liệu bán dẫn nhạy với khí gas, thay đổi điện trở khi tiếp xúc với khí cháy.
- Cuộn dây gia nhiệt (Heater Coil): Làm nóng lớp cảm biến đến nhiệt độ hoạt động ($\sim 300^\circ\text{C}$) để phản ứng hóa học xảy ra chính xác.
- Điện cực đo: Thu thập tín hiệu điện trở thay đổi.
- Lưới chống nổ (Anti-explosion mesh): Hai lớp lưới thép không gỉ mịn bao quanh, ngăn ngọn lửa từ cuộn gia nhiệt gây cháy nổ khi có khí gas.

2.2.3 Nguyên lý hoạt động

Cảm biến MQ-2 hoạt động dựa trên nguyên lý thay đổi điện trở (Chemisorption):

Trạng thái bình thường (không khí sạch):

- Khi cấp nguồn, cuộn dây gia nhiệt làm nóng lớp SnO_2 .
- Các phân tử oxy (O_2) trong không khí hấp phụ lên bề mặt SnO_2 , kéo electron tự do ra khỏi lớp bán dẫn.
- Tạo ra "vùng nghèo electron" (electron depletion region), làm điện trở của cảm biến tăng cao.

Khi có khí gas cháy:

- Khí gas (LPG, methane, propane...) phản ứng với các phân tử oxy trên bề mặt SnO_2 .
- Phản ứng này giải phóng electron trở lại lớp bán dẫn.
- Điện trở giảm \rightarrow dòng điện tăng \rightarrow tín hiệu đầu ra thay đổi.

Công thức cơ bản:

Nồng độ khí tăng \Rightarrow Điện trở giảm \Rightarrow Điện áp đầu ra tăng

2.2.4 Thông số kỹ thuật

- Điện áp hoạt động (VCC): 5V DC
- Điện áp gia nhiệt (VH): $5V \pm 0.1V$
- Công suất tiêu thụ: $\leq 800mW$
- Dải phát hiện: 300–10.000 ppm (khí cháy)
- Điện trở cảm biến (R_s): $2k\Omega - 20k\Omega$ (tại 2000 ppm C_3H_8)
- Điện trở gia nhiệt (RH): $31\Omega \pm 3\Omega$
- Thời gian preheat: $> 24-48$ giờ (lần đầu)
- Nhiệt độ/độ ẩm hoạt động: $20^{\circ}C \pm 2^{\circ}C, 65\% \pm 5\% RH$

2.2.5 Sơ đồ chân (Pinout)

Chân	Tên	Chức năng
1	VCC	Nguồn cấp 5V DC
2	GND	Mass (0V)
3	DO	Digital Output — HIGH/LOW theo ngưỡng
4	AO	Analog Output — điện áp tỷ lệ với nồng độ khí

Bảng 2.1: Sơ đồ chân module MQ-2

- Chân AO cho tín hiệu analog tỷ lệ thuận với nồng độ khí (nồng độ tăng \rightarrow điện áp tăng).
- Chân DO cho tín hiệu số (LOW khi vượt ngưỡng, HIGH khi dưới ngưỡng), có thể điều chỉnh ngưỡng bằng biến trở (potentiometer) trên module.

2.2.6 Sơ đồ kết nối MQ-2 với ESP32

MQ-2	ESP32
VCC	VIN (5V)
GND	GND
AO	GPIO 34 (hoặc chân ADC khác)
DO	GPIO bất kỳ (tùy chọn)

Bảng 2.2: Sơ đồ đấu nối MQ-2 với ESP32

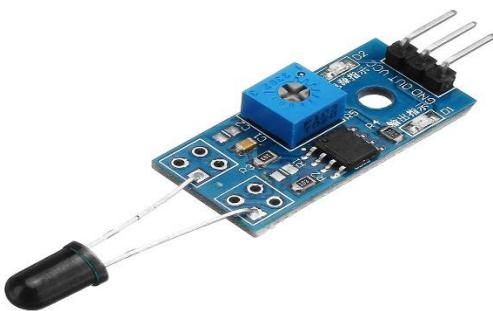
2.2.7 Hiệu chuẩn cảm biến

- Preheat (làm nóng trước): Lần đầu sử dụng cần cấp nguồn liên tục 24–48 giờ để ổn định.
- Điều chỉnh ngưỡng: Xoay biến trở trên module để thiết lập ngưỡng phát hiện phù hợp với môi trường.
- Baseline: Đo giá trị trong không khí sạch làm mốc so sánh.

2.3 Cảm biến phát hiện lửa(Flame Sensor)

2.3.1 Giới thiệu tổng quan

Cảm biến phát hiện lửa (Flame Sensor) là thiết bị điện tử chuyên dụng để nhận diện ngọn lửa hoặc nguồn sáng phát ra bức xạ hồng ngoại (IR). Trong hệ thống cảnh báo rò rỉ khí gas, flame sensor đóng vai trò bổ trợ quan trọng cho cảm biến MQ-2, giúp phát hiện nguy cơ cháy nổ từ hai hướng: rò rỉ khí (MQ-2) và ngọn lửa thực sự (Flame Sensor).



Hình 2.2: Hình modul Flame Sensor 4 chân

2.3.2 Linh kiện cảm biến chính — Phototransistor YG1006



Hình 2.3: Ảnh phototransistor YG1006

YG1006 là linh kiện cốt lõi của module Flame Sensor:

- Loại: NPN Silicon Phototransistor
- Kích thước: 5mm (tiêu chuẩn)
- Vỏ bọc: Epoxy đen — chỉ nhạy với ánh sáng hồng ngoại
- Dải bước sóng: 760 nm – 1100 nm
- Bước sóng nhạy nhất: 940 nm
- Góc phát hiện: 60°
- Tốc độ đáp ứng: ~15 µs (cực nhanh)
- Độ nhạy: Cao, phù hợp phát hiện lửa

Nguyên lý:

- Vỏ epoxy đen lọc bỏ ánh sáng nhìn thấy, chỉ cho phép bức xạ hồng ngoại (IR) đi qua.

- Khi có IR từ ngọn lửa chiếu vào → dòng điện chạy qua transistor tăng → điện trở giảm.

2.3.3 Cấu tạo module Flame Sensor

Module Flame Sensor bao gồm các thành phần:

- Phototransistor YG1006: Nhận bức xạ IR từ ngọn lửa
- IC so sánh LM393: So sánh tín hiệu với ngưỡng, xuất tín hiệu digital
- Biến trở (Potentiometer): Điều chỉnh độ nhạy/ngưỡng phát hiện
- Điện trở phân áp: Tạo mạch phân áp, bảo vệ mạch
- Tụ điện lọc: Lọc nhiễu tín hiệu
- LED đỏ (PWR): Báo nguồn hoạt động
- LED xanh (DO): Báo trạng thái phát hiện lửa
- 4 chân kết nối: VCC, GND, DO, AO

2.3.4 Nguyên lý hoạt động chi tiết

Phát hiện bức xạ hồng ngoại:

- Khi ngọn lửa cháy, nó phát ra bức xạ hồng ngoại trong dải 760 nm – 1100 nm.
- Phototransistor YG1006 nhận bức xạ này.

Thay đổi điện trở:

- Khi không có lửa: Phototransistor có điện trở cao → điện áp tại đầu vào đảo (Inverting) của LM393 cao.
- Khi có lửa: Bức xạ IR làm điện trở của phototransistor giảm → điện áp tại đầu vào đảo thấp.

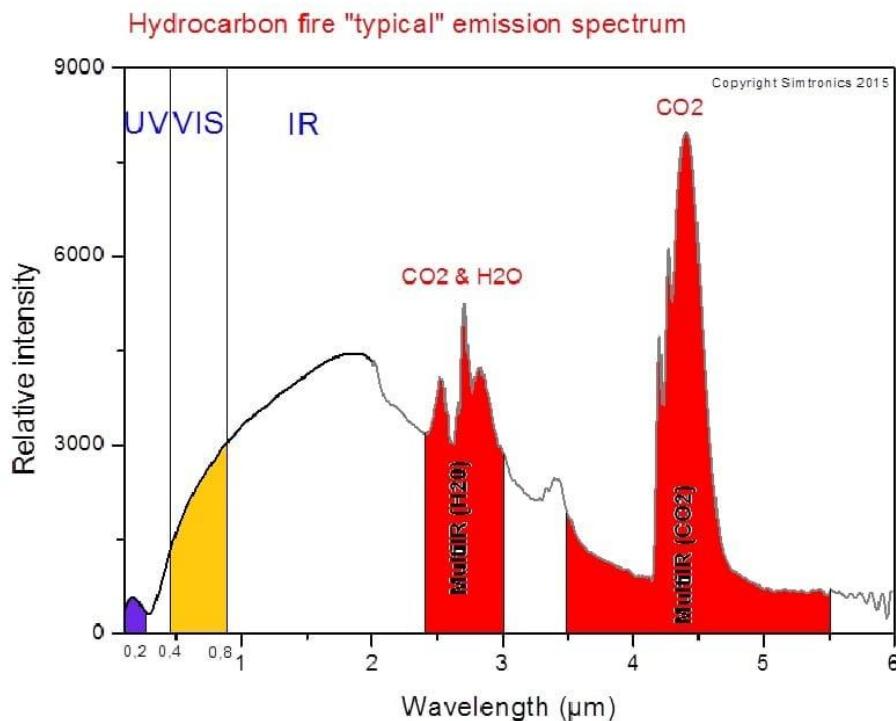
So sánh và xuất tín hiệu:

- IC LM393 so sánh điện áp từ phototransistor với ngưỡng (threshold) được thiết lập bằng biến trở.
- Nếu điện áp đầu vào < ngưỡng (có lửa) → DO = LOW (0V) + LED xanh sáng.
- Nếu điện áp đầu vào > ngưỡng (không lửa) → DO = HIGH (5V).

Công thức logic:

$$\text{Có lửa} \Rightarrow \text{IR tăng} \Rightarrow \text{YG1006 giảm} \Rightarrow V_{in} < V_{ngưỡng} \Rightarrow \text{DO} = \text{LOW}$$

2.3.5 Phổ bức xạ hồng ngoại của ngọn lửa



Hình 2.4: Biểu đồ phổ bức xạ hồng ngoại và vùng phát hiện lửa

Ngọn lửa phát ra bức xạ điện từ ở nhiều bước sóng, trong đó:

- Vùng UV (Ultraviolet): 185–260 nm — dùng cho cảm biến UV flame detector.
- Vùng IR (Infrared): 760–1100 nm — vùng mà YG1006 phát hiện.
- Bước sóng đỉnh: ~940 nm là bước sóng mà YG1006 nhạy nhất.

Chọn vùng IR:

- Ngọn lửa luôn phát ra bức xạ IR mạnh.
- Vỏ epoxy đen của YG1006 lọc bỏ ánh sáng nhìn thấy, giảm báo động giả.
- Chi phí thấp, dễ tích hợp.

2.3.6 Thông số kỹ thuật chi tiết

- Điện áp hoạt động: 3.3V – 5V DC
- Dòng tiêu thụ: ~20 mA
- Dải bước sóng phát hiện: 760 nm – 1100 nm
- Bước sóng nhạy nhất: 940 nm
- Góc phát hiện: 0° – 60°
- Khoảng cách phát hiện: 20 cm – 100 cm (tùy cường độ lửa)
- Thời gian đáp ứng: ~15 μs
- Đầu ra: Digital (DO) + Analog (AO)
- IC so sánh: LM393

- Phototransistor: YG1006
- Nhiệt độ hoạt động: -25°C đến 85°C
- Kích thước PCB: ~32 x 14 mm

2.3.7 Sơ đồ chân (Pinout)

Chân	Tên	Chức năng
1	VCC	Nguồn cấp 3.3V – 5V DC
2	GND	Mass (0V)
3	DO	Digital Output — LOW khi phát hiện lửa, HIGH khi không có lửa
4	AO	Analog Output — điện áp tỷ lệ nghịch với cường độ IR (lửa mạnh → điện áp thấp)

Bảng 2.3: Sơ đồ chân module Flame Sensor 4 pin

LED chỉ thị trên module:

- LED đỏ (PWR): Sáng khi có nguồn.
- LED xanh (DO): Sáng khi phát hiện lửa.

2.3.8 Sơ đồ kết nối Flame Sensor với ESP32

Flame Sensor	ESP32
VCC	3.3V
GND	GND

DO	GPIO 5 (hoặc GPIO digital bất kỳ)
AO	GPIO 34 (chân ADC) — tùy chọn

Bảng 2.4: Sơ đồ đấu nối chi tiết Flame Sensor với ESP32

Lưu ý kết nối:

- ESP32 hoạt động ở mức logic 3.3V, flame sensor tương thích.
- Chân AO dùng để đọc giá trị analog chi tiết hơn.
- Chân DO dùng cho logic đơn giản (có/không có lửa).

2.3.9 Điều chỉnh độ nhạy

Module Flame Sensor có biến trờ (potentiometer) để điều chỉnh ngưỡng phát hiện:

- Theo chiều kim đồng hồ: Tăng độ nhạy, phát hiện lửa xa hơn, dễ báo động giả.
- Ngược chiều kim đồng hồ: Giảm độ nhạy, chỉ phát hiện lửa gần, ít báo động giả.

Cách hiệu chuẩn:

1. Đặt cảm biến trong môi trường bình thường (không có lửa).
2. Xoay biến trờ cho đến khi LED xanh tắt.
3. Đưa nguồn lửa nhỏ (bật lửa) vào vùng phát hiện, xoay biến trờ cho đến khi LED xanh sáng ở khoảng cách mong muốn.

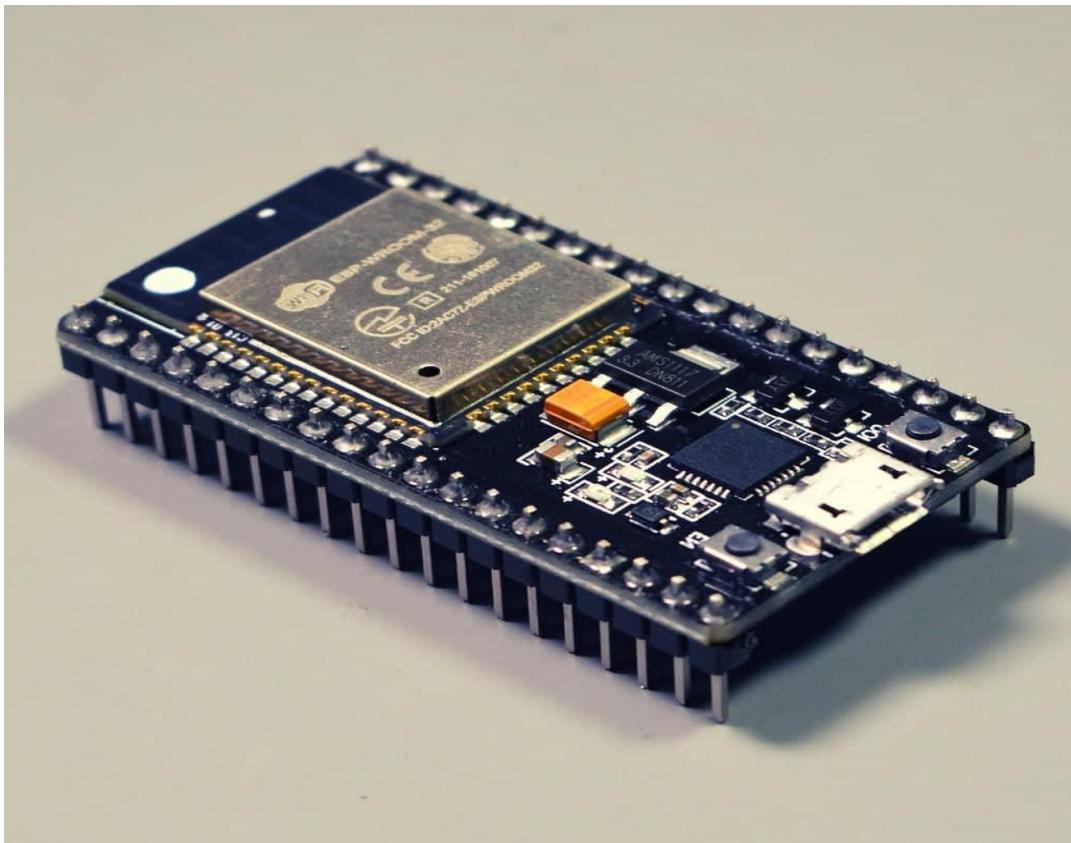
2.3.10 Vai trò trong hệ thống cảnh báo rò rỉ khí gas:

- Phát hiện sớm: MQ-2 cảnh báo khi có rò rỉ khí, trước khi xảy ra cháy.
- Xác nhận nguy hiểm: Flame Sensor xác nhận khi đã có ngọn lửa thực sự.
- Giảm báo động giả: Hai cảm biến xác minh chéo, tăng độ tin cậy.
- An toàn toàn diện: Bảo vệ từ giai đoạn rò rỉ đến giai đoạn cháy thực sự.

2.4 Vi điều khiển ESP32(ESP – WROOM-32S)

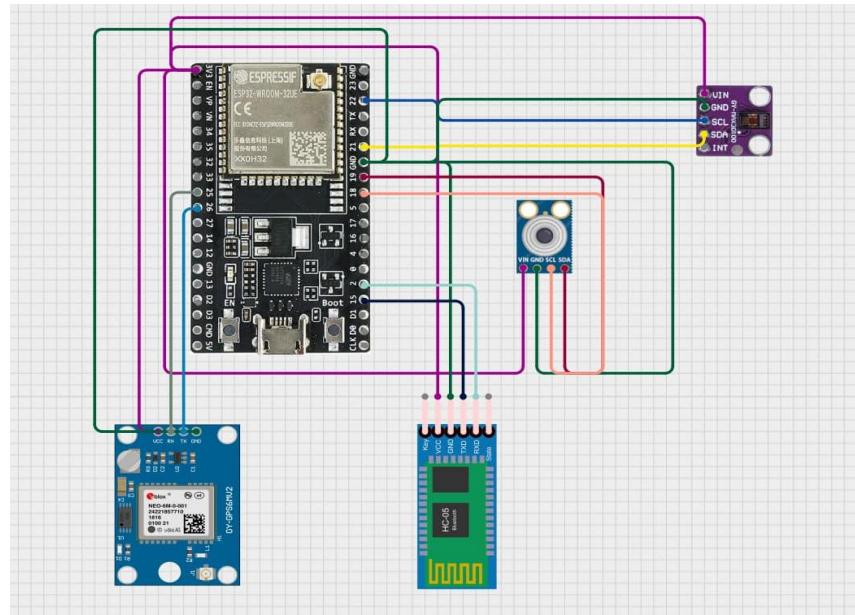
2.4.1 Giới thiệu tổng quan

ESP32-WROOM-32S là module vi điều khiển mạnh mẽ do Espressif Systems phát triển, tích hợp đầy đủ Wi-Fi, Bluetooth và khả năng xử lý cao. Đây là "trái tim" của các dự án IoT hiện đại, đóng vai trò trung tâm xử lý tín hiệu trong hệ thống cảnh báo rò rỉ khí gas.



Hình 2.5: Hình ảnh module ESP32-WROOM-32

2.4.2 Kiến trúc phần cứng



Hình 2.6: Sơ đồ khái niệm trục ESP32

2.4.2.1 CPU Dual-Core Xtensa 32-bit:

- Kiến trúc: Xtensa® LX6 32-bit
- Số nhân: 2 nhân (Dual-Core)
- Xung nhịp: 80 MHz – 240 MHz
- Hiệu năng: Lên đến 600 DMIPS

Ưu điểm của Dual-Core

- Core 0: Xử lý giao thức mạng Wi-Fi/Bluetooth.
- Core 1: Chạy ứng dụng người dùng (đọc cảm biến, điều khiển thiết bị).
- Cho phép xử lý song song, tăng hiệu năng hệ thống.

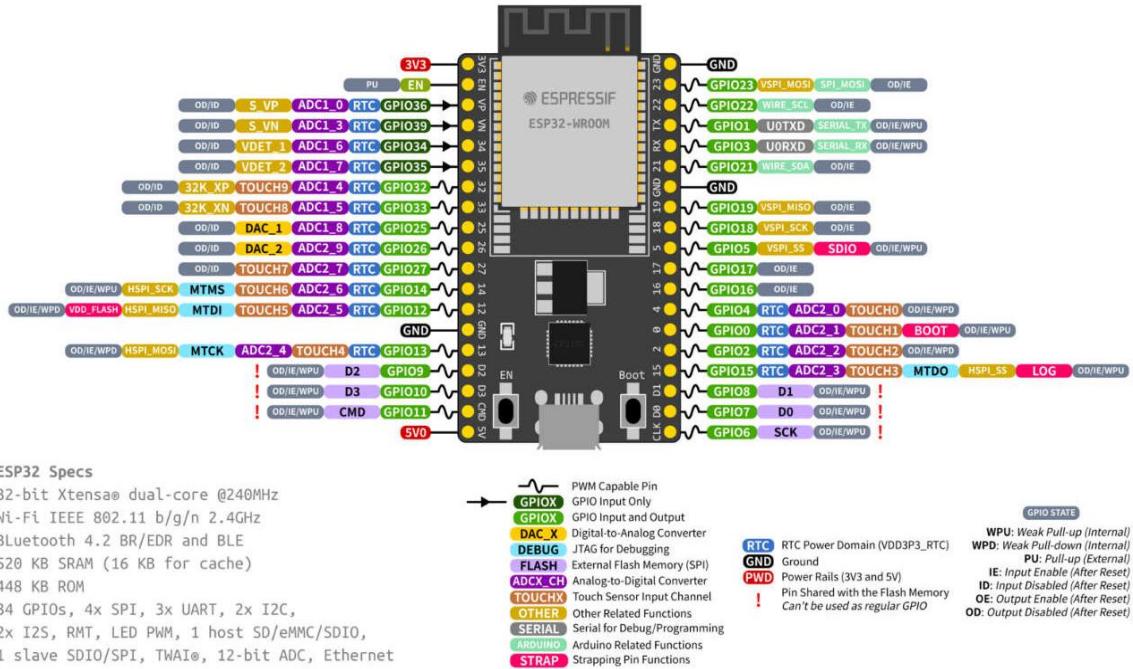
2.4.2.2 Bộ nhớ (Memory)

- SRAM: 520 KB — Bộ nhớ RAM cho xử lý dữ liệu
- ROM: 448 KB — Lưu bootloader, firmware hệ thống
- Flash: 4 MB (mở rộng được) — Lưu chương trình, dữ liệu
- RTC SRAM: 16 KB — Lưu dữ liệu khi deep sleep
- eFuse: 1 Kbit — Lưu thông tin bảo mật

2.4.2.3 Kết nối không dây

- Wi-Fi: 802.11 b/g/n (2.4 GHz)
- Bluetooth: v4.2 BR/EDR và BLE (Bluetooth Low Energy)
- Anten: Tích hợp sẵn trên module
- Đặc điểm Wi-Fi:
 - Tốc độ: lên đến 150 Mbps.
 - Hỗ trợ WPA/WPA2, WEP bảo mật.
 - Có thể hoạt động ở chế độ Station, Access Point, hoặc cả hai.

2.4.3 Sơ đồ chân (Pinout) & Giao tiếp



Hình 2.7: Sơ đồ chân ESP32 chi tiết

2.4.3.1 Tổng quan chân GPIO

- **GPIO (Digital I/O):** 34 chân — Có thể lập trình đa chức năng
- **ADC (Analog-Digital):** 18 kênh x 12-bit — Đọc tín hiệu analog (0–4095)
- **DAC (Digital-Analog):** 2 kênh x 8-bit — Xuất tín hiệu analog
- **PWM:** 16 kênh — Điều khiển servo, LED, buzzer
- **Touch:** 10 chân — Cảm ứng điện dung
- **I2C:** 2 bus — Giao tiếp LCD, cảm biến
- **SPI:** 3 bus — Giao tiếp tốc độ cao
- **UART:** 3 cổng — Giao tiếp serial

2.4.3.2 Phân bổ chân GPIO phổ biến

- GPIO 34: ADC1_CH6 — Đọc MQ-2 (Input only)
- GPIO 35: ADC1_CH7 — Đọc Flame Sensor (Input only)
- GPIO 5: Digital Input — Flame DO (Pull-up)
- GPIO 18: PWM — Điều khiển Servo (Output)
- GPIO 19: PWM — Điều khiển Buzzer (Output)
- GPIO 21: I2C SDA — LCD (Bidirectional)
- GPIO 22: I2C SCL — LCD (Bidirectional)
- GPIO 23: Digital Output — Relay 1 (Output)
- GPIO 25: Digital Output — Relay 2 (Output)

- GPIO 2: LED_BUILTIN (On-board LED)

2.4.3.3 Chi tiết các giao tiếp

ADC (Analog-to-Digital Converter):

- 18 kênh, độ phân giải 12-bit (0–4095).
- Dùng để đọc giá trị analog từ cảm biến MQ-2, Flame Sensor.
- Công thức chuyển đổi:

PWM (Pulse Width Modulation):

- 16 kênh độc lập, tần số và duty cycle có thể điều chỉnh.
- Điều khiển servo (góc quay), buzzer (âm thanh), LED (độ sáng).
- Tần số: 1 Hz – 40 MHz.

I2C (Inter-Integrated Circuit):

- 2 bus I2C, tốc độ lên đến 400 kHz.
- Chân mặc định: SDA = GPIO 21, SCL = GPIO 22.
- Kết nối: LCD 1602, cảm biến BME280, OLED...

2.4.4 Nguyên lý hoạt động

Chu trình xử lý tổng quan:

- Thu thập tín hiệu: MQ-2, Flame Sensor
- Xử lý: So sánh ngưỡng
- Ra quyết định: Relay, Servo, Buzzer, LED
- Truyền thông: Firebase WiFi

2.4.5 Lập trình thời gian thực (FreeRTOS)

- Multitasking: Nhiều task chạy "đồng thời"
- Dual-Core: Core 0: Wi-Fi, Core 1: Ứng dụng
- Priority: Ưu tiên task cảnh báo khẩn cấp
- ISR (Interrupt): Phản hồi tức thì (~1-10 µs)
- Non-Blocking: Delay không block CPU

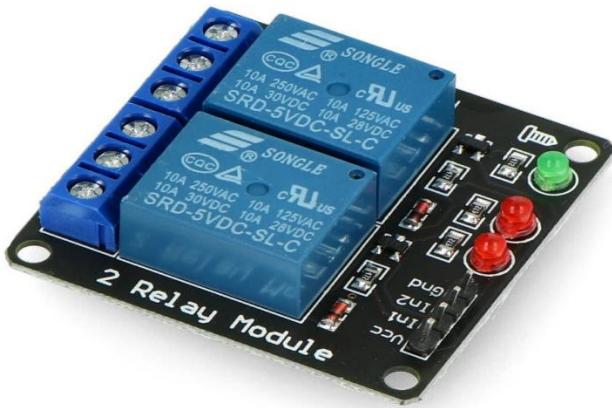
2.4.6 Vai trò trong hệ thống

- Trung tâm xử lý: Thu thập, xử lý dữ liệu từ MQ-2, Flame Sensor
- Ra quyết định: Logic ngưỡng, phân cấp độ nguy hiểm
- Điều khiển thiết bị: Relay, Servo, Buzzer, LED
- Kết nối mạng: Wi-Fi → Firebase, push notification
- Xử lý thời gian thực: FreeRTOS multitasking, ISR khẩn cấp
- Tiết kiệm năng lượng: Deep sleep khi không cần thiết

2.5 Modul Relay Mini 2 kênh 5V 10A

2.5.1 Nguyên lý hoạt động

Module Relay Mini 2 Kênh 5V 10A hoạt động dựa trên nguyên lý relay điện từ (Electromagnetic Relay): khi có tín hiệu điều khiển từ ESP32, dòng điện chạy qua cuộn dây tạo ra từ trường, từ trường này hút cần tiếp điểm (armature) để đóng/mở mạch điện cao áp.

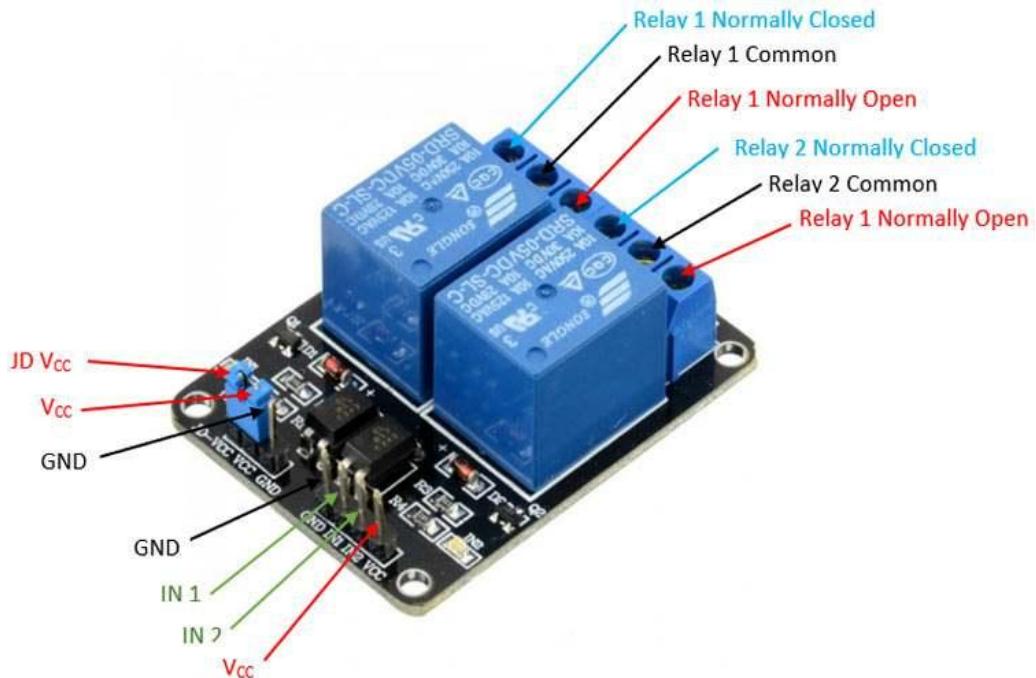


Hình 2.8: Hình ảnh Module Relay Mini 2 Kênh 5V 10A

2.5.1.1 Cấu tạo bên trong Relay điện từ

- Cuộn dây (Coil): Tạo từ trường khi có dòng điện 5V chạy qua
- Lõi sắt (Iron Core): Tăng cường từ trường
- Cần tiếp điểm (Armature): Bộ phận chuyển động, đóng/mở tiếp điểm
- Tiếp điểm NC/NO/COM: Đóng/ngắt mạch tải điện cao áp
- Lò xo hồi (Spring): Đưa armature về vị trí ban đầu khi ngắt điện

2.5.1.2 Quy trình hoạt động chi tiết



Hình 2.9: Sơ đồ pinout Module Relay 2 kênh

2.5.1.3 Công thức logic hoạt động

Module này hỗ trợ chọn mức kích hoạt (Jumper H/L):

Chế độ	Tín hiệu GPIO	Trạng thái Relay
Active LOW (L)	LOW (0V)	ON (Đóng NO-COM)
	HIGH (3.3V)	OFF (Mở NO-COM)
Active HIGH (H)	HIGH (3.3V)	ON (Đóng NO-COM)
	LOW (0V)	OFF (Mở NO-COM)

Công thức (Active LOW - mặc định):

$$Relay = \begin{cases} ON \text{ khi } GPIO = LOW (0V) \\ OFF \text{ khi } GPIO = HIGH (3.3V) \end{cases}$$

2.5.2 Thông số kỹ thuật

- Model: RL-M2-5V / Module Relay Mini 2 Kênh 5V 10A BLK
- Điện áp cuộn dây: 5V DC
- Số kênh: 2 kênh (độc lập)
- Tải AC tối đa: 250V AC / 10A
- Tải DC tối đa: 30V DC / 10A
- Dòng kích hoạt: 15-20 mA
- Dòng tiêu thụ relay hoạt động: ~70 mA (1 kênh), ~140 mA (2 kênh)
- Mức kích hoạt: Chọn được H (High) hoặc L (Low) qua Jumper
- Cách ly: Optocoupler 817C
- Thời gian đóng/mở: < 10 ms
- Kích thước: Nhỏ gọn (~50 x 40 mm)
- Lỗ bắt vít: 4 lỗ, đường kính 3.1 mm

Chân	Tên	Chức năng
VCC	Nguồn dương	5V DC từ nguồn ngoài hoặc VIN ESP32
GND	Nguồn âm	Nối chung với GND ESP32
IN1	Tín hiệu 1	Điều khiển Relay 1 từ GPIO ESP32
IN2	Tín hiệu 2	Điều khiển Relay 2 từ GPIO ESP32
COM1	Common 1	Chân chung Relay 1 — nối với nguồn tải
NO1	Normally Open 1	Thường mở — đóng khi relay ON

NC1	Normally Closed 1	Thường đóng — mở khi relay ON
COM2, NO2, NC2	Relay 2	Tương tự Relay 1
Jumper H/L	Chọn mức kích	H = Active High, L = Active Low

Bảng 2.5: Sơ đồ chân modul relay 2 kênh

2.5.3 Ứng dụng trong hệ thống cảnh báo rò rỉ khí gas

2.5.3.1 Các thiết bị điều khiển qua Relay

- Relay 1 (IN1):
 - Thiết bị: Quạt hút khí
 - Chức năng: Hút khí gas ra ngoài khi phát hiện rò rỉ
 - Cách đấu: NO1-COM1 (BẬT khi có gas)
- Relay 2 (IN2):
 - Thiết bị: (Chưa sử dụng)
 - Chức năng: Dự phòng mở rộng

ESP32	Module Relay	Ghi chú
VIN (5V)	VCC	Cấp nguồn 5V cho relay
GND	GND	Mass chung
GPIO 23	IN1	Điều khiển Relay 1 (Quạt hút)
(Không dùng)	IN2	Dự phòng mở rộng

Bảng 2.6: Sơ đồ kết nối với ESP32 của modul relay

2.5.3.2 *Khả năng mở rộng (Relay 2 - Dự phòng)*

Kênh Relay 2 (IN2) hiện chưa sử dụng, có thể mở rộng trong tương lai:

- Còi báo động: Cảnh báo âm thanh lớn
- Đèn cảnh báo: Đèn nhấp nháy

2.6 Động cơ SERVO SG90

2.6.1 Giới thiệu tổng quan

Servo SG90 là một động cơ servo mini (9 grams) được phát triển bởi TowerPro, là thiết bị điều khiển vị trí phổ biến nhất trong các dự án IoT, robot, và tự động hóa. Servo SG90 có khả năng quay chính xác từ 0° đến 180° với phản hồi vị trí realtime thông qua potentiometer tích hợp, được điều khiển dễ dàng bằng tín hiệu PWM từ ESP32.

2.6.2 Nguyên lý hoạt động

Servo SG90 hoạt động theo nguyên lý điều khiển vòng kín (Closed-Loop Control) sử dụng tín hiệu PWM (Pulse Width Modulation), có khả năng quay góc chính xác từ 0° đến 180° với hệ thống phản hồi vị trí bằng potentiometer.



Hình 2.10: Hình ảnh Servo SG90

2.6.2.1 Nguyên lý hoạt động

Quy trình hoạt động:

- Bước 1: ESP32 gửi tín hiệu PWM đến chân Signal của Servo
- Bước 2: Mạch điều khiển so sánh PWM với vị trí hiện tại từ potentiometer
- Bước 3: Nếu có sai lệch → DC motor quay để điều chỉnh
- Bước 4: Potentiometer đo vị trí mới → phản hồi về mạch điều khiển

- Bước 5: Quá trình lặp lại cho đến khi vị trí khớp với tín hiệu PWM

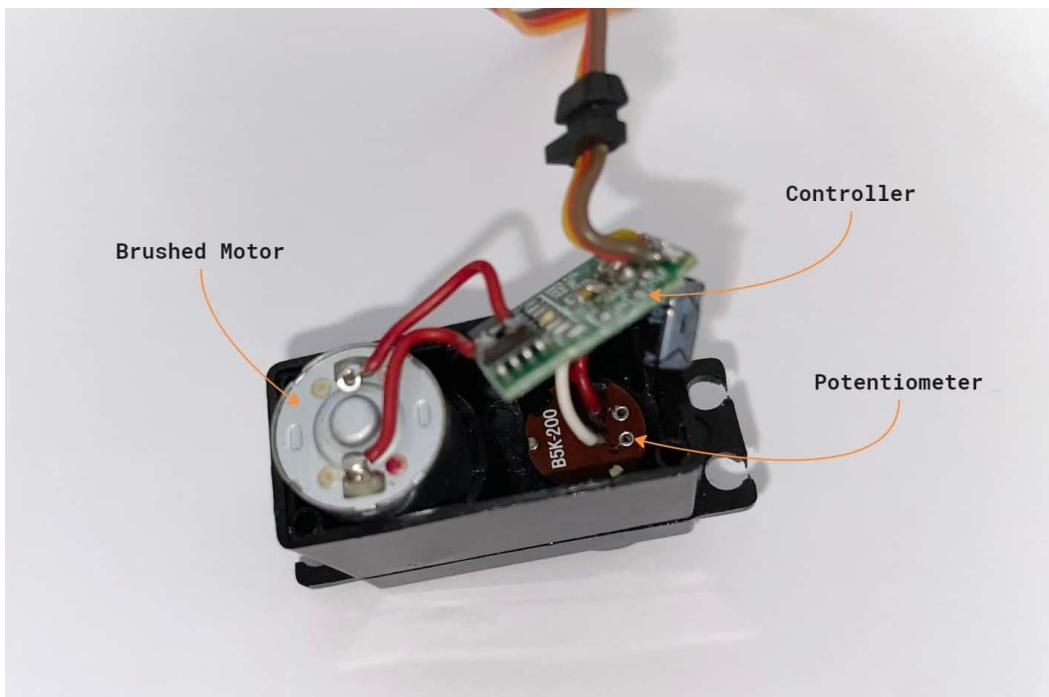
2.6.2.2 Điều khiển góc quay bằng PWM

- Độ rộng xung (Pulse Width): 0.5 ms => Góc quay: 0°
- Độ rộng xung (Pulse Width): 1.0 ms => Góc quay: 45°
- Độ rộng xung (Pulse Width): 1.5 ms => Góc quay: 90° (Vị trí trung tâm)
- Độ rộng xung (Pulse Width): 2.0 ms => Góc quay: 135°
- Độ rộng xung (Pulse Width): 2.5 ms => Góc quay: 180°

Công thức tính góc:

$$\text{Góc} = \frac{(T_{pulse} - 0.5\text{ms})}{2\text{ms}} \times 180^\circ$$

2.6.3 Cấu tạo



Hình 2.11: Hình ảnh cấu tạo bên trong Servo SG90

- DC Motor 3 cực: Tạo chuyển động quay
- Bộ giảm tốc bánh răng: Giảm tốc độ, tăng mô-men xoắn
- Potentiometer: Đo vị trí góc hiện tại, phản hồi về mạch điều khiển
- Mạch điều khiển: So sánh PWM với vị trí thực, điều khiển motor
- Vỏ bảo vệ: Bảo vệ các linh kiện bên trong

2.6.3.1 Thông số kỹ thuật

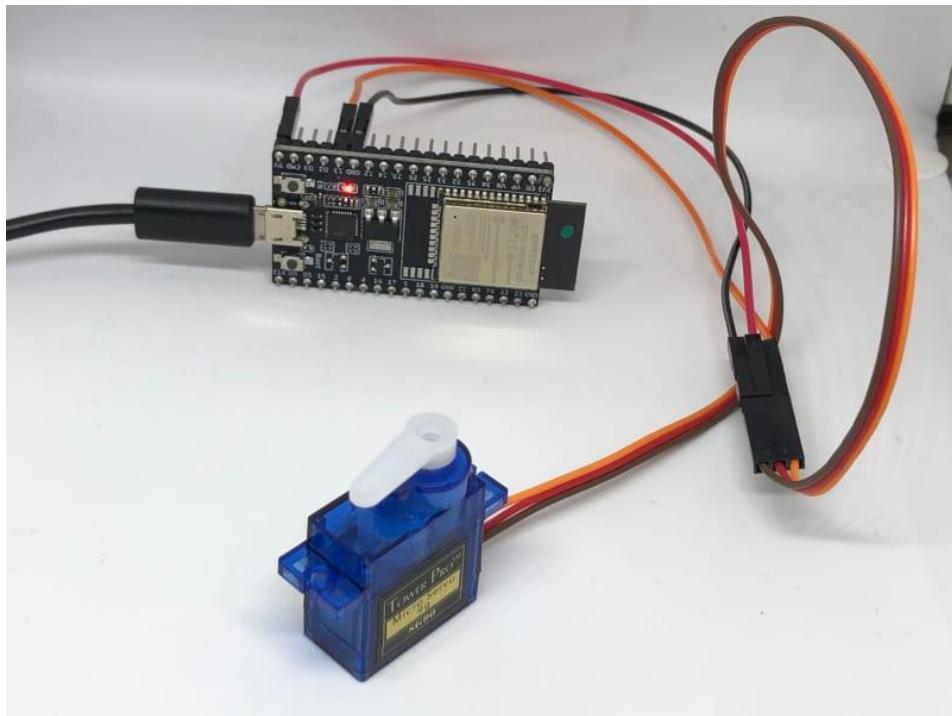
- Kích thước: 23 × 12.2 × 29 mm
- Điện áp hoạt động: 4.8V DC
- Góc quay: 0° – 180°
- Mô-men xoắn: 1.8 kg·cm (4.8V), 2.2 kg·cm (6V)
- Tốc độ: 0.12 sec/60° (4.8V)

- Loại bánh răng: Nhựa (Plastic)
- Tần số PWM: 50 Hz (chu kỳ 20ms)
- Độ rộng xung: 500 – 2400 μ s
- Dòng tiêu thụ: ~200 mA (khi quay)

Dây	Màu	Chức năng
Signal	Cam/Vàng	Nhận tín hiệu PWM từ ESP32
VCC	Đỏ	Nguồn dương 5V
GND	Nâu/Đen	Nguồn âm (Mass)

Bảng 2.7: Sơ đồ chân của Servo SG90

2.6.4 Úng dụng trong hệ thống cảnh báo rò rỉ khí gas



Hình 2.12: Hình ảnh kết nối Servo SG90 với ESP32

2.6.4.1 Chức năng trong hệ thống

Servo SG90 được sử dụng với 1 chức năng duy nhất: mở cửa thoát hiểm tự động khi phát hiện rò rỉ khí gas hoặc lửa.

- Mở cửa thoát hiểm: Khi phát hiện gas/lửa → servo quay mở cửa($0^\circ \rightarrow 180^\circ$)
- Đóng cửa thoát hiểm: Khi an toàn → servo quay đóng cửa về vị trí ban đầu($180^\circ \rightarrow 0^\circ$)

ESP32	Servo SG90	Ghi chú
5V (VIN)	VCC (Đỏ)	Nguồn 5V cho servo
GND	GND (Nâu)	Mass chung
GPIO 18	Signal (Cam)	Tín hiệu PWM điều khiển

Bảng 2.8: Sơ đồ kết nối Servo SG90 với ESP32

2.7 Công nghệ điện toán đám mây (Firebase)

2.7.1 Giới thiệu

Firebase là nền tảng phát triển ứng dụng của Google, cung cấp các dịch vụ Backend-as-a-Service (BaaS) giúp xây dựng hệ thống IoT nhanh chóng mà không cần tự quản lý server.

2.7.2 Các dịch vụ Firebase sử dụng trong dự án

Dịch vụ	Chức năng	Ứng dụng trong hệ thống
Realtime Database	CSDL NoSQL lưu trữ JSON, đồng bộ realtime	Lưu dữ liệu cảm biến, trạng thái hệ thống
Cloud Messaging (FCM)	Gửi push notification	Cảnh báo khẩn cấp đến app/web
Authentication	Xác thực người dùng	Đăng nhập, phân quyền truy cập
Security Rules	Bảo mật dữ liệu	Kiểm soát quyền đọc/ghi

2.7.3 Firebase Realtime Database

Đặc điểm:

- Đồng bộ realtime: Dữ liệu cập nhật tức thì trên tất cả thiết bị
- Hỗ trợ offline: Lưu cache khi mất mạng, tự đồng bộ khi có kết nối
- Dữ liệu JSON: Dễ truy vấn, linh hoạt cấu trúc

Ứng dụng trong đề tài:

- ESP32 ghi dữ liệu cảm biến lên
- Web App lắng nghe realtime, cập nhật dashboard tức thì
- Backend điều khiển ghi lệnh
- ESP32 lắng nghe và thực thi lệnh

2.7.4 Firebase Cloud Messaging (FCM)

Gửi thông báo (push notification) tới web người dùng khi có sự cố.

Luồng hoạt động:

- ESP32 phát hiện gas
- ESP32 ghi vào Firebase
- Backend hoặc Cloud Function phát hiện cảnh báo
- FCM gửi notification
- Người dùng nhận thông báo trên web ngay lập tức

Ứng dụng trong đề tài:

- Cảnh báo khẩn cấp đến tất cả người dùng có quyền
- Thông báo trạng thái hệ thống thay đổi

2.7.5 Authentication & Security Rules

Xác thực người dùng đăng nhập vào dashboard qua email/password, Google,...

Ứng dụng trong đề tài:

- Bảo vệ dữ liệu nhạy cảm (ngưỡng cảnh báo, lịch sử)
- Chỉ admin được phép thay đổi cấu hình
- Người dùng thường chỉ được xem dữ liệu

2.8 Công nghệ phần mềm

2.8.1 Tổng quan công nghệ

Arduino IDE:

- Vai trò: Firmware ESP32
- Chức năng: Lập trình vi điều khiển, đọc cảm biến, điều khiển thiết bị

ReactJS:

- Vai trò: Frontend Web
- Chức năng: Dashboard hiển thị dữ liệu realtime, biểu đồ

Java Spring Boot:

- Vai trò: Backend API
- Chức năng: Xử lý nghiệp vụ, tích hợp Firebase, gửi notification

2.8.2 Giao thức truyền thông

- HTTP/HTTPS: Truyền dữ liệu có mã hóa(ESP32 ↔ Firebase)
- RESTful API: Giao tiếp client-server(Frontend ↔ Backend)
- WebSocket: Kết nối realtime hai chiều(Firebase Realtime sync)

2.8.3 WiFi (IEEE 802.11)

WiFi (Wireless Fidelity) là công nghệ mạng không dây (WLAN - Wireless Local Area Network) dựa trên chuẩn IEEE 802.11, cho phép các thiết bị kết nối internet mà không cần dây cáp

Vai trò trong hệ thống:

WiFi là tầng vật lý (Physical Layer) cho phép ESP32:

- Kết nối internet qua router
- Gửi dữ liệu cảm biến lên Firebase
- Nhận lệnh điều khiển từ Web App
- Gửi/nhận push notification

2.8.4 HTTP/HTTPS (HyperText Transfer Protocol)

HTTP là giao thức truyền tải siêu văn bản, hoạt động theo mô hình Request-Response giữa client và server. HTTPS là phiên bản mã hóa bảo mật của HTTP sử dụng SSL/TLS.

Vai trò trong hệ thống:

ESP32 sử dụng HTTPS để:

- Gửi dữ liệu cảm biến lên Firebase Realtime Database
- Cập nhật trạng thái relay, servo
- Đọc cấu hình ngưỡng, chế độ Auto/Manual từ Firebase

2.8.5 WebSocket

WebSocket là giao thức truyền thông 2 chiều (Bi-directional), Full Duplex, duy trì kết nối liên tục giữa client và server, được chuẩn hóa trong RFC-6455.

So sánh HTTP vs WebSocket:

Đặc điểm	HTTP	WebSocket
Kết nối	Mở → Đóng (mỗi request)	Duy trì liên tục
Hướng	1 chiều (client → server)	2 chiều (client ↔ server)
Độ trễ	Cao (cần request mới)	Thấp (realtime)
Overhead	Lớn (header mỗi request)	Nhỏ (header 1 lần)
Ứng dụng	REST API, tải trang	Chat, IoT, Gaming

- Đồng bộ tức thì: Dữ liệu cập nhật trên Web App ngay khi ESP32 gửi
- Không cần polling: Web App không cần request liên tục
- Tiết kiệm băng thông: Chỉ gửi dữ liệu thay đổi

3. Phân tích yêu cầu chức năng

Yêu cầu chức năng mô tả các chức năng chính mà hệ thống phải cung cấp, tập trung vào quy trình thu thập, xử lý, cảnh báo và tương tác. Các tính năng này được triển khai dựa trên kiến trúc IoT, với ESP32 làm trung tâm xử lý biên và Firebase làm nền tảng đám mây. Dưới đây là phân tích chi tiết từng chức năng cơ bản:

3.1 Thu thập dữ liệu nồng độ khí gas

- Mô tả yêu cầu: Hệ thống phải liên tục thu thập dữ liệu từ cảm biến để giám sát môi trường theo thời gian thực. Dữ liệu bao gồm giá trị nồng độ khí (đơn vị ppm), ID thiết bị (để phân biệt nhiều thiết bị), vị trí lắp đặt (ví dụ: "bếp nhà", "kho gas") và thời gian đo (timestamp theo định dạng ISO 8601). Tần suất thu thập mặc định là mỗi 1-2 giây để đảm bảo phát hiện sớm.
- Tính năng triển khai:

- Sử dụng cảm biến MQ-2 kết nối với chân analog (ADC) và digital của ESP32. Firmware ESP32 đọc giá trị analog để tính toán nồng độ chính xác (sử dụng công thức Rs/Ro với hiệu chuẩn ban đầu), và tín hiệu digital để phát hiện nhanh vượt ngưỡng phần cứng.
- Dữ liệu được lưu tạm cục bộ trên ESP32 (sử dụng buffer nếu mất mạng) trước khi gửi lên đám mây.
- Ràng buộc và giả định: Phụ thuộc vào độ chính xác cảm biến (sai số $\pm 5\text{-}10\%$), giả định môi trường không có nhiễu mạnh (như khói bếp).
- Lợi ích: Đảm bảo dữ liệu đáng tin cậy cho phân tích và lưu trữ lịch sử.

3.2 Phân tích và xử lý sự kiện

- Mô tả yêu cầu: Hệ thống phải phân tích dữ liệu thu thập để phát hiện sự cố. ESP32 hoặc đám mây so sánh nồng độ gas với ngưỡng an toàn (mặc định 300-1000 ppm, có thể cấu hình). Nếu vượt ngưỡng, kích hoạt sự kiện "Rò rỉ khí gas" và chuỗi phản ứng tự động.
- Tính năng triển khai:
 - Xử lý biên trên ESP32: Sử dụng vòng lặp loop() trong firmware để so sánh thời gian thực, tránh độ trễ từ đám mây. Nếu phát hiện, chuyển trạng thái hệ thống từ "NORMAL" sang "ALERT".
 - Hỗ trợ xử lý đám mây (Firebase Cloud Functions) cho phân tích nâng cao, như phát hiện xu hướng tăng dần (ví dụ: nếu nồng độ tăng 20% trong 10 giây).
 - Ràng buộc và giả định: Ngưỡng phải được hiệu chỉnh dựa trên loại khí (LPG cao hơn methane), giả định người dùng có kiến thức cơ bản để cấu hình.
 - Lợi ích: Giảm thiểu thời gian phản hồi, tăng tính an toàn bằng cách kết hợp edge và cloud computing.

3.3 Gửi dữ liệu và cảnh báo từ xa

- Mô tả yêu cầu: Dữ liệu phải được truyền trực tiếp qua Wi-Fi đến đám mây sử dụng giao thức nhẹ như MQTT hoặc API Firebase. Khi có sự cố, gửi push notification khẩn cấp đến ứng dụng di động (ví dụ: "Cảnh báo rò rỉ gas tại bếp!").
- Tính năng triển khai:
 - ESP32 sử dụng thư viện Firebase-ESP32 để gửi JSON data theo chu kỳ.
 - Cảnh báo từ xa: Firebase Cloud Messaging (FCM) tích hợp với backend Spring Boot để gửi notification đến thiết bị đăng ký (token từ Firebase Auth). Hỗ trợ tùy chỉnh thông báo (thêm vị trí, mức độ nghiêm trọng).
 - Ràng buộc và giả định: Phụ thuộc kết nối Wi-Fi ổn định; nếu mất mạng, lưu cục bộ và đồng bộ sau. Giả định người dùng có app di động (iOS/Android).

- Lợi ích: Cho phép giám sát từ xa, phù hợp với ứng dụng trong chung cư hoặc nhà hàng.

3.4 Điều khiển thiết bị an toàn (Actuators)

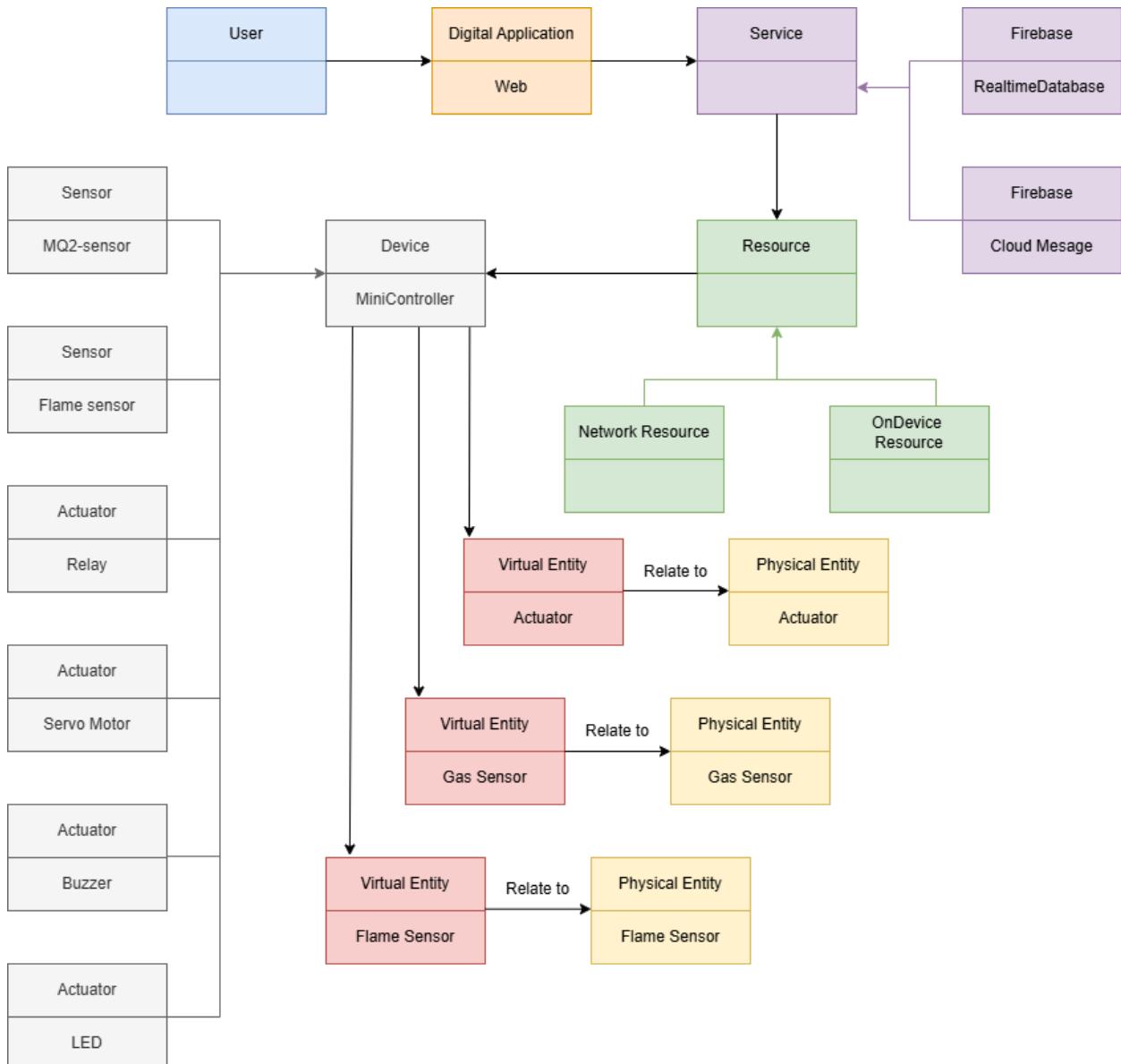
- Mô tả yêu cầu: Đây là chức năng cốt lõi, phản ứng tự động khi phát hiện rò rỉ: kích hoạt báo động tại chỗ (còi), hệ thống thông gió (quạt hút), và mở lối thoát hiểm (cửa sổ/cửa chính).
- Tính năng triển khai:
 - Kích hoạt báo động: digitalWrite() trên ESP32 bật còi buzzer (tần số 1000Hz, kéo dài 5 giây lặp lại) và đèn LED nhấp nháy.
 - Hệ thống thông gió: Relay module kênh 1 bật quạt hút (tự động tắt khi nồng độ giảm < ngưỡng an toàn + hysteresys 100ppm).
 - Mở lối thoát hiểm: Servo quay từ 0° đến 180° để mở cửa (sử dụng thư viện Servo.h), với tùy chọn đóng lại thủ công qua web.
- Ràng buộc và giả định: Actuators phải chịu tải an toàn (relay 10A), giả định lắp đặt đúng vị trí (quạt gần nguồn rò rỉ).
- Lợi ích: Giảm thiểu rủi ro cháy nổ bằng phản ứng tức thì, không cần can thiệp con người.

3.5 Tương tác với người dùng

- Mô tả yêu cầu: Cung cấp giao diện thân thiện để hiển thị, lưu trữ và điều khiển. Dashboard hiển thị trạng thái ("BÌNH THUỐNG" xanh, "CÁNH BÁO" đỏ) và biểu đồ; lưu trữ lịch sử trên cloud; điều khiển thủ công (tắt báo động, thay đổi ngưỡng).
- Tính năng triển khai:
 - Frontend ReactJS: Dashboard với Chart.js cho biểu đồ realtime, nút điều khiển gửi lệnh qua API đến Firebase.
 - Lưu trữ: Firebase lưu history dưới dạng mảng JSON, hỗ trợ query theo thời gian (ví dụ: xem dữ liệu 24h qua).
 - Ràng buộc và giả định: Giao diện tương thích trình duyệt phổ biến (Chrome, Firefox), giả định người dùng đăng nhập qua Firebase Auth.
 - Lợi ích: Tăng tính tương tác, cho phép tùy chỉnh và phân tích dữ liệu lịch sử.

4. Phân tích thiết kế

4.1 Đặc tả mô hình miền



Hình 4.1: *Đặc tả mô hình miền*

1. User (Người dùng)

Người dùng là thực thể tương tác với hệ thống thông qua ứng dụng số (Digital Application). Người dùng có thể theo dõi trạng thái cảm biến và nhận thông báo cảnh báo.

2. Digital Application (Ứng dụng số – Web App)

Ứng dụng Web là giao diện để người dùng:

- Xem dữ liệu cảm biến (Gas, Flame)
- Theo dõi trạng thái thiết bị IoT

- Nhận cảnh báo gas hoặc lửa
- Gửi lệnh điều khiển (bật relay, bật còi, mở servo...)

Ứng dụng Web kết nối với Service.

3. Service (Dịch vụ – Firebase)

Service bao gồm các dịch vụ Firebase:

- Realtime Database: Lưu trữ dữ liệu cảm biến và trạng thái thiết bị
- Cloud Messaging: Gửi thông báo cảnh báo (push notification) Service cung cấp API và tài nguyên cho ứng dụng và thiết bị.

Service kết nối với Resource.

4. Resource (Tài nguyên)

Resource đại diện cho các tài nguyên mà Service cung cấp.

Gồm hai loại:

a. Network Resource

- Tài nguyên nằm trên mạng (Firebase Database)
- Lưu trữ dữ liệu từ cảm biến
- Lưu trạng thái actuator

b. OnDevice Resource

- Tài nguyên nằm trên thiết bị ESP32 (biến, trạng thái, chế độ)
- Thiết bị đọc và ghi dữ liệu tại đây

Resource liên kết lên Service và xuống Device.

5. Device (Thiết bị – MiniController / ESP32)

Thiết bị chính điều khiển toàn bộ hệ thống:

- Thu thập dữ liệu từ cảm biến
- Xử lý và gửi dữ liệu lên Network Resource
- Nhận lệnh điều khiển từ Resource
- Điều khiển các actuator

Device có mối quan hệ "attached to" với toàn bộ Sensor và Actuator.

6. Sensors (Cảm biến)

Gồm hai cảm biến cụ thể:

- a. MQ2 Sensor: Đo nồng độ khí gas (LPG, CO, methane)
- b. Flame Sensor: Phát hiện lửa bằng hồng ngoại

Cảm biến kết nối vào Device.

7. Actuators (Thiết bị chấp hành)

Gồm 4 loại actuator:

- Relay: Bật quạt hút gas
- Servo: Mở cửa
- Buzzer: Báo động âm thanh
- LED: Báo hiệu trạng thái hệ thống

Actuator nhận lệnh điều khiển từ Device.

8. Virtual Entities (Thực thể ảo)

Virtual Entities đại diện cho dữ liệu và trạng thái của cảm biến/actuator trong hệ thống.

Gồm 3 thực thể ảo:

- Virtual Entity – Actuator: Biểu diễn trạng thái của relay, buzzer, led, servo và lưu trên Firebase
- Virtual Entity – Gas Sensor: Biểu diễn giá trị gas đo được
- Virtual Entity – Flame Sensor: Biểu diễn trạng thái có/không có lửa

9. Physical Entities (Thực thể vật lý)

Là đối tượng thật ngoài đời:

- Physical Actuator: Relay, Buzzer, Servo, LED thật
- Physical Gas Sensor: Module MQ2
- Physical Flame Sensor: Module Flame Sensor

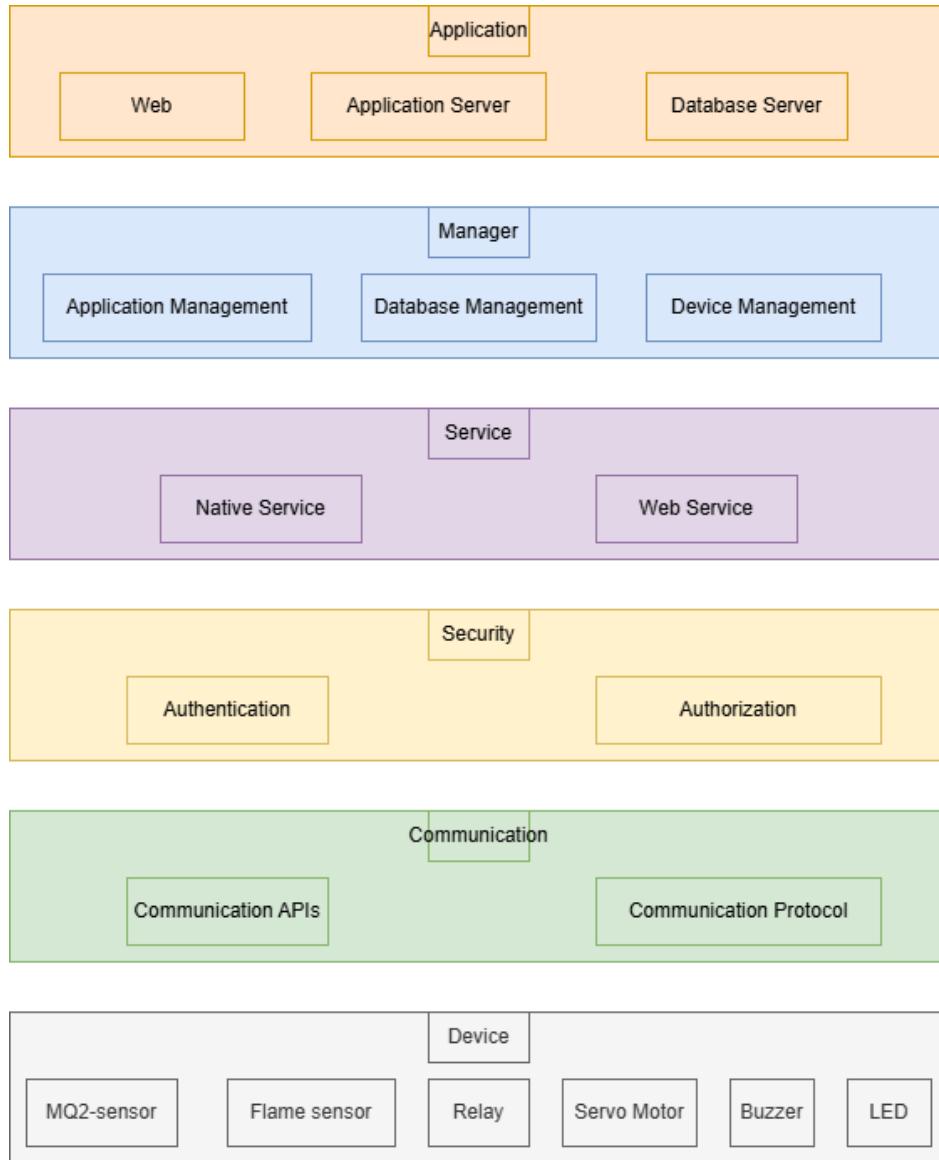
Các Physical Entity tương ứng 1-1 với Virtual Entity.

10. Mối quan hệ giữa các thành phần

Tổng hợp ngắn gọn sơ đồ:

1. User interacts with Digital Application
2. Digital Application uses Service (Firebase)
3. Service exposes Resource
4. Resource is divided into Network Resource and OnDevice Resource
5. Device is associated with Resource
6. Sensors (MQ2, Flame) are attached to Device
7. Actuators (Relay, Servo, Buzzer, LED) are attached to Device
8. Virtual Entities correspond to Physical Entities
9. Physical Sensors monitor environment (gas, flame)
10. Device sends data → Network Resource
11. Device receives commands → Actuators
12. Service → Cloud Message → User (cảnh báo)

4.2 Đặc tả các thành phần chức năng và hoạt động



Hình 4.2: Các thành phần chức năng và hoạt động

4.2.1 Thành phần chức năng

1. Thiết bị IoT (IoT Device)

Trong hệ thống, Thiết bị IoT được ánh xạ vào Device Layer:

- Physical Sensors : MQ-2 Gas Sensor, Flame Sensor
- Physical Actuators: Relay, Servo Motor, Buzzer, LED
- Computing Device: ESP32 (MiniController)

Device Management:

- Quản lý kết nối
- Quản lý cấu hình cảm biến
- Theo dõi trạng thái thiết bị

2. Tài nguyên (Resources)

Trong hệ thống, Resource được ánh xạ vào

Device Layer:

- OnDevice Resource (state của cảm biến, trạng thái actuator, local processing)

Communication Layer

- Communication APIs: Firebase API, HTTP
- Communication Protocols: Wi-Fi 802.11, Firebase WebSocket/REST stream

3. Controller Service

Controller Service được ánh xạ tới Service Layer:

- Native Services: Device Control Service, Sensor Data Processing

4. WebService

WebService được ánh xạ tới Service Layer

- Web Services: REST API hoặc Firebase Hosting Functions

5. Cơ sở dữ liệu (Database)

CSDL sử dụng Firebase Realtime Database để chứa: Dữ liệu cảm biến, Trạng thái thiết bị, Log cảnh báo, được ánh xạ vào Management Layer:

- Database Management: Quản lý dữ liệu thời gian thực, Log và lịch sử cảm biến

Security Layer

- Authorization: Quyền đọc/ghi Database, Firebase Database Rules

6. WebApp (Ứng dụng Web)

WebApp được ánh xạ tới

Application Layer

- Web App (Frontend)

- App Server (Firebase Hosting hoặc Local Server)
- Database Server (Firebase Database)

Management Layer

- Application Management: Giám sát hoạt động WebApp, Quản lý phiên bản

Security Layer

- Authentication: Đăng nhập bằng Firebase Auth, Xác minh người dùng trước khi điều khiển thiết bị

4.2.2 Thành phần hoạt động

Application:

- WebApp: React JS Web
- Application Server: Java SpringBoot
- Database Server: FireBase Realtime Database

Management:

- Application Management: Java SpringBoot
- Database Management: FireBase Realtime Database
- Device Management: Web

Communication:

- Communication APIs: RestFulAPI, FireBaseAPI
- Communication Protocols: HTTP, WebSocket

Security:

- Authentication: Web App, Database
- Authorization: Web App, Database

Device:

- Sensor: MQ2 sensor
- Sensor: Flame sensor
- Actuator: Relay
- Actuator: Servo

- Actuator: Buzzer
- Actuator: LED
- Computing Devices: ESP32

4.3 Đặc tả luồng hoạt động

4.3.1 Thiết bị cảm biến và vi xử lý

- Các hàm chức năng

Hàm	Chức năng
ConnectWifi()	Kết nối WiFi cho thiết bị ESP32 để truyền dữ liệu và nhận lệnh từ server.
readGasSensor()	Đọc dữ liệu nồng độ khí gas (CH4 / LPG) từ cảm biến MQ-2
ShowGasOnOLED()	Hiển thị phần trăm nồng độ khí gas và trạng thái cảnh báo lên màn hình OLED.
getData()	Xử lý dữ liệu nhận từ server có dạng JSON: { "type": "", "message": "" }.
onEventCallback()	Xử lý các sự kiện/command real-time từ server qua WebSocket.
controlBuzzer()	Điều khiển còi báo động bật/tắt dựa trên ngưỡng gas hoặc chế độ điều khiển từ server.
controlRelayFan()	Điều khiển quạt hút/van điện từ khi phát hiện rò rỉ khí gas.
SendData()	Gửi dữ liệu gas hiện tại, trạng thái cảnh báo, trạng thái còi/quạt lên server qua HTTP.
setup()	Khởi tạo chân cảm biến, OLED, WiFi, WebSocket; đặt ngưỡng gas và trạng thái ban đầu.
loop()	Định kỳ đọc nồng độ khí gas, hiển thị OLED, gửi dữ liệu lên server, và kích hoạt cảnh báo khi vượt ngưỡng.

Bảng 4.1: Các hàm chức năng

- Luồng hoạt động

- Đầu tiên, thiết lập chế độ hoạt động (INPUT/OUTPUT) cho các chân của vi điều khiển. Cụ thể, chân kết nối với cảm biến gas MQ-2 được khai báo làm INPUT để đọc giá trị ADC, trong khi các chân điều khiển còi báo động (buzzerPin) và quạt

hút/van điện từ (fanPin) được khai báo làm OUTPUT nhằm cho phép thiết bị kích hoạt cảnh báo khi cần thiết. Đồng thời, khởi tạo màn hình OLED để hiển thị nồng độ gas và trạng thái cảnh báo cho người dùng quan sát.

- Tiếp theo, hệ thống tiến hành kết nối WiFi bằng hàm ConnectWifi() (thực chất sử dụng WiFi.begin(ssid, password) bên trong). Sau khi kết nối thành công, module WebSocket được khởi tạo để lắng nghe các sự kiện từ server (qua hàm onEventCallback()), đồng thời thiết bị lấy ngưỡng cảnh báo gas đã lưu trong bộ nhớ (EEPROM) nhằm duy trì cấu hình ngay cả khi bị mất nguồn.
- Sau mỗi khoảng thời gian xác định được lưu trong biến measureInterval, thiết bị tiến hành đo nồng độ khí gas bằng hàm readGasSensor(). Kết quả đo được so sánh với giá trị trước đó; nếu có thay đổi hoặc vượt ngưỡng cảnh báo đã đặt, hệ thống cập nhật giá trị mới và hiển thị lên màn hình OLED thông qua hàm ShowGasOnOLED().
- Khi giá trị nồng độ gas vượt ngưỡng cho phép, vi xử lý sẽ kích hoạt cơ chế cảnh báo bằng cách bật còi báo động (qua hàm controlBuzzer()) và bật quạt hút/van điện từ để giảm nồng độ gas (qua hàm controlRelayFan()). Nếu mức gas trở lại bình thường, thiết bị tự động tắt cảnh báo tùy theo chế độ hoạt động (manual hoặc auto).
- Sau khi thực hiện đo lường và xử lý, dữ liệu gồm nồng độ gas hiện tại, trạng thái còi, trạng thái quạt và cảnh báo sẽ được gửi lên server qua hàm SendData() theo một chu kỳ nhất định không nhỏ hơn updateInterval. Đây là bước kết thúc một vòng chạy của hệ thống.
- Trong quá trình hoạt động, nếu thiết bị nhận được message hoặc sự kiện từ server thông qua WebSocket (được xử lý trong getData() và onEventCallback()), hệ thống sẽ thay đổi thông số tương ứng. Ví dụ: điều khiển bật/tắt còi, bật/tắt quạt, hoặc cập nhật ngưỡng cảnh báo gas mới. Đặc biệt, khi nhận được ngưỡng gas mới, thiết bị lưu giá trị vào EEPROM để đảm bảo cấu hình được giữ lại khi khởi động lại.

4.3.2 Website quản lý

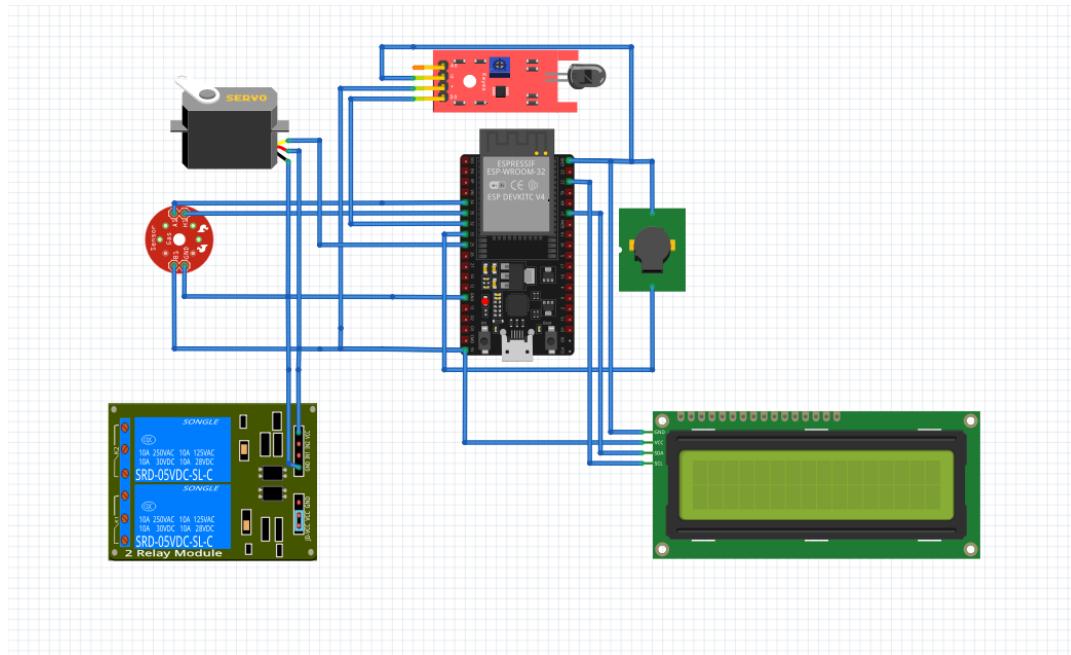
a. Các API sử dụng

STT	Mô tả	Endpoint	Phương thức	Tham số	Trả về
1	Đăng nhập hệ thống	/auth/login	POST	username (string), password (string)	token để cấp quyền
2	Lấy tất cả log khí gas	/gas/logs	GET	Không có	Danh sách log gồm: gasLevel (%), alarmState

					(ON/OFF), fanState, timestamp
3	Đặt ngưỡng cảnh báo gas	/gas/set-threshold	POST	threshold (int - % gas)	message: “success” / “fail”
4	Điều khiển còi báo động	/alarm/manual	POST	action (ON/OFF)	message: “success” / “fail”
5	Chuyển chế độ tự động (auto)	/alarm/auto	POST	Không có	message: “success” / “fail”
6	Điều khiển quạt hút	/fan/manual	POST	action (ON/OFF)	message: “success” / “fail”

Bảng 4.2: Các API sử dụng

4.4 Tích hợp thiết bị



Hình 4.3: Sơ đồ nối chân

STT	Linh kiện	Chân linh kiện	Chân ESP32
1		VCC	Vin (5V)

	Cảm biến khí gas MQ-2	GND	GND
		AO (Analog Out)	GPIO34
		DO (Digital Out)	GPIO35
2	Cảm biến lửa (Flame Sensor)	VCC	Vin (5V)
		GND	GND
		DO	GPIO32
3	Module Relay 2 kênh	VCC	Vin (5V)
		GND	GND
		IN1 (quạt thông gió)	GPIO26
		IN2 (dự phòng)	GPIO27
4	Servo SG90	Dây đỏ (VCC)	Vin (5V)
		Dây nâu/đen (GND)	GND
		Dây cam/vàng (Signal)	GPIO25
5	Còi Buzzer	Chân +	GPIO33
		Chân -	GND
6	LCD 1602 + module I2C	VCC	Vin (5V)
		GND	GND
		SDA	GPIO21
		SCL	GPIO22

Bảng 4.3: Bảng nối chân thiết bị với ESP32

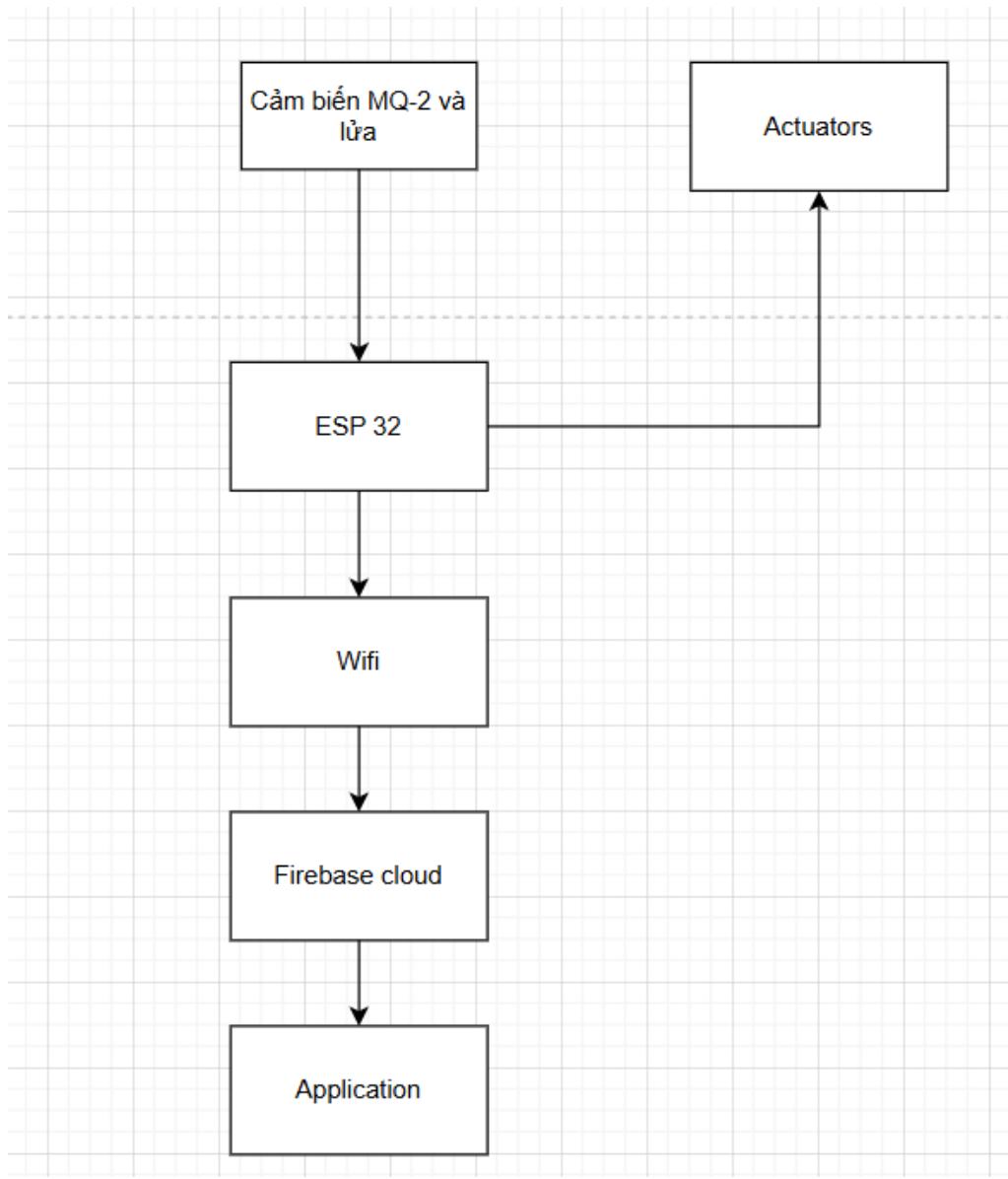
4.5 Thiết kế tổng thể hệ thống

4.5.1 Kiến trúc hệ thống

Hệ thống theo mô hình IoT phân tầng:

- Layer 1: Perception (Cảm biến): Thu thập dữ liệu từ môi trường (MQ-2, cảm biến lửa).
- Layer 2: Processing (Xử lý biên) ESP32 xử lý dữ liệu cục bộ, ra quyết định nhanh (edge computing) để đáp ứng độ trễ ≤ 1 giây.
- Layer 3: Network (Mạng): Truyền dữ liệu qua Wi-Fi (HTTPS/MQTT) đến đám mây.
- Layer 4: Cloud Storage & Analysis: Firebase lưu trữ và đồng bộ realtime. Kiến trúc này đảm bảo tính tin cậy (hoạt động ngoại tuyến cho chức năng cốt lõi) và mở rộng (hỗ trợ nhiều thiết bị qua cấu trúc users/devices).
- Layer 5: Application (Ứng dụng): Backend và frontend xử lý dữ liệu đám mây, cung cấp giao diện người dùng và notification.

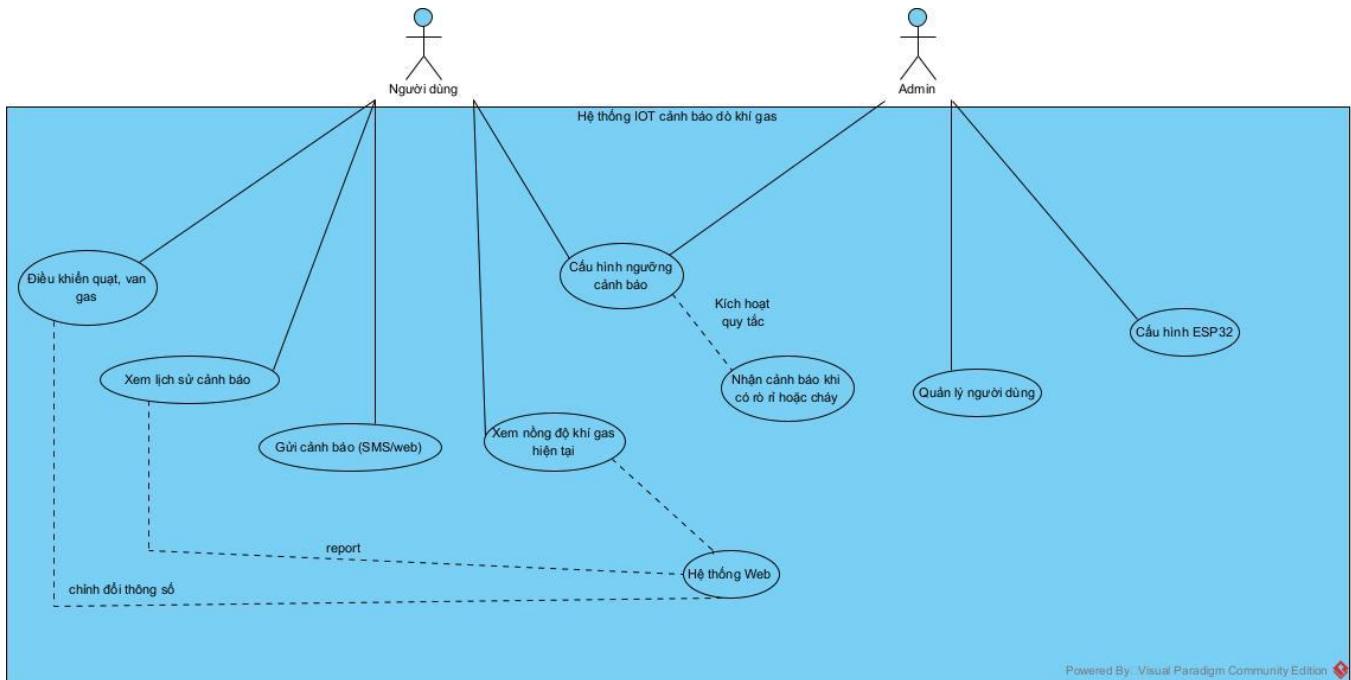
4.5.2 Sơ đồ khái tổng thể



Hình 4.4: Sơ đồ khái tổng thể

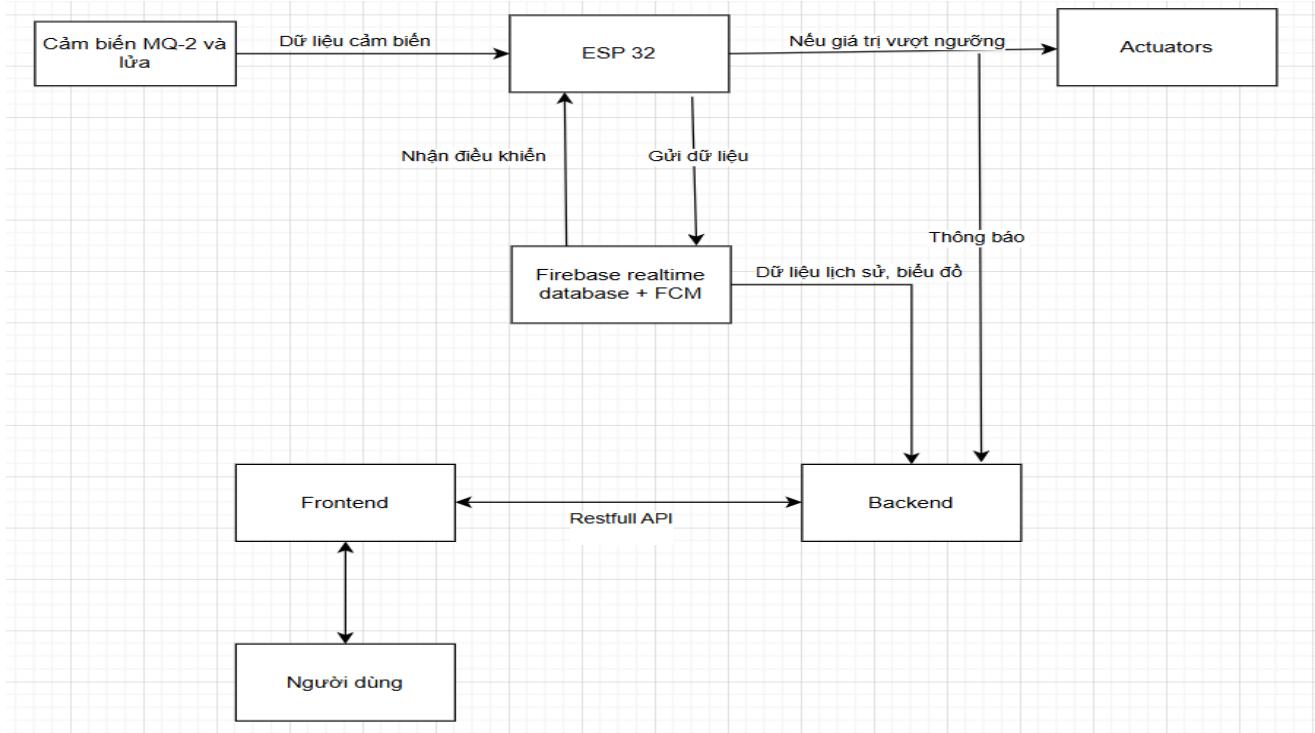
4.6 Thiết kế hệ thống IoT

4.6.1 Usecase tổng quan



Hình 4.5: Usecase tổng quan

4.6.2 Data Flow Diagram



Hình 4.6: Data Flow Diagram

5. Kết quả đạt được

5.1 Về mặt lý thuyết

Qua quá trình nghiên cứu và triển khai, nhóm đã nắm vững nguyên lý hoạt động của vi điều khiển ESP32, cảm biến khí gas MQ-2, cảm biến lửa, cũng như các linh kiện chấp hành (relay, servo, buzzer, LCD I2C). Nhóm hiểu rõ cách thức giao tiếp ngoại vi (ADC, GPIO, I²C, PWM) và các giao thức truyền thông không dây (Wi-Fi 802.11 với TLS). Việc áp dụng mô hình xử lý tại biên (edge computing) kết hợp điện toán đám mây Firebase Realtime Database đã được phân tích và triển khai hiệu quả, đảm bảo dữ liệu được đồng bộ liên tục theo thời gian thực đồng thời vẫn duy trì khả năng phản ứng tức thì khi mất kết nối mạng.

5.2 Về mặt thực tiễn

Hệ thống đã hoạt động ổn định và đáp ứng đầy đủ các yêu cầu đề ra:

- Phát hiện chính xác nồng độ khí gas dễ cháy và sự hiện diện của lửa.

- Khi vượt ngưỡng an toàn, các thiết bị chấp hành (còi báo động, quạt thông gió, servo mở cửa thoát hiểm, đèn LED, LCD) được kích hoạt trong vòng dưới 1 giây ngay cả khi không có Internet.
- Dữ liệu được cập nhật liên tục lên Firebase và hiển thị realtime trên giao diện web ReactJS với độ trễ dưới 5 giây.
- Người dùng nhận được thông báo đẩy khẩn cấp qua FCM trên điện thoại khi xảy ra sự cố.

Hệ thống đã được thử nghiệm thực tế thành công trong nhiều tình huống (rò rỉ gas, phát hiện lửa, mất Wi-Fi, mất điện lưới), chứng minh tính khả thi và hiệu quả cao trong việc nâng cao an toàn phòng chống cháy nổ do khí gas.

6. Kết luận và hướng mở rộng

6.1 Về hệ thống

- Bổ sung thêm cảm biến CO, cảm biến khói quang học, cảm biến nhiệt độ-độ ẩm để tăng khả năng phát hiện đa nguy cơ.
- Phát triển tính năng tự động gọi tổng đài hoặc gửi SMS khi phát hiện sự cố nghiêm trọng (kết hợp module GSM/4G).
- Thiết kế thêm cơ chế chống tràn khí (tự động đóng van điện từ khi phát hiện rò rỉ lớn).
- Nâng cấp giao diện thành ứng dụng di động native (React Native/Flutter) hỗ trợ thông báo offline và widget.
- Chuyển sang sử dụng giao thức MQTT với broker riêng hoặc nền tảng Blynk/Node-RED để giảm chi phí và tăng khả năng mở rộng lên hàng nghìn thiết bị.

6.2 Về bảo mật và sản phẩm hóa

- Áp dụng mô hình phân quyền RBAC (Role-Based Access Control) và xác thực hai lớp (2FA) trên Firebase Authentication.
- Toàn bộ dữ liệu truyền tải và lưu trữ đều được mã hóa end-to-end bằng TLS 1.3 và AES-256.
- Thiết kế vỏ hộp chống cháy, chống nước đạt tiêu chuẩn IP65 và chứng nhận an toàn cháy nổ theo TCVN.
- Thiết kế PCB chuyên dụng, tối ưu kích thước và chi phí để đưa sản phẩm ra thị trường thực tế.