**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

****

**BÁO CÁO**

**Môn học: Các giải pháp và hệ thống IoT tiên tiến**

**Đề tài: Thiết kế hệ thống giám sát, cảnh báo khí gas rò rỉ trong nhà**

**Giảng viên hướng dẫn: Nguyễn Thị Thu Hằng**

**Nhóm 10**

|  |  |
| --- | --- |
| **Thành viên nhóm** | **Mã sinh viên** |
| **Trần Ngọc Đại** | **B21DCVT011** |
| **Hoàng Văn Quang** | **B21DCVT363** |
| **Nguyễn Văn Vũ** | **B21DCVT467** |
| **Nguyễn Vũ Hoài Nam** | **B21DCVT315** |

**Hà Nội - 2025**

**MỤC LỤC**

**[Danh mục hình vẽ](#_Toc199405735)** [2](#_Toc199405735)

**[Thuật ngữ viết tắt](#_Toc199405736)** [3](#_Toc199405736)

**[Lời nói đầu](#_Toc199405737)** [4](#_Toc199405737)

**[CHƯƠNG 1: GIỚI THIỆU CHUNG](#_Toc199405738)** [6](#_Toc199405738)

**[1.1 Lý do lựa chọn đề tài](#_Toc199405739)** [6](#_Toc199405739)

**[1.1.1 Vấn đề rò rỉ khí gas trong nhà](#_Toc199405740)** [6](#_Toc199405740)

**[1.1.2 Vấn đề của hệ thống cảnh báo rò rỉ khí gas hiện nay](#_Toc199405741)** [6](#_Toc199405741)

**[1.2 Yêu cầu và ràng buộc của khách hàng về hệ thống](#_Toc199405742)** [6](#_Toc199405742)

**[1.3 Đề xuất giải pháp thiết kế hệ thống IoT cho khách hàng](#_Toc199405743)** [7](#_Toc199405743)

**[1.4 Mục tiêu và phương pháp thực hiện](#_Toc199405744)** [8](#_Toc199405744)

**[1.4.1 Mục tiêu:](#_Toc199405745)** [8](#_Toc199405745)

**[1.4.2 Phương pháp thực hiện](#_Toc199405746)** [8](#_Toc199405746)

**[CHƯƠNG 2: LỰA CHỌN CÁC THÀNH PHẦN CHO HỆ THỐNG IoT](#_Toc199405747)** [9](#_Toc199405747)

**[2.1 Khối xử lý hệ thống](#_Toc199405748)** [9](#_Toc199405748)

**[2.2 Khối cảm biến](#_Toc199405749)** [12](#_Toc199405749)

**[2.2.1 Cảm biến phát hiện khí gas](#_Toc199405750)** [12](#_Toc199405750)

***[2.2.2 Cảm biến phát hiện lửa](#_Toc199405751)*** [15](#_Toc199405751)

**[2.3 Khối cơ cấu chấp hành](#_Toc199405752)** [17](#_Toc199405752)

**[2.3.1 Động cơ servo](#_Toc199405753)** [17](#_Toc199405753)

**[2.3.2 Module relay](#_Toc199405754)** [19](#_Toc199405754)

**[2.3.3 Động cơ bơm nước](#_Toc199405755)** [20](#_Toc199405755)

**[2.3.4 Quạt gió](#_Toc199405756)** [21](#_Toc199405756)

**[2.4 Khối hiển thị](#_Toc199405757)** [22](#_Toc199405757)

**[2.5 Phần mềm giám sát và điều khiển](#_Toc199405758)** [23](#_Toc199405758)

**[2.6 Các thiết bị khác](#_Toc199405759)** [26](#_Toc199405759)

**[2.6.1 Khối nguồn](#_Toc199405760)** [26](#_Toc199405760)

**[2.6.2 Các nút nhấn (SW1, SW2, SW3, SW4)](#_Toc199405761)** [27](#_Toc199405761)

**[2.6.3 Mạch in PCB](#_Toc199405762)** [27](#_Toc199405762)

**[2.6.4 Khung dựng sản phẩm giả định](#_Toc199405763)** [28](#_Toc199405763)

**[2.7 Đánh giá chi phí và khả năng thực hiện](#_Toc199405764)** [28](#_Toc199405764)

**[CHƯƠNG 3: THI CÔNG HỆ THỐNG VÀ PHẦN MỀM](#_Toc199405765)** [31](#_Toc199405765)

**[3.1 Thiết kế phần cứng hệ thống](#_Toc199405766)** [31](#_Toc199405766)

**[3.1.2 Thiết kế mạch PCB hệ thống](#_Toc199405767)** [34](#_Toc199405767)

**[3.1.3 Viết phần mềm nhúng](#_Toc199405768)** [37](#_Toc199405768)

**[3.1.4 Lắp đặt phần cứng](#_Toc199405769)** [41](#_Toc199405769)

**[3.2 Thiết kế phần mềm hệ thống](#_Toc199405770)** [43](#_Toc199405770)

**[3.2.1 Thiết lập phần mềm](#_Toc199405771)** [43](#_Toc199405771)

**[CHƯƠNG 4: THỬ NGHIỆM VÀ HIỆU CHỈNH](#_Toc199405772)** [49](#_Toc199405772)

**[4.1 Kịch bản thử nghiệm hệ thống](#_Toc199405773)** [49](#_Toc199405773)

**[4.1.1 Kịch bản tổng quan:](#_Toc199405774)** [49](#_Toc199405774)

**[4.1.2 Kịch bản chi tiết:](#_Toc199405775)** [49](#_Toc199405775)

**[4.2 Triển khai hệ thống thực tế và đánh giá ưu nhược điểm](#_Toc199405776)** [51](#_Toc199405776)

**[TÀI LIỆU THAM KHẢO](#_Toc199405777)** [54](#_Toc199405777)

**DANH MỤC HÌNH VẼ**

[Hình 1.1: Thiết kế sơ đồ khối chức năng 5](#_Toc31670)

[Hình 2.1: Sơ đồ chân ESP-WROOM-32S 8](#_Toc32168)

[Hình 2.2: Esp-wroom-32s 10](#_Toc27540)

[Hình 2.3 : Cảm biến khí gas MQ2 11](#_Toc20792)

[Hình 2.4 : Các thành phần trên module cảm biến 13](#_Toc12807)

[Hình 2.5 : Cảm biến phát hiện ngọn lửa 14](#_Toc15002)

[Hình 2.6: Các thành phần trên module cảm biến phát hiện lửa 15](#_Toc16205)

[Hình 2.7: Động cơ servo SG90 16](#_Toc14240)

[Hình 2.8: Module relay 2 kênh 17](#_Toc22104)

[Hình 2.9: Động cơ bơm chìm mini 3-5v 19](#_Toc21065)

[Hình 2.10: Quạt gió (5VDC) 20](#_Toc18584)

[Hình 2.11: Màn hình LCD 16x2 21](#_Toc4086)

[Hình 2.12: Nền tảng Node-Red 23](#_Toc16625)

[Hình 2.13 : Giao diện người dùng của Node-Red 24](#_Toc18159)

[Hình 2.14: Giao thức MQTT trong IoT 24](#_Toc8089)

[Hình 2.15: Nguồn ADAPTER 5VDC 2A 25](#_Toc30047)

[Hình 3.1: sơ đồ nguyên lý hoạt động 30](#_Toc13746)

[Hình 3.2: lớp Top 34](#_Toc19704)

[Hình 3.4: Mạch mô phỏng 36](#_Toc26327)

[Hình 3.5: Mạch thực tế 41](#_Toc214)

[Hình 3.6: Mô hình sản phẩm 42](#_Toc10233)

[Hình 3.7: Tổng quan giao diện của nền tảng Node-Red 43](#_Toc3932)

[Hình 3.8: Ví dụ về giao diện dashboard khi tạo trên Node-Red 44](#_Toc20032)

[Hình 3.9: Giao diện phần mềm trên nền tảng Node-Red 45](#_Toc20513)

[Hình 3.10: Luồng hiển thị dữ liệu cảm biến khí gas trên biểu đồ 46](#_Toc30150)

[Hình 3.11: Luồng hành động khi cảm biến phát hiện được lửa 46](#_Toc3627)

[Hình 3.12: Luồng điều chỉnh ngưỡng cảm biến khí gas và cảnh báo vượt ngưỡng 46](#_Toc18285)

[Hình 3.13: Luồng thay đổi chế độ của hệ thống 46](#_Toc30979)

[Hình 3.14: Luồng điều khiển các thiết bị cơ cấu chấp hành 47](#_Toc24093)

[Hình 3.15: MQTT Broker Mosquitto 47](#_Toc19306)

**Thuật ngữ viết tắt**

|  |  |  |
| --- | --- | --- |
| Thuật ngữ viết tắt | Tên tiếng Anh | Tên tiếng Việt |
| VCC | Voltage Common  Collector | Nguồn dương cấp cho mạch |
| GND | Ground | Chân nguồn âm |
| VR | Variable Resistor | Biến trở |
| I2C | Inter-Integrated Circuit | Giao tiếp nối tiếp nội mạch |
| GPIO | General Purpose Input/Output | chân (pin) trên vi điều khiển hoặc các linh kiện điện tử |
| PWM | Pulse Width Modulation | Điều chế độ rộng xung |
| VDC | Voltage Direct Current | điện áp một chiều |
| AO | Analog Output | Tín hiệu điện áp |
| AP | Access Poin | Điểm truy cập mạng |
| BLE | Bluetooth Low Energy | Bluetooth năng lượng thấp |
| PCB | Printed Circuit Board | Bảng mạch in |

**LỜI NÓI ĐẦU**

Trong bối cảnh xã hội ngày càng phát triển, nhu cầu đảm bảo an toàn phòng cháy chữa cháy và kiểm soát nguy cơ rò rỉ khí gas ngày càng trở nên cấp thiết. Những sự cố cháy nổ liên quan đến khí gas không chỉ gây thiệt hại lớn về tài sản mà còn đe dọa nghiêm trọng đến tính mạng con người, để lại những hậu quả lâu dài và khó lường. Chính vì vậy, việc nghiên cứu và ứng dụng các giải pháp công nghệ hiện đại nhằm giám sát, phát hiện và cảnh báo kịp thời các nguy cơ tiềm ẩn luôn nhận được sự quan tâm sâu sắc từ cộng đồng khoa học và kỹ thuật.

Với sự phát triển mạnh mẽ của công nghệ Internet of Things (IoT), việc xây dựng các hệ thống giám sát và điều khiển từ xa đã trở nên khả thi hơn bao giờ hết..

Đề tài "Thiết kế hệ thống giám sát, cảnh báo khí gas rò rỉ trong nhà" được thực hiện với mục tiêu đưa ra một giải pháp hệ thống IoT trong việc phát hiện nguy cơ cháy nổ. Để đưa ra giải pháp, cùng cần có những yêu cầu của khách hàng và những chi tiết kỹ thuật chuyên môn của nhóm đưa ra giải pháp trên. Vì vậy khi hệ thống đã được tạo ra, hệ thống thu nhập dữ liệu để phát hiện có rò rỉ khí gas không, trường hợp xảy ra sẽ kích hoạt các thiết bị như còi hú, quạt thông gió và gửi thông báo tức thời đến ứng dụng trên phần mềm web người dùng. Từ đó, người dùng có thể nhanh chóng phản ứng, hạn chế thiệt hại về người và tài sản.

Giải pháp cho hệ thống này không chỉ phù hợp với yêu cầu của khách hàng mà sau nó có thể sử dụng rộng rãi trong môi trường trong phòng nhỏ hoặc trong nhà Với chi phí hợp lý, tính năng linh hoạt và khả năng mở rộng cao, hệ thống là một giải pháp hiệu quả đáp ứng yêu cầu của cuộc sống hiện đại.

Báo cáo dưới đây sẽ trình bày chi tiết về quá trình phân tích yêu cầu người dùng, lựa chọn thiết bị, thiết kế và xây dựng giải pháp cho hệ thống. Việc xây dựng hệ thống từ kết nối phần cứng, lập trình phần mềm cho đến thử nghiệm thực tế. Qua đó, báo cáo sẽ phân tích và làm rõ hiệu quả của giải pháp, mở ra tiềm năng ứng dụng thực tiễn trong tương lai.

**CHƯƠNG 1: GIỚI THIỆU CHUNG**

**1.1 Lý do lựa chọn đề tài**

**1.1.1 Vấn đề rò rỉ khí gas trong nhà**

Hiện nay, khí gas (LPG, CNG, CH4,...) được sử dụng rộng rãi trong gia đình, nhà hàng, khách sạn và các khu công nghiệp. Tuy nhiên, việc rò rỉ khí gas có thể dẫn đến nguy cơ cháy nổ, ngộ độc khí, gây thiệt hại nghiêm trọng về tài sản và con người.Thực tế đã có nhiều vụ tai nạn do rò rỉ khí gas xảy ra, đặc biệt là tại các khu vực bếp ăn, nhà máy, hoặc nơi sử dụng gas với số lượng lớn. Nguyên nhân chủ yếu là do rò rỉ từ van, ống dẫn, bình gas cũ, hoặc do bất cẩn của người sử dụng. Vì vậy, việc xây dựng một hệ thống cảnh báo khí gas thông minh là rất cần thiết để phát hiện sớm rò rỉ, cảnh báo kịp thời và tránh các sự cố nguy hiểm.

**1.1.2 Vấn đề của hệ thống cảnh báo rò rỉ khí gas hiện nay**

Các hệ thống cảnh báo rò rỉ khí gas trong nhà là một phần quan trọng trong giải pháp an toàn, đặc biệt với những gia đình sử dụng bếp gas hoặc các thiết bị đốt nhiên liệu. Mặc dù nó có hiệu quả nhưng nó vẫn có một số vấn đề như không phát hiện đủ sớm, báo động giả do thiết bị quá nhạy, thiết bị cắm điện đặt gần chỗ bếp gas dẫn đến cháy nổ khi bị rò rỉ. Ngoài ra một số thiết bị chưa kết nối với hệ thống điều khiển hoặc cảnh báo từ xa dẫn đến việc khó giám sát.

**1.2 Yêu cầu và ràng buộc của khách hàng về hệ thống**

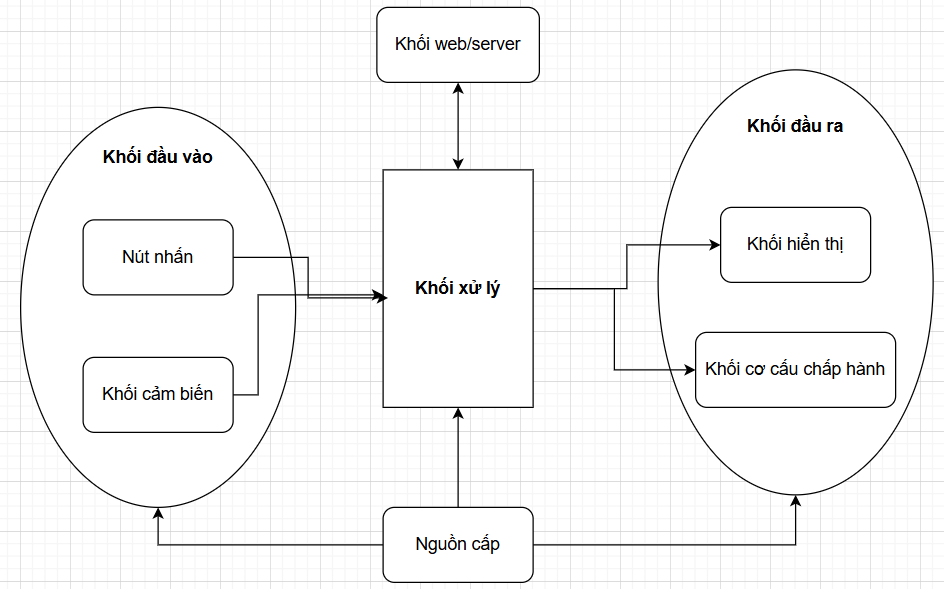
Sau khi tìm hiểu và tham khảo ý kiến khách hàng về hệ thống thì nhóm đã đưa ra được những yêu cầu sơ bộ về sản phẩm cũng như những thông tin, chức năng của sản phẩm.

Hệ thống IoT cảnh báo khí gas được phát triển dựa trên nhu cầu cấp thiết của khách hàng về một giải pháp giám sát an toàn, phát hiện kịp thời rò rỉ khí gas trong không gian sinh hoạt. Khách hàng yêu cầu hệ thống phải đáp ứng các tiêu chí cốt lõi: Phát hiện nhanh chóng thông qua cảm biến có độ nhạy cao, đảm bảo cảnh báo ngay khi nồng độ khí vượt ngưỡng an toàn; Đa dạng phương thức cảnh báo, bao gồm cảnh báo vật lý tại chỗ và thông báo từ xa qua ứng dụng di động. Tiết kiệm năng lượng với khả năng hoạt động liên tục. Thiết kế đơn giản, dễ lắp đặt và vận hành, phù hợp với người dùng không chuyên. Chi phí tối ưu, cân bằng giữa hiệu năng và giá thành để phù hợp với ngân sách hộ gia đình.

**1.3 Đề xuất giải pháp thiết kế hệ thống IoT cho khách hàng**

Nhóm chúng em đề xuất giải pháp hệ thống IoT giám sát, cảnh báo khí gas rò rỉ trong nhà như sau:

Hệ thống sử dụng một vi xử lý kết hợp với các cảm biến phát hiện khí gas và phát hiện lửa. Hệ thống sẽ tự động điều khiển cơ cấu chấp hành khi phát hiện sự cố rò rỉ. Thiết bị kết nối với wifi trong nhà để gửi dữ liệu đến server. Dữ liệu hiển thị trên phần mềm dưới dạng biểu đồ. Hệ thống có chức năng cảnh báo qua email khi phát hiện khí gas vượt ngưỡng và cho phép người quản lý theo dõi từ xa qua thiết bị di động. Người dùng có thể cài đặt điều khiển thủ công hoặc tự động cho hệ thống khi gặp sự cố. Giải pháp này dễ mở rộng, chi phí thấp và phù hợp với khả năng lắp đặt trong nhà của khách hàng.



Hình 1.1: Thiết kế sơ đồ khối chức năng

Mô hình gồm có 3 khối chính: Khối đầu vào, khối xử lý và khối đầu ra, trong các khối chứa một số khối để thực hiện các chức năng khác nhau. Ngoài ra, toàn bộ hệ thống được cấp nguồn để hoạt động ổn định, khối xử lý được kết nối 2 chiều với khối server dùng để truyền và nhận dữ liệu trên môi trường Internet.

Hệ thống hoạt động với khối xử lý làm trung tâm, liên tục giám sát các tín hiệu từ cảm biến. Khi phát hiện có rò rỉ gas hoặc cháy, nó sẽ:

* Hiển thị cảnh báo trên khối hiển thị.
* Điều khiển hoạt động của khối cơ cấu chấp hành, giúp lưu thông khí gas rò rỉ ra bên ngoài

Ngoài ra, nút nhấn được sử dụng để kích hoạt các chức năng điều khiển và thiết lập thông số. Hệ thống có khả năng phản ứng tự động, nhanh chóng và thông minh khi có sự cố xảy ra, phù hợp cho các ứng dụng phòng chống cháy nổ thông minh trong môi trường gia đình hoặc công nghiệp nhỏ.

**1.4 Mục tiêu và phương pháp thực hiện**

**1.4.1 Mục tiêu:**

Mục tiêu của đề tài là xây dựng một mô hình sản phẩm nhỏ gọn, phù hợp để lắp đặt trong không gian gia đình hoặc những khu vực nhỏ cần giám sát an toàn. Hệ thống được thiết kế để phát hiện rỏ rỉ khí gas ở những khu vực dễ phát hiện, dữ liệu thu được sẽ hiển thị trực tiếp trên màn hình và cho phép người dùng điều khiển các chức năng cơ bản ngay tại chỗ. Ngoài ra, hệ thống còn hỗ trợ kết nối từ xa, giúp người dùng có thể theo dõi và giám sát các chỉ số như khí gas thông qua mạng Internet. Khi phát hiện có dấu hiệu rò rỉ gas hoặc cháy nổ, hệ thống sẽ tự động kích hoạt cảnh báo và thông báo đến người dùng kịp thời. Bên cạnh đó, phần mềm điều khiển được xây dựng với tiêu chí hoạt động ổn định, dễ sử dụng, giao diện thân thiện nhằm mang lại trải nghiệm tốt nhất cho người dùng cuối.

**1.4.2 Phương pháp thực hiện**

Trong quá trình thực hiện đề tài, trước tiên nhóm em tiến hành phân tích các yêu cầu kỹ thuật và ràng buộc, bao gồm phạm vi hoạt động, điều kiện môi trường, khả năng cấp nguồn và tính tương thích giữa các linh kiện. Dựa trên đó, nhóm em tiến hành lựa chọn các thành phần phần cứng phù hợp cho mô hình cũng như các chức năng và khả năng xử lý. Sau khi hoàn tất khâu chuẩn bị, nhóm em tiến hành thi công mô hình, kết nối các linh kiện theo sơ đồ nguyên lý và xây dựng phần mềm điều khiển hệ thống. Cuối cùng, hệ thống được thử nghiệm thực tế trong nhiều tình huống, tiến hành đánh giá mức độ ổn định, độ chính xác và khả năng phản hồi để đảm bảo đề tài đạt được các mục tiêu đã đề ra.

**CHƯƠNG 2: LỰA CHỌN CÁC THÀNH PHẦN CHO HỆ THỐNG IoT**

**2.1 Khối xử lý hệ thống**

Khối xử lý đóng vai trò trung tâm trong hệ thống, chịu trách nhiệm tiếp nhận dữ liệu đầu vào từ các cảm biến, xử lý thông tin và đưa ra tín hiệu điều khiển tương ứng. Để đảm bảo hiệu quả hoạt động của hệ thống trong môi trường IoT, khối xử lý cần đáp ứng các yêu cầu quan trọng sau:

* **Khả năng xử lý**: Đảm bảo đủ năng lực để xử lý đồng thời nhiều tác vụ như thu thập dữ liệu, truyền thông không dây, mã hóa và điều khiển thiết bị ngoại vi.
* **Nhỏ gọn và tích hợp cao:** Kích thước nhỏ giúp dễ dàng tích hợp vào các thiết kế phần cứng có không gian hạn chế.
* **Chi phí thấp:** Lựa chọn giải pháp xử lý tiết kiệm giúp giảm giá thành sản phẩm cuối cùng, phù hợp với quy mô triển khai của nhóm.
* **Hỗ trợ kết nối mạng không dây:** Khả năng kết nối Wi-Fi giúp thiết bị giao tiếp với hệ thống giám sát, điều khiển từ xa hoặc gửi dữ liệu về máy chủ lưu trữ.
* **Hỗ trợ đa giao diện:** Dễ dàng tích hợp với cảm biến, màn hình, mô-đun điều khiển để thu dữ liệu và điều khiển
* **Tiết kiệm năng lượng:** Có thể chuyển sang các chế độ ngủ sâu (deep sleep) hoặc ngủ nhẹ (light sleep) khi không hoạt động để kéo dài thời lượng pin.

Với các yêu cầu trên, ESP32 là một lựa chọn lý tưởng, đặc biệt là module ESP WROOM-32S nhờ sự cân bằng hoàn hảo giữa hiệu năng, tính năng và chi phí.

ESP-WROOM-32S là một module Wi-Fi và Bluetooth dựa trên vi điều khiển ESP32 của Espressif Systems. Được thiết kế cho các ứng dụng IoT, ESP-WROOM-32S tích hợp đầy đủ các tính năng mạng không dây, khả năng xử lý dữ liệu mạnh mẽ và tiết kiệm năng lượng. Đây là một giải pháp lý tưởng cho các hệ thống nhúng thông minh, từ nhà thông minh, tự động hóa công nghiệp, cho đến các ứng dụng y tế và nông nghiệp thông minh.

Module này được sản xuất dựa trên bộ vi điều khiển ESP32, một SoC (System-on-Chip) mạnh mẽ với lõi kép (dual-core) Xtensa LX6 có tốc độ xung nhịp tối đa lên đến 240MHz. Với bộ nhớ RAM và Flash tích hợp, ESP-WROOM-32S cung cấp môi trường lý tưởng để thu thập dữ liệu cảm biến, điều khiển thiết bị từ xa, cho đến xử lý tín hiệu thời gian thực.

A close-up of a circuit board

Description automatically generated

Hình 2.1: Sơ đồ chân ESP-WROOM-32S

**Cấu tạo và tính năng nổi bật**

ESP-WROOM-32S được đóng gói nhỏ gọn, sử dụng vỏ kim loại để giảm nhiễu điện từ (EMI) và bảo vệ các linh kiện bên trong. Module bao gồm vi điều khiển ESP32, anten tích hợp và các thành phần phụ trợ giúp đảm bảo hiệu suất hoạt động ổn định. Anten PCB được tối ưu hóa để tăng phạm vi phủ sóng và độ tin cậy trong kết nối không dây.

Module hỗ trợ cả Wi-Fi chuẩn 802.11 b/g/n (băng tần 2.4GHz) và Bluetooth v4.2 (bao gồm Bluetooth Classic và BLE). Nó không chỉ kết nối mạng LAN mà còn dễ dàng giao tiếp với các thiết bị di động hoặc cảm biến qua Bluetooth. Với các cổng giao tiếp UART, SPI, I2C, PWM, và ADC, module có thể kết nối với nhiều loại thiết bị ngoại vi, từ cảm biến, động cơ, cho đến màn hình hiển thị.

**Thông số kỹ thuật của ESP-WROOM-32S**

- Lõi kép Xtensa LX6, tần số tối đa 240MHz.

- Bộ nhớ: 520KB SRAM và 4MB Flash.

- Dải điện áp hoạt động: 3.0V - 3.6V.

- Tiêu thụ điện năng thấp nhờ các chế độ tiết kiệm năng lượng (ngủ sâu và ngủ nhẹ).

- Wi-Fi: Chuẩn 802.11 b/g/n với tốc độ truyền tối đa 150Mbps.

- Bluetooth: Hỗ trợ cả BLE và Bluetooth Classic.

-34 chân GPIO có thể cấu hình.

- Hỗ trợ UART, I2C, SPI, ADC (12-bit), DAC (8-bit), và PWM.

**-** Hỗ trợ mã hóa WPA/WPA2, SSL/TLS để đảm bảo an toàn dữ liệu.

-Module nhỏ gọn, dễ dàng tích hợp vào các mạch in PCB.

-Anten PCB tích hợp giúp tăng cường hiệu suất truyền tải tín hiệu.

**Ứng dụng và lợi ích**

ESP-WROOM-32S mang lại nhiều lợi ích trong các ứng dụng IoT nhờ tích hợp đầy đủ tính năng mạnh mẽ. Một số ứng dụng chính như Nhà thông minh, vườn cây, Xe điều khiển từ xa,…

**Nguyên lý hoạt động:**

- Khi được cấp nguồn và lập trình firmware, ESP-WROOM-32S hoạt động như một bộ điều khiển trung tâm, thực hiện thu thập dữ liệu từ các thiết bị ngoại vi qua các cổng giao tiếp. Dữ liệu sau đó được xử lý trực tiếp trên module và có thể truyền tới máy chủ hoặc thiết bị điều khiển thông qua Wi-Fi hoặc Bluetooth.

- Với khả năng chạy đa nhiệm nhờ hệ điều hành thời gian thực (FreeRTOS), nó cho phép xử lý nhiều tác vụ cùng lúc, chẳng hạn như thu thập dữ liệu cảm biến và truyền dữ liệu không dây mà không gây gián đoạn hệ thống.

ESP-WROOM-32S là một giải pháp toàn diện cho các ứng dụng IoT, với khả năng xử lý mạnh mẽ, giao tiếp linh hoạt và tính năng tiết kiệm năng lượng. module này phù hợp cho nhiều dự án IoT trong thực tế. Với sự hỗ trợ mạnh mẽ từ cộng đồng và tài liệu hướng dẫn chi tiết, ESP-WROOM-32S là sự lựa chọn lý tưởng cho cả người mới học lập trình IoT và các nhà phát triển chuyên nghiệp.



Hình 2.2: Esp-wroom-32s

**2.2 Khối cảm biến**

**2.2.1 Cảm biến phát hiện khí gas**

Cảm biến khí gas là thiết bị dùng để phát hiện và đo lường các loại khí có trong môi trường xung quanh hoặc các ứng dụng đo lường nồng độ khí trong công nghiệp. Các loại khí gas này có thể là khí độc hại, nổ, cháy, hoặc có khả năng gây ảnh hưởng đến sức khỏe của con người và môi trường làm việc.

Cảm biến rò rỉ khí gas hoạt động dựa trên nguyên lý rằng các loại khí khác nhau sẽ tương tác với các chất khác nhau và có thể làm thay đổi điện trở của vật liệu dẫn điện, chẳng hạn như chất bán dẫn hoặc oxide kim loại. Khi khí gas tương tác với vật liệu dẫn điện, nó sẽ làm thay đổi điện trở của vật liệu này. Cảm biến sử dụng một mạch chuyển đổi để đo lường điện trở này và phát hiện sự thay đổi khi có sự hiện diện của khí gas. Từ đó, con người có nhiều phương pháp phát hiện sự rò rỉ khí gas như sau:

- **Phương pháp hấp phụ:** Cảm biến sử dụng các vật liệu hấp phụ như than hoạt tính hoặc zeolite để hấp thụ các loại khí độc hại. Khi khí tương tác với vật liệu hấp phụ, nó sẽ được hấp thụ bởi các vật liệu này và dẫn đến thay đổi điện trở của cảm biến.

- **Phương pháp quang phổ:** Cảm biến sử dụng kỹ thuật quang phổ để phát hiện sự có mặt của các loại khí trong không khí. Khi khí đi qua, nó sẽ hấp thụ một số loại ánh sáng, gây ra sự thay đổi màu sắc của ánh sáng. Cảm biến sẽ phát hiện và đo lường sự thay đổi này để xác định loại khí có mặt trong môi trường.

- **Phương pháp dẫn điện:** Cảm biến sử dụng sự thay đổi điện trở khi khí tương tác với các vật liệu dẫn điện như bạc hoặc kim loại dẫn điện khác. Khi khí tương tác với vật liệu dẫn điện, nó sẽ thay đổi khả năng dẫn điện của vật liệu và dẫn đến thay đổi điện trở của cảm biến.

Từ các phương pháp đó, người ta đã phân loại cảm biến phát hiện rò rỉ khí gas. Một số loại cảm biến điển hình như sau:

- **Cảm biến khí gas bán dẫn:** Sử dụng các vật liệu bán dẫn như SnO2 (oxit thiếc) hoặc WO3 (oxit wolfram) để tương tác với khí và thay đổi điện trở.

- **Cảm biến khí gas điện hóa:** Đây là một loại cảm biến được sử dụng để phát hiện khí trong môi trường xung quanh bằng cách sử dụng phản ứng điện hóa giữa khí và điện cực.

- **Cảm biến khí gas xúc tác:** Cảm biến hoạt động dựa trên nguyên lý rằng khi khí gas tương tác với chất xúc tác, các phản ứng hóa học xảy ra và có thể đo được bằng các phương pháp điện hóa.

Với một số thông tin mà chúng em cung cấp được, nhóm đã lựa chọn loại cảm biến khí gas bán dẫn để phát hiện rò rỉ khí gas, cụ thể hơn chúng em lựa chọn cảm biến phát hiện rò rỉ khí gas MQ2.



Hình 2.3 : Cảm biến khí gas MQ2

MQ2 là cảm biến khí có độ nhạy cao với LPG, Propane và Hydrogen, mê-tan (CH4) và hơi dễ bắt lửa khác, với chi phí thấp và phù hợp cho các ứng dụng khác nhau. Điều này làm cho nó trở thành lựa chọn phổ biến trong môi trường trong phòng hoặc công nghiệp, nhờ vào mạch điện đơn giản và chi phí sản xuất thấp.

**Một số thông tin về thông số kỹ thuật của cảm biến như sau:**

* Điện áp hoạt động : 5V
* Dòng điện : 180mA
* Điện trở tải: 20 KΩ
* Điện trở cảm biến: 10 KΩ – 60 KΩ
* Công suất : 900mW
* Đầu ra: Digital & Analog
* Dải hoạt động: 300 ppm – 10000 ppm
* Khí phát hiện: LPG, CO, CH4

Sơ đồ khối của module bao gồm:

**VCC:** Chân cấp nguồn, điện áp hoạt động thường là 5V

**GND:** Chân nối đất

**Digital out:** Chân xuất đầu ra digital, bằng cách đặt giá trị ngưỡng trên chiết áp

**Analog Out:** Chân xuất điện áp analog 0-5V dựa trên nồng độ khí

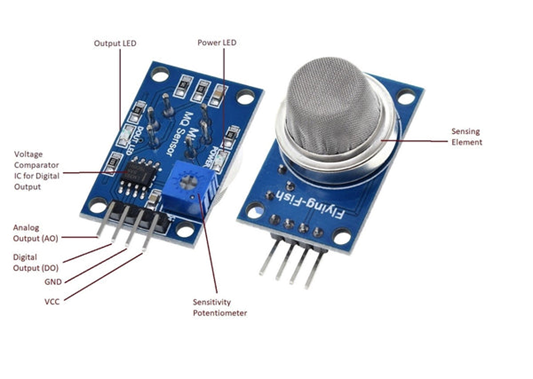
Ngoài ra, còn có một số linh kiện khác trên module như sau:

**Đèn LED nguồn:** Cho biết module có đang hoạt động không

**Đèn LED báo tín hiệu D0:** Cho biết đầu ra của cảm biến

**Biến trở cảm biến:** Điều chỉnh độ nhạy cho cảm biến

**IC LM393:** Dùng để so sánh điện áp cho Digital Output

****

Hình 2.4 : Các thành phần trên module cảm biến

Module hoạt động tốt khi được đặt gần bếp gas, đường ống khí, khói, nó nhanh chóng phát hiện và tăng giá trị nồng độ khí gas lên, từ đó, khối xử lý thu nhập được và nhanh chóng xử lý các thành phần khác. Ngoài ra, ta có thể điều chỉnh độ nhạy cảm biến, giúp cho chúng phát hiện khí gas một cách hiệu quả hơn bằng việc mở rộng dải cảm biến của ngưỡng khí.

Trên thị trường, module này có giá khá rẻ, chỉ khoảng 30 nghìn đồng nên phù hợp cho môi trường trong phòng, trong ô tô.

***2.2.2 Cảm biến phát hiện lửa***

Cảm biến phát hiện lửa là thiết bị được sử dụng để phát hiện sự hiện diện của ngọn lửa và các yếu tố liên quan đến sự cháy nổ như nhiệt độ cao, tia hồng ngoại, tia cực tím, v.v. Khi phát hiện thấy ngọn lửa, cảm biến ngọn lửa sẽ truyền tín hiệu đến hệ thống báo cháy để kích hoạt chuông báo động, đèn nhấp nháy và các biện pháp dập lửa tự động. Cảm biến ngọn lửa thường được sử dụng trong các khu công nghiệp, tòa nhà thương mại và trong nhà để phát hiện sớm các đám cháy và giúp ngăn ngừa các thảm họa tiềm ẩn.

Cảm biến ngọn lửa thường sử dụng cảm biến phát hiện ánh sáng hoặc phát hiện để ngọn lửa phát ra. Khi có ngọn lửa, cảm biến sẽ phát hiện bức xạ hồng ngoại hoặc ánh sáng và tạo ra tín hiệu điện. Tín hiệu này sau đó được hệ thống phát hiện cháy xử lý để kích hoạt báo động hoặc kích hoạt các biện pháp chữa cháy.

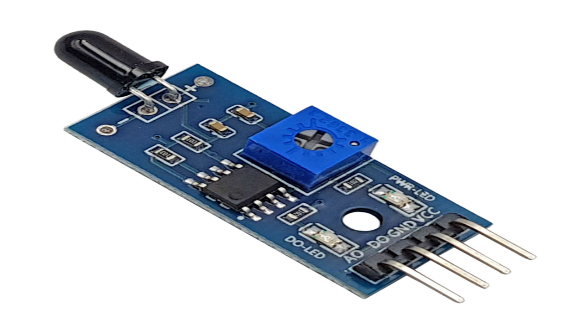
Cảm biến phát hiện lửa được phân loại như sau:

- **Cảm biến ngọn lửa quang học:** Cảm biến ngọn lửa quang học dựa vào việc phát hiện ánh sáng phát ra từ ngọn lửa. Chúng thường bao gồm một bộ tách sóng quang, chuyển đổi ánh sáng thành tín hiệu điện và một thấu kính tập trung ánh sáng vào máy dò.

- **Cảm biến ngọn lửa hồng ngoại:** Cảm biến ngọn lửa hồng ngoại phát hiện bức xạ hồng ngoại phát ra từ ngọn lửa. Họ sử dụng một cảm biến chuyên dụng nhạy cảm với bước sóng hồng ngoại. Khi có ngọn lửa, nó sẽ phát ra bức xạ hồng ngoại và được cảm biến hấp thụ, tiếp đó nó chuyển thành tín hiệu điện cho biết sự hiện diện của ngọn lửa.

- **Cảm biến ngọn lửa tia cực tím:** Cảm biến ngọn lửa tia cực tím phát hiện bức xạ cực tím (UV) phát ra từ ngọn lửa. Những cảm biến này thường bao gồm bộ tách sóng quang nhạy tia cực tím và bộ lọc ngăn chặn bức xạ tia cực tím không bắt lửa.

Từ những thông tin tìm hiểu trên, nhóm chúng em lựa chọn cảm biến phát hiện ngọn lửa IR. Đây là cảm biến ngọn lửa hồng ngoại, nghĩa là nó phát hiện bức xạ hồng ngoại để phát hiện xem có lửa hay không.

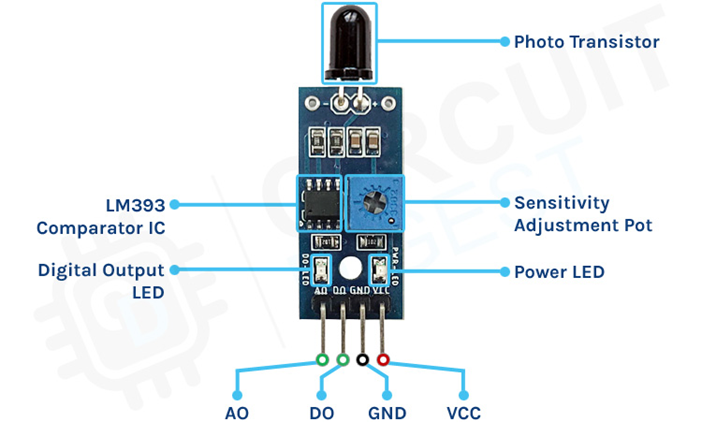


Hình 2.5 : Cảm biến phát hiện ngọn lửa

Một số thông số kỹ thuật của cảm biến như sau:

* **Nguồn cấp:** 3.3V - 5VDC
* **Dòng tiêu thụ:** 15mA
* **Tín hiệu ra:** Digital 3.3 - 5VDC tùy nguồn cấp hoặc Analog.
* **Khoảng cách:** 80 cm
* **Góc quét:** 60 độ
* **Kích thước:** 3.2 x 1.4 cm
* **Bước sóng:** 760nm - 1100nm

Sơ đồ khối của thiết bị được thể hiện ở hình sau:



Hình 2.6: Các thành phần trên module cảm biến phát hiện lửa

**VCC:** Chân nguồn dương VCC cấp nguồn từ 5VDC

**GND:** Chân nguồn âm

**D0:** Chân ngõ ra Digital

**A0:** Chân ngõ ra Analog

Ngoài ra, còn có một số linh kiện khác trên module như sau:

**Photo Transitor:** Cảm biến hồng ngoại YG 1006, dùng để phát hiện ra ngọn lửa.

**IC LM393**: IC so sánh điện áp cho đầu ra Digital

**Biến trở cảm biến**: Điều chỉnh độ nhạy của cảm biến

**LED nguồn**: Cho biết cảm biến đang hoạt động không

**LED cho đầu ra Digital**: Cho biết ra đầu ra của cảm biến

Với những thông số trên, ta thấy cảm biến có thể phát hiện ngọn lửa ở khoảng cách trung bình, phù hợp với môi trường trong phòng. Ngoài ra, sử dụng cảm biến hồng ngoại để phát hiện bức xạ giúp cho việc nhận diện ngọn lửa một cách chính xác.

Trên thị trường, module này có giá khoảng 10 nghìn đồng, mức giá rẻ và hợp lý cho giải pháp của chúng em.

**2.3 Khối cơ cấu chấp hành**

**2.3.1 Động cơ servo**

****

Hình 2.7: Động cơ servo SG90

Động cơ RC Servo 9G 0-180° có kích thước nhỏ, là loại được sử dụng nhiều nhất để làm các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng, động cơ RC Servo 9G có tốc độ phản ứng nhanh, các bánh răng được làm bằng nhựa nên cần lưu ý khi nâng tải nặng vì có thể làm hư bánh răng, động cơ RC Servo 9G có tích hợp sẵn Driver điều khiển động cơ bên trong nên có thể dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

**Thông số kỹ thuật của động cơ Servo SG90**

* Điện áp hoạt động: 4.8-5V DC
* Tốc độ quay: 0.12 giây/60 độ (ở 4.8V DC)
* Lực kéo tối đa: 1.6 kg.cm
* Kích thước vật lý: 21 x 12 x 22 mm
* Trọng lượng: 9g

**Cấu tạo và chức năng các chân kết nối**

Động cơ Servo SG90 sử dụng ba dây với chức năng như sau:

* **Dây đỏ:** Nguồn dương (VCC), được kết nối với nguồn 4.8-5V DC.
* **Dây đen:** Nguồn âm (GND), kết nối với đất (ground) của hệ thống.
* **Dây cam:** Tín hiệu (PWM), dùng để điều khiển góc quay của động cơ thông qua tín hiệu PWM từ vi điều khiển.

**Nguyên lý hoạt động và phương pháp điều khiển**

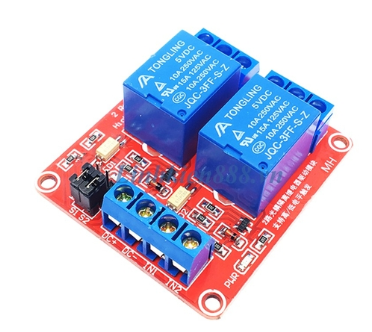
Động cơ Servo SG90 hoạt động dựa trên nguyên lý điều chế độ rộng xung (PWM). Tín hiệu PWM sẽ quyết định góc quay của động cơ trong khoảng từ 0 đến 180 độ.

* Độ rộng xung:
* Xung có độ rộng 1ms: Động cơ quay về góc 0 độ.
* Xung có độ rộng 2ms: Động cơ quay đến góc 180 độ.
* Các giá trị xung ở giữa (1-2ms) sẽ tương ứng với các góc quay từ 0 đến 180 độ.
* Chu kỳ PWM:
* Chu kỳ tín hiệu cố định là 20ms (tần số 50Hz).
* Độ rộng xung trong khoảng 1ms đến 2ms được lặp lại trong mỗi chu kỳ để duy trì góc quay của động cơ.

Động cơ Servo SG90 là một lựa chọn lý tưởng cho các ứng dụng nhỏ gọn, yêu cầu điều chỉnh góc chính xác và phản hồi nhanh. Với thiết kế tích hợp sẵn driver, động cơ này dễ dàng được sử dụng trong các dự án nhúng, robot và mô hình điều khiển từ xa. Nó phù hợp để điều khiển đóng mở cửa khi phát hiện rò rỉ khí gas.

**2.3.2 Module relay**

Module relay 2 kênh 5V DC là một thiết bị điện tử giúp điều khiển các thiết bị có dòng và điện áp cao từ tín hiệu điều khiển mức thấp của các vi điều khiển hoặc vi xử lý. Với hai relay được tích hợp sẵn trên board, module này cho phép điều khiển đồng thời hai thiết bị độc lập. Đặc biệt, module được thiết kế để kích hoạt relay ở mức tín hiệu thấp (Active Low) và có tích hợp đèn LED báo trạng thái cho từng kênh, giúp người dùng dễ dàng theo dõi trạng thái hoạt động.

****

Hình 2.8: Module relay 2 kênh

**Thông số kỹ thuật của module relay 2 kênh**

* Điện áp hoạt động: 5V DC
* Điện áp điều khiển:
* Dòng tải tối đa mỗi kênh: 10A
* Điện áp tải tối đa: AC: 250V và DC: 30V
* Loại relay: Relay kích mức thấp (Active Low)
* Opto-Isolator 817C
* Đèn LED báo trạng thái.

**Cấu tạo và chức năng các thành phần**

Module relay bao gồm các thành phần chính:

* **Relay cơ học:** Có khả năng chuyển mạch giữa hai trạng thái (NO - Normally Open và NC - Normally Closed) để điều khiển thiết bị tải.
* **Opto-Isolator 817C:** Cách ly điện giữa tín hiệu điều khiển và mạch tải, bảo vệ vi điều khiển khỏi nhiễu điện áp hoặc dòng cao từ tải.
* **Mạch điều khiển:** Bao gồm các linh kiện điện tử (transistor, diode, điện trở) để đảm bảo tín hiệu điều khiển hoạt động chính xác.

**Chân kết nối:**

* IN1, IN2: Chân điều khiển tín hiệu cho relay 1 và relay 2.
* VCC, GND: Cấp nguồn cho module.
* NO, NC, COM: Các chân kết nối tải của từng relay: NO (Normally Open): Trạng thái hở khi relay không được kích hoạt. NC (Normally Closed): Trạng thái đóng khi relay không được kích hoạt. COM: Chân chung của relay.

**Nguyên lý hoạt động của module**

Khi tín hiệu điều khiển (IN1 hoặc IN2) được đưa xuống mức logic 0 (0V), relay tương ứng sẽ được kích hoạt. Điều này làm thay đổi trạng thái từ NO sang NC, cho phép dòng điện đi qua mạch tải.

Module relay 2 kênh 5V DC là một giải pháp đơn giản và hiệu quả để điều khiển các thiết bị điện gia dụng và công nghiệp. Với thiết kế tích hợp opto-isolator, module này đảm bảo an toàn và dễ dàng tích hợp vào các hệ thống điều khiển. Mục đích dùng để điều khiển động cơ bơm nước và quạt gió khi phát hiện

**2.3.3 Động cơ bơm nước**



Hình 2.9: Động cơ bơm chìm mini 3-5v

**Thông số kỹ thuật**

* Điện áp sử dụng: 3~5VDC.
* Dòng điện sử dụng: 100~200mA.
* Lưu lượng bơm: 1.2~1.6L / 1 phút.
* Đường kính ngoài ống dẫn: 7.5mm
* Kích thước: 34 x 43 mm
* Trọng lượng: 28g

**Nguyên lý hoạt động:**

Máy bơm chìm hoạt động theo **nguyên tắc của lực ly tâm**, theo đó, lưu chất (nhất là chất lỏng) sẽ được hướng vào tâm quay của cánh máy bơm, từ đó, đẩy các chất lỏng lên trên mặt đất. Lúc này, bánh công tác được kích hoạt. Lực ly tâm sẽ tác động đẩy nước văng theo hướng từ tâm quay ra cánh của mép máy bơm, rồi đi theo máng dẫn để vào ống đẩy nhờ áp suất cao hơn.

**2.3.4 Quạt gió**

Quạt gió 5VDC là một thiết bị sử dụng nguồn điện một chiều 5V để tạo ra luồng gió, giúp làm mát hoặc thông gió trong các hệ thống điện tử, máy móc hoặc các ứng dụng cần điều chỉnh nhiệt độ hoặc lưu thông không khí. Với điện áp hoạt động là 5VDC. Quạt gió 5VDC rất phổ biến trong các ứng dụng cần làm mát các thiết bị điện tử hoặc các không gian nhỏ mà không muốn sử dụng các loại quạt điện cao cấp hoặc phức tạp.



Hình 2.10: Quạt gió (5VDC)

**Cấu tạo và nguyên lý hoạt động**

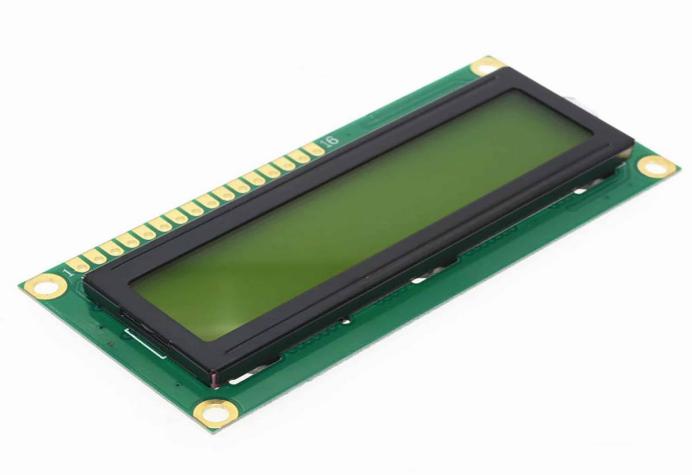
Quạt gió 5VDC gồm các thành phần cơ bản sau:

* **Mô-tơ điện một chiều:** Đây là bộ phận chính của quạt, giúp chuyển đổi năng lượng điện thành năng lượng cơ học. Mô-tơ hoạt động khi có dòng điện một chiều chạy qua, tạo ra lực quay và làm quay cánh quạt.
* **Cánh quạt:** Cánh quạt giúp tạo ra luồng gió khi được mô-tơ quay. Các cánh quạt có thể có các hình dạng và kích thước khác nhau tùy thuộc vào công suất và ứng dụng của quạt.
* **Mạch điện điều khiển:** Các quạt gió 5VDC thường có một mạch điều khiển giúp kiểm soát tốc độ quay của mô-tơ và đảm bảo quạt hoạt động ổn định.
* **Vỏ quạt:** Vỏ bảo vệ mô-tơ và cánh quạt, giúp đảm bảo an toàn và bảo vệ các bộ phận bên trong khỏi bụi bẩn và hư hỏng.

Nguyên lý hoạt động của quạt gió 5VDC dựa trên hiện tượng từ trường. Khi có dòng điện một chiều chạy qua cuộn dây của mô-tơ, từ trường được tạo ra, khiến rotor (phần quay của mô-tơ) quay. Quá trình quay của rotor làm quay các cánh quạt, tạo ra luồng không khí giúp làm mát hoặc thông gió.

**2.4 Khối hiển thị**

Màn hình LCD (Liquid Crystal Display) là một loại màn hình phẳng sử dụng công nghệ tinh thể lỏng để hiển thị thông tin. Đây là công nghệ phổ biến trong nhiều thiết bị điện tử, từ máy tính, tivi, điện thoại di động. Màn hình LCD nổi bật với ưu điểm tiêu thụ ít điện năng, độ phân giải tốt và khả năng hiển thị rõ ràng trong nhiều điều kiện ánh sáng.



Hình 2.11: Màn hình LCD 16x2

**Cấu tạo và nguyên lý hoạt động:**

Màn hình LCD hoạt động dựa trên việc kiểm soát ánh sáng thông qua các tinh thể lỏng. Các lớp cấu tạo chính bao gồm:

* **Lớp kính phân cực:** Giúp định hướng ánh sáng trước và sau khi đi qua màn hình.
* **Lớp tinh thể lỏng:** Tinh thể lỏng nằm giữa hai lớp kính phân cực, có thể thay đổi hướng ánh sáng khi được điều khiển bởi điện áp.
* **Lớp điện cực:** Các điện cực trong màn hình tạo ra điện trường để kiểm soát hướng của tinh thể lỏng.
* **Đèn nền (Backlight):** Đối với LCD hiện đại, đèn LED thường được sử dụng làm nguồn sáng, giúp tăng độ sáng và tiết kiệm năng lượng.

` Nguyên lý hoạt động của LCD dựa trên hiện tượng ánh sáng bị phân cực. Khi điện áp được áp dụng, các tinh thể lỏng thay đổi góc xoắn của chúng, điều chỉnh lượng ánh sáng đi qua và tạo ra các điểm sáng hoặc tối trên màn hình.

**Thông số kỹ thuật quan trọng**

* Điện áp làm việc: 3.3-5V
* Độ phân giải: 16x2
* Chế độ giao tiếp: SPI, I2C
* Kích thước vật lý: 1.8 inch

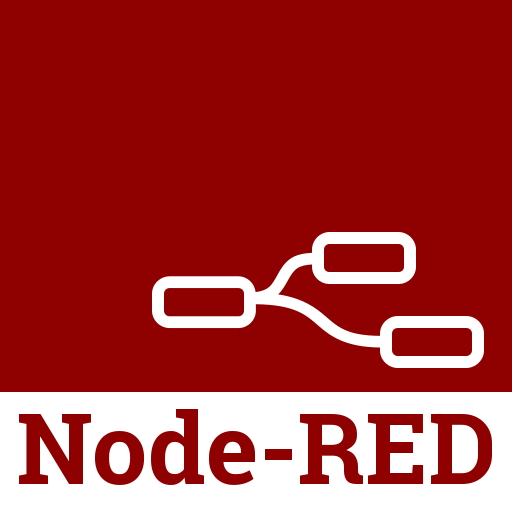
Màn hình LCD là một thành phần quan trọng và phổ biến trong các ứng dụng nhúng và thiết bị điện tử hiện đại. Với sự đa dạng về chủng loại và tính năng, màn hình LCD đáp ứng được nhu cầu từ các ứng dụng đơn giản đến phức tạp.

**2.5 Phần mềm giám sát và điều khiển**

Trong một hệ thống IoT, ngoài các thiết bị cảm biến để thu thập dữ liệu, các thiết bị cơ cấu chấp hành để điều khiển, thiết lập thông số các nói chung, chúng ta có cần phải có một phần mềm, ứng dụng giúp ta có thể giám sát, điều khiển thiết bị từ xa mà không cần phải làm trực tiếp trên chúng. Với sự phổ biến của điện thoại thông minh và máy tính trong đời sống con người, cùng với sự phát triển của các phần mềm ứng dụng hiện nay, chúng ta hoàn toàn có thể xây dựng một phần mềm giám sát và điều khiển thiết bị từ xa, giúp cho người dùng dễ dàng nắm bắt thông tin, xử lý sự cố một cách nhanh chóng và tiện lợi.

Như đã nói ở trên, phần mềm của người dùng phải có những chức năng đáp ứng với các yêu cầu trên, đó là có thể trực quan hoá dữ liệu theo thời gian thực của cảm biến, điều khiển hoạt động của các thiết bị cơ cấu chấp hành, phần mềm dễ sử dụng và thân thiện với người dùng. Để thực hiện điều đó, chúng em đã tìm hiểu và thấy rằng có rất nhiều nền tảng, phần mềm IoT mã nguồn mở phục vụ cho những nhà phát triển, người dùng IoT. Một số nền tảng IoT phổ biến như Blynk IoT, Things Board, Microsoft Azure IoT, Amazon Web Services IoT,…

Đối với nhóm chúng em, chúng em lựa chọn nền tảng IoT là Node-Red cho hệ thống IoT. Node-RED là một công cụ mã nguồn mở và trực quan được sử dụng để xây dựng các luồng làm việc (workflows) và ứng dụng Internet of Things (IoT). Nó cung cấp một giao diện đồ họa dựa trên trình duyệt web, cho phép người dùng kết nối các nút (node) với nhau để xử lý dữ liệu và tương tác với các thiết bị và dịch vụ khác nhau. Node-RED được xây dựng trên nền tảng Node.js và sử dụng trình duyệt web để tạo ra một giao diện dễ sử dụng. Người dùng có thể kéo và thả các nút từ thư viện có sẵn để tạo ra luồng làm việc theo ý muốn. Các nút có thể thực hiện các nhiệm vụ khác nhau, bao gồm xử lý dữ liệu, kết nối và tương tác với các dịch vụ web, cơ sở dữ liệu, thiết bị IoT và nhiều hơn nữa.



Hình 2.12: Nền tảng Node-Red

Một số tính năng chính của chúng bao gồm:

* **Giao diện trực quan:** Node-RED cung cấp một giao diện đồ họa dễ sử dụng dựa trên trình duyệt web. Người dùng có thể kéo thả các nút để tạo và kết nối các luồng làm việc một cách trực quan.
* **Các nút và luồng đa dạng:** Node-RED đi kèm với một bộ sưu tập nút mở rộng lớn, cho phép bạn tương tác với các thiết bị, dịch vụ và giao thức khác nhau. Các nút này bao gồm MQTT, HTTP, cơ sở dữ liệu, xử lý dữ liệu, gửi email, gửi tin nhắn và nhiều hơn nữa.
* **Tích hợp IoT:** Node-RED được phát triển đặc biệt cho việc xây dựng ứng dụng IoT. Bạn có thể tương tác với các thiết bị như Raspberry Pi, ESP32, các cảm biến và các nền tảng IoT khác.
* **Xử lý dữ liệu linh hoạt:** Node-RED cho phép bạn xử lý, biến đổi và lọc dữ liệu theo ý muốn. Nó thực hiện các hoạt động như chuyển đổi định dạng, phân tích văn bản, tính toán, lưu trữ và đồng bộ dữ liệu.
* **Giao tiếp thời gian thực:** Node-RED hỗ trợ giao tiếp thời gian thực và khả năng phát hiện sự kiện.



Hình 2.13 : Giao diện người dùng của Node-Red

Ngoài ra, nhóm chúng em còn sử dụng giao thức kết nối MQTT cho việc trao đổi dữ liệu giữa phần mềm và thiết bị phần cứng. MQTT (Message Queueing Telemetry Transport) là một giao thức mạng kích thước nhỏ (lightweight), hoạt động theo cơ chế publish – subscribe (tạm dịch: xuất bản – đăng ký) theo tiêu chuẩn ISO (ISO/IEC 20922) và OASIS mở để truyền tin nhắn giữa các thiết bị.



Hình 2.14: Giao thức MQTT trong IoT

Giao thức này hoạt động trên nền tảng TCP/IP. MQTT được thiết kế cho các kết nối cho việc truyền tải dữ liệu cho các thiết bị ở xa, các thiết bị hay vi điều khiển nhỏ có tài nguyên hạn chế hoặc trong các ứng dụng có băng thông mạng bị hạn chế.

MQTT có nhiều lợi ích vượt trội trong hệ thống IoT như tính gọn nhẹ, hiệu quả, phù hợp các thiết bị IoT với tài nguyên thấp, tiết kiệm dữ liệu và dung lượng mạng, hỗ trợ kết nối đến một số lượng lớn thiết bị IoT, đảm bảo độ tin cậy trong truyền thông và hỗ trợ rộng rãi trong cộng đồng phát triển với nhiều thư viện và framework có sẵn.

**2.6 Các thiết bị khác**

**2.6.1 Khối nguồn**

Khối nguồn là một thành phần không thể thiếu trong bất kỳ hệ thống nhúng nào, đảm bảo cung cấp năng lượng ổn định và phù hợp cho toàn bộ hệ thống hoạt động.

Khối nguồn sử dụng là Adapter 5V 2A đểcấp nguồn điện áp 5V ổn định cho các linh kiện nhạy cảm cần nguồn chính xác như vi điều khiển, cảm biến, relay và màn hình LCD,v.v.



Hình 2.15: Nguồn ADAPTER 5VDC 2A

**Hoạt động của khối nguồn trong hệ thống**

Adapter 5V 2A được kết nối trực tiếp với các chân nguồn của ESP32 và các linh kiện yêu cầu điện áp 5V. Điện áp từ adapter này được sử dụng trực tiếp mà không cần qua thêm các mạch hạ áp.

**2.6.2 Các nút nhấn (SW1, SW2, SW3, SW4)**

Nút nhấn là một linh kiện điện tử rất phổ biến trong các hệ thống điện và IoT. Nó hoạt động như một công tắc tạm thời, cho phép người dùng điều khiển thiết bị bằng cách nhấn tay.Mỗi nút nhấn thường đi kèm một điện trở kéo (pull-up hoặc pull-down) 10kΩ. Khi người dùng nhấn nút, mức logic trên chân GPIO của ESP32 thay đổi (từ HIGH sang LOW hoặc ngược lại), vi điều khiển có thể đọc được sự kiện nhấn.

**2.6.3 Mạch in PCB**

Mạch PCB là viết tắt củaPrinted Circuit Board (bảng mạch in). Đây là một thành phần rất quan trọng trong hầu hết các thiết bị điện tử. Mạch PCB dùng để kết nối và gắn kết các linh kiện điện tử thông qua cácđường dẫn dẫn điện (mạch in) được khắc trên một lớp vật liệu nền (thường là sợi thủy tinh hoặc vật liệu cách điện khác).

Lý do nhóm chọn thiết kế PCB thay vì board có sẵn vì mạch in pcb tích hợp nhiều kết nối: Hệ thống IoT thường yêu cầu kết nối đa dạng (cảm biến, module truyền thông, nguồn, vi điều khiển). PCB cho phép thiết kế các đường mạch chuyên biệt, tránh tình trạng dây nhảy rối như trên breadboard hoặc mạch prototype, giảm nguy cơ nhiễu tín hiệu và đứt kết nối. Nó giúp tích hợp tất cả thành một board duy nhất, tối ưu hóa đường đi tín hiệu và nguồn. PCB thiết kế tùy chỉnh giúp giảm kích thước hệ thống, phù hợp với ứng dụng IoT cần đóng gói nhỏ gọn PCB cho phép kiểm soát impedance, phân tách ground plane, và bố trí linh kiện hợp lý để giảm nhiễu điện từ (EMI), đặc biệt quan trọng với tín hiệu analog (cảm biến) và tần số cao (Wi-Fi, Bluetooth)..

**2.6.4 Khung dựng sản phẩm giả định**

Khung dựng sản phẩm giả định nhóm sử dụng tấm nhựa mica, một vật liệu phổ biến sử dụng trong các sản phẩm Iot. Tấm mica dùng để dựng mô hình IoT là một loại vật liệu nhựa trong suốt hoặc mờ, thường được dùng để làm khung, vỏ hoặc giá đỡ cho các linh kiện điện tử trong các dự án IoT, Arduino, Raspberry Pi, robot, v.v.Tên gọi phổ biến là mica, nhưng tên kỹ thuật là Acrylic (PMMA) – viết tắt của Polymethyl Methacrylate. Đặc điểm nhẹ hơn kính nhưng vẫn cứng cáp trong suốt hoặc có thể là màu mờ, đen, trắng tùy theo nhu cầu. Dễ cắt, khoan, khắc bằng laser để làm lỗ gắn linh kiện. Chịu lực, chịu nhiệt nhẹ và không dẫn điện.

Lý do nhóm chọn tấm mica làm vật liệu dựng khung sản phẩm là mica dễ cắt thành các chi tiết phức tạp như cửa sổ, mái nhà, tường, phù hợp để mô phỏng kiến trúc ngôi nhà mini. Mica là vật liệu không dẫn điện, phù hợp với hệ thống IoT có nhiều linh kiện điện tử (PCB, dây nối), tránh rủi ro chập cháy do tiếp xúc với khung kim loại. Mica chịu được nhiệt độ lên đến 80–90°C, phù hợp với nhiệt độ tỏa ra từ linh kiện điện tử trong nhà mini.

**2.7 Đánh giá chi phí và khả năng thực hiện**

Đánh giá về tổng chi phí và khối lượng công việc:  
 Để triển khai hệ thống IoT cảnh báo khí gas và phòng chống cháy nổ, nhóm đã lựa chọn các linh kiện điện tử phổ biến, dễ tiếp cận trên thị trường và có chi phí hợp lý, nhằm đảm bảo cân đối giữa hiệu năng và giá thành. Tổng chi phí để hoàn thiện một hệ thống hoàn chỉnh dao động trong khoảng 550.000 – 700.000 VNĐ, bao gồm cả chi phí linh kiện, gia công mạch và vật tư phụ.

Trong đó, linh kiện cốt lõi là vi điều khiển ESP32 (ESP-WROOM-32S) – đóng vai trò xử lý trung tâm – có giá khoảng 80.000 đến 100.000 VNĐ. Các cảm biến như MQ-2 (phát hiện khí gas) và cảm biến lửa có giá từ 30.000 đến 40.000 VNĐ mỗi loại. Màn hình LCD 16x2 phục vụ hiển thị thông tin trạng thái hệ thống có giá khoảng 50.000 VNĐ. Một số thiết bị đầu ra như động cơ servo SG90, còi hú và quạt gió cũng góp phần đáng kể vào tổng chi phí, trung bình từ 30.000 đến 60.000 VNĐ mỗi linh kiện.

Đặc biệt, một khoản chi phí không thể thiếu là gia công bảng mạch, được thực hiện với giá khoảng 200.000 VNĐ. Việc đầu tư gia công mạch giúp đảm bảo độ bền cơ khí, khả năng cố định linh kiện tốt hơn, giảm thiểu hiện tượng kết nối lỏng lẻo khi vận hành thực tế. Đây là bước cần thiết nếu muốn đưa sản phẩm từ mức nguyên mẫu (prototype) sang sử dụng lâu dài hoặc thử nghiệm trong môi trường thực tế.

Ngoài linh kiện và mạch, hệ thống còn sử dụng các vật tư phụ như dây nối, điện trở, tụ, đầu nối, vít, vỏ nhựa, PCB và vật liệu cách điện. Dù chi phí không lớn (khoảng 50.000 – 100.000 VNĐ), nhưng đây là phần không thể thiếu để hệ thống được lắp đặt gọn gàng, an toàn và dễ bảo trì.

Về khối lượng công việc, hệ thống được phân chia thành nhiều phần rõ ràng như phần cứng, phần mềm, lắp ráp và thử nghiệm. Phần cứng tương đối đơn giản nhờ các linh kiện có sẵn dạng module, hạn chế tối đa việc hàn linh kiện thủ công. Phần mềm sử dụng Arduino IDE – môi trường lập trình thân thiện với người học, giúp nhóm nhanh chóng triển khai chương trình điều khiển vi điều khiển ESP32, giao tiếp với cảm biến và truyền tín hiệu cảnh báo về nền tảng Node-Red

Toàn bộ quá trình tích hợp và thử nghiệm diễn ra suôn sẻ nhờ việc thiết kế hệ thống có cấu trúc rõ ràng, tài liệu tham khảo đầy đủ và sự phối hợp hiệu quả giữa các thành viên nhóm. Các công việc đều được triển khai theo từng bước, đảm bảo độ chính xác và dễ dàng sửa lỗi.

Tổng quan cho thấy, chi phí đầu tư cho hệ thống ở mức thấp đến trung bình, hoàn toàn phù hợp với sinh viên và các dự án nghiên cứu quy mô nhỏ. Mặt khác, khối lượng công việc phân bổ hợp lý, không đòi hỏi kỹ năng chuyên sâu về phần cứng hay lập trình phức tạp, nên rất khả thi để thực hiện trong thời gian ngắn với nhóm nhỏ.

Đánh giá về thời gian và khả năng thực hiện:

Tính từ thời điểm bắt đầu nghiên cứu đến khi hoàn thiện mô hình và báo cáo, nhóm đã trải qua đầy đủ các giai đoạn từ thiết kế ý tưởng, lựa chọn linh kiện, lắp ráp, lập trình, đến kiểm thử và hiệu chỉnh. Mỗi giai đoạn đều được thực hiện trong khung thời gian tương đối ngắn, nhờ áp dụng cách tiếp cận từng bước và luôn thử nghiệm sau mỗi phần hoàn thành.

* Thời gian nghiên cứu và chuẩn bị linh kiện kéo dài 3 ngày. Trong giai đoạn này, nhóm tìm hiểu về các module cảm biến, phương pháp kết nối với ESP32, và lựa chọn các linh kiện phù hợp.
* Thời gian thiết kế và gia công mạch mất 4 ngày, bao gồm cả thời gian thiết kế sơ đồ mạch, bố trí linh kiện trên PCB và đặt dịch vụ gia công.
* Giai đoạn lắp ráp phần cứng và kiểm tra kết nối thực tế mất 3 ngày.
* Phần lập trình và hiệu chỉnh kéo dài 5 ngày, trong đó đáng kể nhất là quá trình xử lý tín hiệu từ cảm biến, điều kiện kích hoạt thiết bị cảnh báo và gửi dữ liệu về phần mềm Node-Red
* Giai đoạn thử nghiệm, hoàn thiện mô hình và viết báo cáo chiếm khoảng 3 ngày.

Tổng cộng, thời gian để hoàn thiện toàn bộ dự án dao động trong 18 ngày, phù hợp với kế hoạch bài tập lớn học kỳ.

Về khả năng thực hiện, hệ thống được đánh giá là có tính khả thi cao, không chỉ trong khuôn khổ môn học mà còn trong ứng dụng thực tế. Với khả năng phát hiện sớm rò rỉ khí gas và phát hiện lửa, hệ thống có thể triển khai trong môi trường hộ gia đình, nhà bếp, nhà trọ hoặc các khu xưởng nhỏ. Việc ứng dụng công nghệ IoT giúp người dùng có thể giám sát và nhận cảnh báo từ xa thông qua điện thoại, nâng cao độ an toàn và giảm thiểu thiệt hại.

**CHƯƠNG 3: THI CÔNG HỆ THỐNG VÀ PHẦN MỀM**

**3.1 Thiết kế phần cứng hệ thống**

***3.1.1. Sơ đồ nguyên lý***

**A diagram of a computer

AI-generated content may be incorrect.**

Hình 3.1: sơ đồ nguyên lý hoạt động

Dựa trên sơ đồ nguyên lý được chia thành 4 khối: **Khối nguồn**, **Khối đầu vào**, **Khối xử lý** và **Khối đầu ra**, dưới đây là phân tích chi tiết về vai trò của từng khối, cách các linh kiện liên kết và nguyên lý hoạt động tổng thể của mạch.

**a. Khối Nguồn**

**Nguồn cung cấp (5V)**

* Thường được lấy từ adapter 5V.
* Cung cấp điện cho các linh kiện cần 5V như LCD, servo, relay (nếu có), v.v.

**Tụ lọc**

* Được đặt ở đầu vào/đầu ra của bộ ổn áp để giảm nhiễu, giúp điện áp ra ổn định.
* **Chức năng:** Khối nguồn đảm bảo cung cấp đầy đủ và ổn định các mức điện áp cần thiết (5V và 3.3V) cho toàn bộ mạch hoạt động.

**b. Khối Đầu Vào**

Trong hình, khối đầu vào có thể bao gồm **các nút nhấn (button)** và **chân header** dành cho cảm biến.

**Các nút nhấn (SW1, SW2, SW3, SW4)**

* Mỗi nút nhấn thường đi kèm một điện trở kéo (pull-up hoặc pull-down) 10kΩ.
* Khi người dùng nhấn nút, mức logic trên chân GPIO của ESP32 thay đổi (từ HIGH sang LOW hoặc ngược lại), vi điều khiển có thể đọc được sự kiện nhấn.

**Header (HD2) cho Cảm biến**

* Trong nhiều thiết kế, header này dùng để cắm cảm biến khí gas (MQ-2)hoặc cảm biến lửa (Flame Sensor).
* Cảm biến thường được cấp nguồn 5V (hoặc 3.3V, tuỳ loại). Tín hiệu đầu ra của cảm biến (AO/DO) được đưa về chân ADC hoặc GPIO của ESP32.
* Nếu là cảm biến analog, ta đưa về chân ADC của ESP32. Nếu là cảm biến digital, ta đưa về chân GPIO đọc mức logic.

**Chức năng:** Khối đầu vào thu thập tín hiệu từ môi trường (cảm biến) và từ người dùng (nút nhấn), sau đó chuyển các tín hiệu này về khối xử lý (ESP32) để phân tích.

c**. Khối Xử Lý**

Ở sơ đồ, khối xử lý chính là **vi điều khiển ESP32**.

**ESP32 Module**

* Tích hợp Wi-Fi, Bluetooth, CPU 32-bit, bộ nhớ Flash, SRAM.
* Chân cấp nguồn: 3.3V, GND, EN (enable).
* Các chân GPIO: dùng cho nhiều chức năng (ADC, PWM, I2C, SPI, UART…).
* Trong sơ đồ, các chân này được kết nối tới:
* **Nút nhấn** (nhận tín hiệu digital).
* **Cảm biến** (analog hoặc digital).
* **Thiết bị đầu ra** (LCD, servo, …).

**Chức năng:** Vi điều khiển ESP32 đóng vai trò “bộ não” của toàn hệ thống, xử lý các tín hiệu đầu vào và đưa ra quyết định điều khiển khối đầu ra.

**d. Khối Đầu Ra**

Dựa vào sơ đồ, khối đầu ra bao gồm **LCD**, **buzzer**, **servo** ,**Relay hoặc Transistor**

**LCD 16x2**

* Kết nối ở chế độ 4-bit (D4, D5, D6, D7) hoặc 8-bit.
* Các chân RS, E (Enable), RW (thường nối GND), VCC, GND.
* LCD được cấp nguồn 5V và có thể giao tiếp ở mức logic 5V hoặc 3.3V

**Servo**

* Servo (SG90, MG995...) thường hoạt động ở điện áp 5V.
* Chân điều khiển servo (tín hiệu PWM) được nối với 1 chân GPIO của ESP32.

**Relay hoặc Transistor**

* Điều khiển các thiết bị công suất cao, bật/tắt bơm, quạt, van điện từ...
* Relay cần transistor kích, diode bảo vệ, điện trở kéo.

**Chức năng:** Khối đầu ra chịu trách nhiệm hiển thị thông tin (LCD), điều khiển cơ cấu chấp hành (servo, relay...) dựa trên quyết định từ vi điều khiển ESP32.

**Nguyên Lý Hoạt Động Tổng Quát**

**Cấp nguồn**

* Mạch nhận nguồn 5V từ adapter hoặc USB.
* Nếu ESP32 chưa có sẵn bộ chuyển đổi, IC ổn áp tạo ra 3.3V cho ESP32.

**Khởi tạo và đọc tín hiệu đầu vào**

* ESP32 khởi động, thiết lập các chân GPIO cho nút nhấn, cảm biến.
* Đọc giá trị cảm biến (nồng độ khí gas, tín hiệu lửa…) và trạng thái nút nhấn.

**Xử lý thông tin**

* Chương trình bên trong ESP32 phân tích ngưỡng đo từ cảm biến (VD: nồng độ khí gas > ngưỡng an toàn).
* Nếu phát hiện vượt ngưỡng, kích hoạt cảnh báo hoặc điều khiển thiết bị liên quan.

**Xuất tín hiệu điều khiển**

* Gửi lệnh hiển thị lên LCD (thông báo “Cảnh báo Gas”).
* Điều khiển servo (đóng/mở van gas) hoặc relay (bật quạt hút).
* Gửi dữ liệu qua Wi-Fi/Bluetooth (nếu có lập trình).

**Giám sát và cập nhật liên tục**

* Hệ thống chạy vòng lặp, liên tục cập nhật trạng thái cảm biến.
* Người dùng có thể tương tác qua nút nhấn (tắt cảnh báo, reset hệ thống…).

**3.1.2 Thiết kế mạch PCB hệ thống**

**Phân tích mạch PCB đã được thiết kế.**

Để đảm bảo tính ổn định, gọn gàng và dễ lắp ráp cho hệ thống IoT cảnh báo khí gas và phòng chống cháy nổ, nhóm đã tiến hành thiết kế mạch in (PCB) với bố cục rõ ràng, khoa học. Hình ảnh trên thể hiện hai mặt của mạch PCB: mặt top (màu đỏ) và mặt bottom (màu xanh), cho thấy đây là mạch 2 lớp – một lựa chọn hợp lý cho hệ thống có số lượng linh kiện vừa phải nhưng vẫn yêu cầu định tuyến phức tạp.

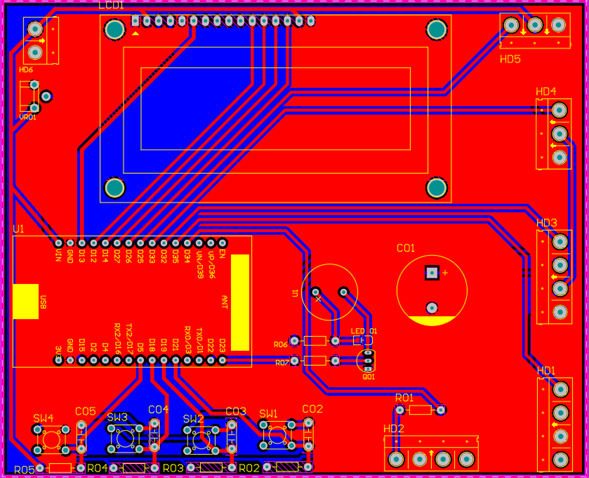
**Cấu trúc tổng thể của PCB**

PCB được thiết kế dưới dạng hình chữ nhật với các cạnh bo tròn nhẹ, phù hợp để lắp vào hộp nhựa hoặc khung bảo vệ tiêu chuẩn. Diện tích mạch đủ lớn để bố trí đầy đủ các thành phần chính của hệ thống, đồng thời vẫn đảm bảo khoảng cách an toàn điện và chống nhiễu hiệu quả.

Mạch được chia thành các khu vực chức năng rõ ràng, tương ứng với các khối của hệ thống:

* Khu vực trung tâm: là nơi bố trí vi điều khiển ESP-WROOM-32S, với các đường tín hiệu đi ra các chân GPIO được định tuyến rõ ràng.
* Góc dưới bên trái: là khu vực bố trí 4 nút nhấn, cho phép người dùng điều chỉnh chế độ hệ thống. Mỗi nút đi kèm với điện trở kéo và có đánh số RO1 – RO4 tương ứng.
* Góc phải mạch: là vị trí các header giao tiếp với các module ngoại vi, như cảm biến khí gas, cảm biến lửa, còi hú, quạt gió...
* Phía trên mạch: bố trí header cho màn hình LCD 16x2 – một thành phần quan trọng dùng để hiển thị trạng thái hệ thống.

**Phân tích mặt top (màu đỏ)**



Hình 3.2: lớp Top

Mặt top chứa phần lớn các đường mạch dẫn nguồn và tín hiệu chính, với thiết kế ưu tiên độ rộng đường nguồn lớn, đặc biệt là các đường cấp nguồn cho thiết bị tiêu thụ dòng cao như quạt và còi. Một số điểm đáng chú ý:

* ESP32 được gắn ở trung tâm, các chân GPIO được bố trí hợp lý, giảm thiểu giao cắt tín hiệu.
* Các tụ lọc và điện trở được bố trí gần các chân cấp nguồn để giảm nhiễu và ổn định điện áp.
* Lỗ khoan bắt vít hoặc chân định vị được bố trí ở các góc, giúp cố định mạch chắc chắn khi lắp vào hộp.

**Phân tích mặt bottom (màu xanh)**

A blue circuit board with red and yellow lines

AI-generated content may be incorrect.  
 Hình 3.3: Lớp Bottom

Mặt bottom chủ yếu để định tuyến các tín hiệu còn lại, giúp giảm tắc nghẽn tín hiệu ở mặt top. Nhờ sử dụng PCB 2 lớp, các đường tín hiệu không cần chồng chéo phức tạp, góp phần tăng độ tin cậy cho hệ thống. Ngoài ra:

* Một số đường tín hiệu quan trọng như đường điều khiển relay, tín hiệu PWM cho servo và đường dữ liệu từ cảm biến được định tuyến qua mặt này để tránh nhiễu.
* Mặt bottom chứa các ký hiệu linh kiện giúp dễ dàng nhận biết khi lắp ráp thủ công.

**Đặc điểm kỹ thuật và chất lượng thiết kế**

* Khoảng cách giữa các đường mạch và lỗ pad được bố trí hợp lý, đảm bảo tiêu chuẩn an toàn cho mạch hoạt động ở mức 3.3V và 5V.
* Các đường mạch dẫn dòng lớn được thiết kế với chiều rộng đủ lớn để tránh quá nhiệt khi hoạt động lâu dài.
* Via (lỗ chuyển tầng) được sử dụng hiệu quả tại các giao điểm quan trọng, giúp kết nối hai lớp mạch mà không ảnh hưởng đến độ bền cơ học của PCB.

**Đánh giá tổng quan**

Mạch PCB được thiết kế cẩn thận, hợp lý, tối ưu cả về kỹ thuật lẫn thẩm mỹ. Việc sử dụng mạch 2 lớp cho phép:

* Tăng mật độ linh kiện.
* Đảm bảo định tuyến dễ dàng và logic.
* Hạn chế giao thoa tín hiệu và giảm nhiễu điện từ.

Bên cạnh đó, mạch còn có khả năng mở rộng nếu muốn bổ sung thêm cảm biến, mô-đun hoặc thiết bị điều khiển trong tương lai.

A green circuit board with a rectangular display

AI-generated content may be incorrect.

Hình 3.4: Mạch mô phỏng

**3.1.3 Viết phần mềm nhúng**

**- Lập trình đọc cảm biến với ESP32**

#define SENSOR\_MQ2 35 // Định nghĩa chân kết nối cảm biến

// Hàm đọc giá trị cảm biến khí gas MQ2

int readMQ2(){

float MQ2\_Value = analogRead(SENSOR\_MQ2);

MQ2\_Value = map(MQ2\_Value, 0 , 4095, 0, 10000 );

Serial.println(MQ2\_Value);

return MQ2\_Value;

}

# Định nghĩa chân kết nối cảm biến phát hiện lửa

#define SENSOR\_FIRE 34

void setup(){

pinMode(SENSOR\_FIRE, INPUT\_PULLUP); // Cấu hình chân kết nối

}

// Hàm xác định phát hiện có lửa hay không

int readFireSensor() {

int Fire\_Value = !digitalRead(SENSOR\_FIRE);

if(Fire\_Value == 1){

digitalWrite(BUZZER,BUZZER\_ON);

}

else {

digitalWrite(BUZZER,BUZZER\_OFF);

}

Serial.println(Fire\_Value);

return Fire\_Value;

}  
 - **Lập trình cơ cấu chấp hành**

// Định nghĩa nút nhấn điều khiển cơ cấu chấp hành

#define buttonPinFan 5

#define buttonPinPump 18

#define buttonPinDoor 19

#define buttonPinAutoMode 21

// Định nghĩa chân kết nối đến còi

#define BUZZER 23

#define BUZZER\_ON 1

#define BUZZER\_OFF 0

// Định chân kết nối relay và servo

#define RELAY1 22

#define RELAY2 32

#define SERVO1 33

#define SERVO2 25

// Định nghĩa trạng thái servo

bool relay1State = OFF;

bool relay2State = OFF;

int mq2Thresshold = 1000;

// Định nghĩa các thiết bị servo

Servo myservo1;

Servo myservo2;

// Cấu hình chân kết nối điều khiển relay

void setup(){

pinMode(RELAY1, OUTPUT);

pinMode(RELAY2, OUTPUT);

myservo1.setPeriodHertz(50);

myservo2.setPeriodHertz(50);

myservo1.attach(SERVO1, 500, 2400);

myservo2.attach(SERVO2, 500, 2400);

}

// điều khiển relay thông qua app

void callback(char\* topic, byte\* payload, unsigned int length) {

Serial.print("Nhận dữ liệu từ topic:");

Serial.println(topic);

char message[length + 1];

memcpy(message, payload, length);

message[length] = '\0'; // Thêm ký tự kết thúc chuỗi

StaticJsonDocument<200> doc;

DeserializationError error = deserializeJson(doc, message);

if(doc.containsKey("status")) {

int status = doc["status"];

Serial.print("Trạng thái nhận được: ");

Serial.println(status);

// Xử lý theo từng topic

if (strcmp(topic, "Devices/Fan") == 0) {

digitalWrite(RELAY1, status ? HIGH : LOW);

controlWindow(status);

Serial.println(status ? "Quạt BẬT" : "Quạt TẮT");

}

else if (strcmp(topic, "Devices/Pump") == 0) {

digitalWrite(RELAY2, status ? HIGH : LOW);

Serial.println(status ? "Máy bơm BẬT" : "Máy bơm TẮT");

}

}

}

// Điều khiển cửa đóng mở

void openWindow() {

myservo1.write(0);

myservo2.write(180);

}

void closeWindow() {

myservo1.write(90);

myservo2.write(90);

}

void controlWindow(int onoff) {

if(onoff == 0) closeWindow();

else openWindow();

}

// Dùng nút nhấn điều khiển các cơ cấu chấp hành

void TaskButton(void \*pvParameters) {

bool lastStateBTN1 = HIGH;

bool lastStateBTN2 = HIGH;

bool lastStateBTN3 = HIGH;

while (1) {

bool currentStateBTN1 = digitalRead(5);

bool currentStateBTN2 = digitalRead(19);

bool currentStateBTN3 = digitalRead(18);

if (currentStateBTN1 == LOW && lastStateBTN1 == HIGH) {

relay1State = !relay1State;

digitalWrite(RELAY1, relay1State ? HIGH : LOW);

digitalWrite(2, relay1State ? HIGH : LOW);

Serial.print("RELAY1: ");

Serial.println(relay1State ? "ON" : "OFF");

}

lastStateBTN1 = currentStateBTN1;

if (currentStateBTN2 == LOW && lastStateBTN2 == HIGH) {

relay2State = !relay2State;

digitalWrite(RELAY2, relay2State ? HIGH : LOW);

digitalWrite(2, relay1State ? HIGH : LOW);

Serial.print("RELAY2: ");

Serial.println(relay2State ? "ON" : "OFF");

}

lastStateBTN2 = currentStateBTN2;

if (currentStateBTN3 == LOW && lastStateBTN3 == HIGH) {

window = !window;

controlWindow(window);

}

lastStateBTN3 = currentStateBTN3;

vTaskDelay(pdMS\_TO\_TICKS(50));

}

}

**3.1.4 Lắp đặt phần cứng**

Việc lắp đặt phần cứng đóng vai trò quan trọng trong quá trình hiện thực hóa hệ thống, từ bản thiết kế điện tử đến mô hình vật lý hoàn chỉnh. Trong phần này, nhóm đã tiến hành hai công đoạn chính: ghép nối các linh kiện lên mạch PCB và dựng mô hình giả định hệ thống cảnh báo khí gas và phòng cháy chữa cháy tự động.

Ghép nối các linh kiện lên mạch PCB

Sau khi hoàn tất thiết kế và gia công mạch in (PCB), nhóm tiến hành bước lắp ráp phần cứng bằng cách hàn thủ công các linh kiện điện tử vào bảng mạch. Quy trình thực hiện được tiến hành tuần tự như sau:

* **Chuẩn bị linh kiện và công cụ**: Nhóm chuẩn bị đầy đủ các linh kiện điện tử bao gồm vi điều khiển ESP32, các linh kiên điện tử, các cảm biến, màn hình LCD, cùng động cơ servo và các thiết bị điều khiển ngoại vi. Ngoài ra, các công cụ như mỏ hàn, thiếc hàn, kìm cắt, đồng hồ đo điện và hút thiếc cũng được sử dụng để đảm bảo chất lượng hàn tốt nhất.
* **Lắp đặt theo thứ tự hợp lý:** Nhóm bắt đầu bằng việc hàn các linh kiện nhỏ và sát mạch. Tiếp theo là các chân cắm header, relay, màn hình LCD và cuối cùng là các module cảm biến.
* **Đảm bảo kết nối chính xác:** Sau khi hàn xong từng thành phần, nhóm sử dụng đồng hồ đo điện (multimeter) để đo đạc tính liên thông giữa các chân kết nối, đảm bảo không có điểm hàn nguội, ngắn mạch hoặc hàn sai chân. Đây là bước quan trọng giúp phát hiện lỗi sớm trước khi cấp nguồn cho toàn hệ thống.
* **Kiểm tra sơ bộ chức năng từng phần:** Trước khi đưa vào mô hình thực tế, nhóm tiến hành cấp nguồn thử cho từng thành phần (ví dụ: màn hình LCD, còi, cảm biến) để đảm bảo mạch hoạt động ổn định và chính xác theo thiết kế.

Qua quá trình hàn và kiểm tra tỉ mỉ, bảng mạch PCB được hoàn thiện có thể dễ dàng kết nối với các mô-đun và thiết bị bên ngoài thông qua các cổng header, giúp cho việc lắp ráp mô hình thuận tiện và có tính mở rộng cao.

A green circuit board with wires and a display

AI-generated content may be incorrect.

Hình 3.5: Mạch thực tế

Sau khi hoàn thiện bảng mạch điều khiển, nhóm tiến hành dựng mô hình giả định nhằm mô phỏng một môi trường thực tế có thể xảy ra rò rỉ khí gas hoặc cháy nổ trong hộ gia đình hoặc nhà kho nhỏ.

* **Vật liệu và cấu trúc mô hình:** Nhóm sử dụng các tấm mica trắng để tạo khung mô hình, chia thành nhiều ngăn tương ứng với các khu vực chức năng như: nơi đặt nguồn phát khí gas, khu vực phát sinh cháy,v.v. Các tấm mica được cố định bằng keo dán chuyên dụng, đảm bảo sự chắc chắn và dễ quan sát khi trình bày.
* **Hệ thống dập cháy mô phỏng:** Sử dụng máy bơm mini nối với hệ thống ống dẫn, để làm chữa cháy tự động. Khi cảm biến lửa phát hiện có cháy, vi điều khiển sẽ kích hoạt relay để bật máy bơm, đưa nước qua ống đến vị trí phát cháy nhằm dập lửa kịp thời.
* **Thiết bị ngoại vi đi kèm:** hệ thống lắp quạt hút khí để xử lý khí gas khi nồng độ vượt ngưỡng và động cơ servo mô phỏng quá trình đóng mở cửa khẩn cấp.
* **Lắp ráp tổng thể:** Sau khi mô hình hoàn chỉnh, toàn bộ hệ thống điều khiển được lắp cố định vào một bên mô hình, giúp dễ dàng trình diễn nguyên lý hoạt động. Các dây dẫn được đi gọn gàng, kết nối từ mạch điều khiển đến các cảm biến và thiết bị thực thi, đảm bảo an toàn và tính thẩm mỹ.

A white box with a green circuit board and a blue screen

AI-generated content may be incorrect.

Hình 3.6: Mô hình sản phẩm

**3.2 Thiết kế phần mềm hệ thống**

**3.2.1 Thiết lập phần mềm**

Để cài đặt phần mềm NodeRed, ta cần phải cài đặt NodeJs trên trang chủ của nó, lựa chọn phiên bản  LTS (Long-Term Support) của Node.js dành cho máy mình. Chạy tệp cài đặt đã tải xuống và hoàn thành quá trình cài đặt [Node.js](http://node.js).

Mở Command Prompt hoặc PowerShell trên Windows. Chạy lệnh sau để cài đặt Node-RED: npm install -g --unsafe-perm node-red

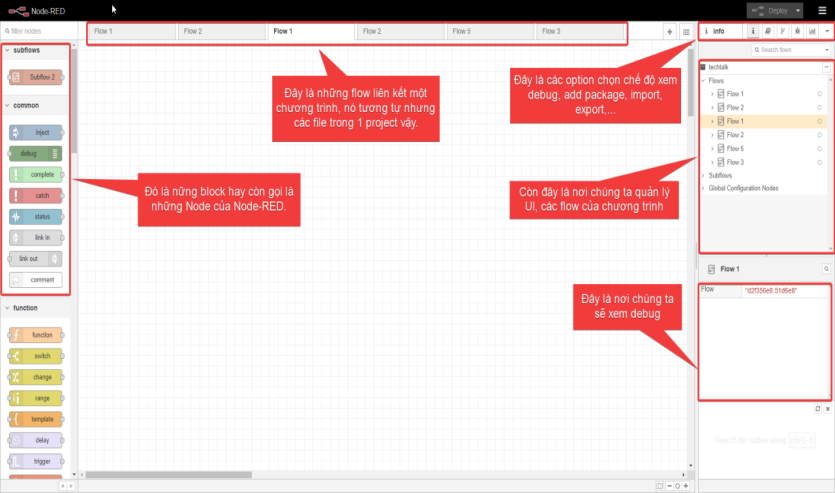
Lệnh trên sẽ tải và cài đặt Node-RED từ kho lưu trữ npm (Node Package Manager). --unsafe-perm được sử dụng để tránh lỗi quyền hạn trong quá trình cài đặt.

Để chạy Node-Red, ta thực hiện như sau:

Sau khi quá trình cài đặt hoàn tất, bạn có thể chạy Node-RED bằng cách chạy lệnh sau trong Command Prompt hoặc PowerShell: node-red

Node-RED sẽ khởi chạy và hiển thị các thông tin liên quan đến quá trình khởi tạo. Mở trình duyệt web và truy cập vào địa chỉ http://localhost:1880 để truy cập giao diện Node-RED.

Giao diện phần mềm được thiết kế để có thể kéo thả các thành phần ra màn hình chính, giúp dễ dàng thiết kế luồng hoạt động của phần mềm, hệ thống. Ngoài ra, ta có thể điều chỉnh cài đặt các thông số của các node thành phần trong hệ thống



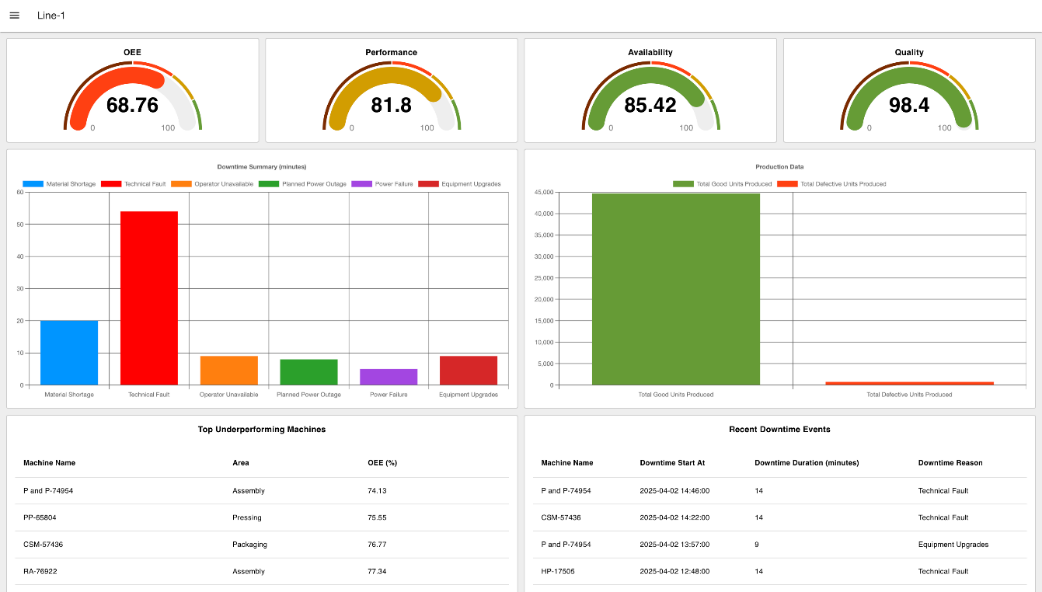
Hình 3.7: Tổng quan giao diện của nền tảng Node-Red

Phía bên tay trái là các palette, bao gồm các node mà ta có thể thêm vào luồng, các node có đầu vào, đầu ra, mỗi node có các chức năng khác nhau, xử lý các hành động cụ thể. Ở chính giữa là phần ta đặt luồng hoạt động giữa các node, ta có thể kéo thả các node ra màn hình đấy, kết nối các node với nhau thông qua dây nối. Phần bên phải là nơi ta quản lý giao diện, quản lý các luồng hoạt động, cài đặt các thông tin cho các node và luồng hoạt động, tải xuống hoặc nhập các luồng có sẵn.

**3.2.2 Thiết kế giao diện phần mềm**

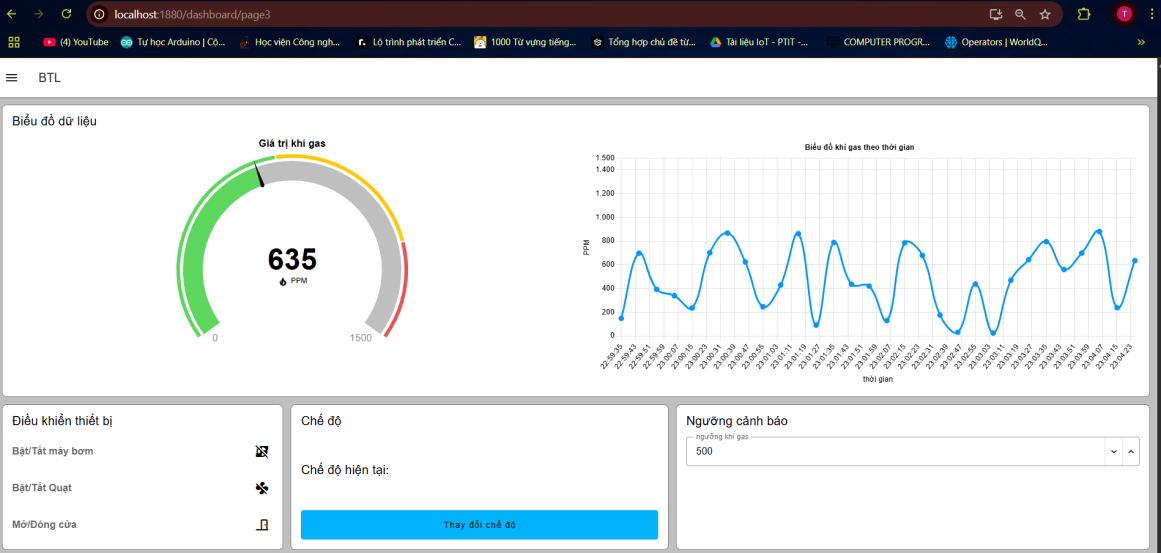
Để xây dựng giao diện trên nền tảng trên, nhóm chúng em sử dụng thư viện NodeRed Dashboard 2.0 để thiết kế giao diện. Thư viện cung cấp các node để xây dựng giao diện người dùng và trực quan hoá dữ liệu cho nhà phát triển. Nó mang lại tính linh hoạt trong việc tuỳ chỉnh bố cục của trang giao diện, đồng thời thể hiện rõ hành đông chức năng xử lý.

Việc cài đặt thư viện và thiết kế vô cùng đơn giản và dễ dàng, giúp cho nhóm có thể thiết kế ra một giao diện web thân thiện với người dùng, dữ liệu được trực quan hoá và điều khiển một cách dễ dàng.



Hình 3.8: Ví dụ về giao diện dashboard khi tạo trên Node-Red

Với nhóm chúng em, chúng em lựa chọn thiết kế một giao diện có đầy đủ chức năng bao gồm trực quan hoá dữ liệu, điều khiển các thiết bị phần dưới, kết nối đến thiết bị thông qua giao thức MQTT. Sau một khoảng thời gian nghiên cứu, tìm hiểu cách thiết kế, chúng em đưa ra một giao diện người dùng và luồng hoạt động của chúng, nó được thực hiện trên nền tảng này như sau:



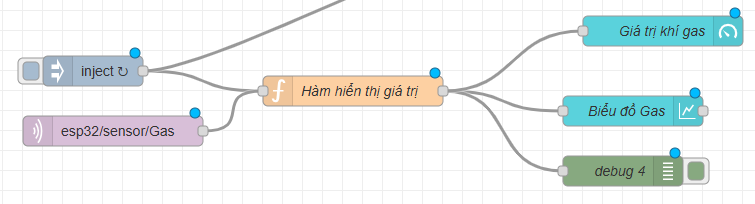
Hình 3.9: Giao diện phần mềm trên nền tảng Node-Red

Mô tả giao diện gồm có 3 phần chính: **Phần biểu đồ dữ liệu, phần điều khiển thiết bị và phần thiết lập chế độ ngưỡng cảnh báo**. Đối với biểu đồ dữ liệu, nó thể hiện giá trị của dữ liệu thu từ cảm biến khí gas MQ2 theo thời gian thực và mốc thời gian cụ thể. Ngoài ra nó còn thể hiện qua các khoảng giá trị khi dữ liệu để biết được dữ liệu đang ở mức tốt hay xấu.

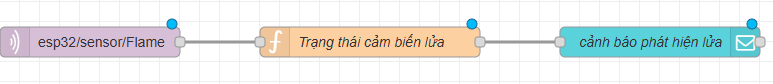
Về phần điều khiển thiết bị, nó đơn giản là việc bật tắt các cơ cấu chấp hành trong thiết bị thông qua nút nhấn và các hình hiển thị. Điều đó, giúp người dùng dễ dàng biết được trạng thái của thiết bị.

Ngoài ra, để nâng cao chức năng cho người dùng, nhóm còn tạo phần chức năng giúp thay đổi chế độ cho người dùng, từ chế độ tự động sang chế độ thủ công để người dùng có thể điều khiển hệ thống. Để thiết lập cảnh báo, nhóm chúng em còn đặt thanh điều chỉnh ngưỡng cảnh báo khí gas để giúp cho hệ thống cảnh báo chính xác hơn.

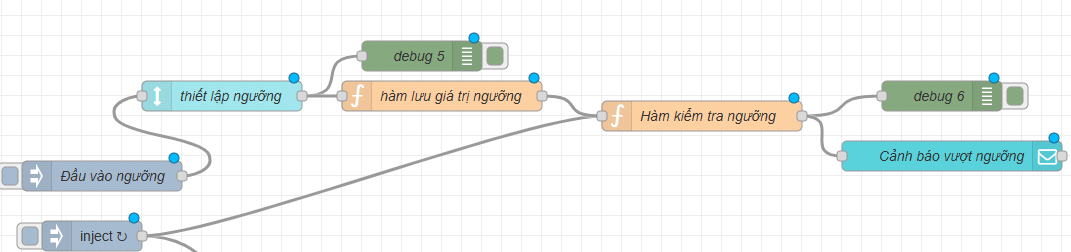
Về các luồng thực hiện biểu diễn trên NodeRed như sau:



Hình 3.10: Luồng hiển thị dữ liệu cảm biến khí gas trên biểu đồ



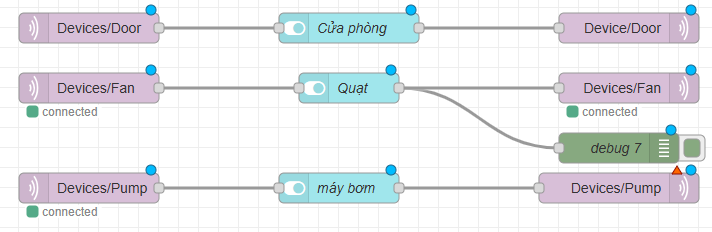
Hình 3.11: Luồng hành động khi cảm biến phát hiện được lửa



Hình 3.12: Luồng điều chỉnh ngưỡng cảm biến khí gas và cảnh báo vượt ngưỡng



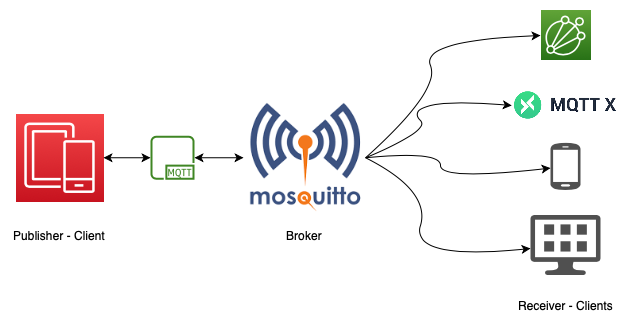
Hình 3.13: Luồng thay đổi chế độ của hệ thống



Hình 3.14: Luồng điều khiển các thiết bị cơ cấu chấp hành

**3.2.3 Giao thức kết nối phần mềm và phần cứng**

Như đã nói ở trên, chúng em sử dụng giao thức MQTT để trao đổi dữ liệu giữa các thiết bị với phần mềm trên máy tính. Để thực hiện, trước tiên ta cài đặt server MQTT trên máy tính cá nhân, em lựa chọn **Eclipse Mosquitto** là server (broker), nơi nhận dữ liệu truyền đến từ thiết bị phần cứng. Việc cài đặt mã nguồn này cũng rất đơn giản, bằng việc lên trang chủ https://mosquitto.org/ , ta tải xuống bản cài đặt và thực hiện cài đặt broker cho mqtt. Sau khi hoàn thành, ta đã tạo ra một broker cho giao thức MQTT để truyền và nhận dữ liệu ngay trên máy tính cá nhân.



Hình 3.15: MQTT Broker Mosquitto

Còn đối với các clients thì đó là ESP32 và nền tảng NodeRed, các clients này có nhiệm vụ truy cập vào topic đã được đăng ký trên broker, sau đó gửi hoặc lấy dữ liệu để hiển thị dữ liệu, điều khiển dữ liệu.

Ở bên ESP32, cảm biến thu được dữ liệu và đưa qua khối xử lý, khối xử lý này đăng ký vào topic và gửi đi thông qua giao thức MQTT đến máy tính cá nhân. Còn phần mềm Node-Red Dashboard cùng đăng kí topic giống đó để lấy dữ liệu nhận được và trực quan hoá dữ liệu.

Ngược lại khi điều khiển thiết bị, người dùng nhấn nút trên giao diện, tự động sự kiện gửi gói tin đến thiết bị kia qua topic điều khiển, từ đó khối xử lý nhận được bản tin và thực hiện việc điều khiển các cơ cấu chấp hành cần thiết.

**CHƯƠNG 4: THỬ NGHIỆM VÀ HIỆU CHỈNH**

**4.1 Kịch bản thử nghiệm hệ thống**

**4.1.1 Kịch bản tổng quan:**

- **Mục tiêu kiểm tra:** Kết nối wifi, MQTT Broker, dữ liệu cảm biến gửi lên dashboard.

- **Công việc thực hiện:** Cắm nguồn, theo dõi kết quả LCD và màn hình laptop

- **Mô tả quá trình hoạt động:** Khi phát hiện khí ga cảm biến sẽ gửi tín hiệu về vi điều khiển để nó thực hiện điều khiển việc mở cửa và bật quạt để thổi khí gas ra bên ngoài nhà, và khi phát hiện lửa cảm biến phát hiện lửa sẽ gửi dữ liệu cho vi điều khiển biết và để điều khiển việc bật còi báo cháy. Cũng trong quá trình phát hiện khí gas và lửa, vi điều khiển ESP32 cũng sẽ đồng thời gửi dữ liệu đến server và cập nhật dashboard theo thời gian thực để thông báo dù người nhận không có ở nhà.

- **Kết quả mong đợi:** LED trạng thái báo OK, màn hình LCD hiển thị kết nối thành công, dữ liệu cập nhật thời gian thực lên server.

**4.1.2 Kịch bản chi tiết:**

**Kịch bản 1: Hệ thống khởi động lần đầu**

- **Mục tiêu kiểm tra:** Kết nối wifi, hệ thống kết nối.

- **Công việc thực hiện:** Cắm nguồn, theo dõi kết quả LCD.

- **Mô tả quá trình hoạt động:** Khởi động hệ thống lần đầu để kiểm tra kết nối wifi và MQTT Broker, bắt đầu tự động cấu hình và các thành phần hệ thống bắt đầu kết nối với nhau.

- **Kết quả mong đợi:** Sau khi khởi động xong các thành phần trong hệ thống đã kết nối với nhau và bắt đầu hoạt động. LCD hiển thị kết nối thành công.

**Kịch bản 2: Hệ thống hoạt động an toàn(Không có khí gas)**

- **Mục tiêu kiểm tra:** Kết nối wifi, hiển thị laptop.

- **Công việc thực hiện:** Theo dõi kết quả hiển thị trên dashboard.

- **Mô tả quá trình hoạt động:** Khi trong trạng thái hoạt động bình thường sau khi khởi động hệ thống, dữ liệu từ cảm biến và màn hình LCD hiển thị các thông số bình thường, và dữ liệu gửi qua MQTT liên tục khi không có vấn đề gì về khí gas hay phát hiện có dấu hiệu lửa (dữ liệu bình thường).

- **Kết quả mong đợi:** Hiển thị trên dashboard các thông số bình thường, kết nối mạng duy trì ổn định.

**Kịch bản 3: Hệ thống phát hiện rò rỉ khí gas**

- **Mục tiêu kiểm tra:** Khi có khí gas cửa sổ mở và quạt gió được bật, thông báo gửi lên dashboard theo thời gian thực.

- **Công việc thực hiện:** Sử dụng khí gas từ bật lửa để vào vùng đo của cảm biến, theo dõi hoạt động sau đó và thông tin hiển thị trên dashboard.

- **Mô tả quá trình hoạt động:** Sau khi phát hiện khí gas qua cảm biến khí gas MQ-2 thì cửa sổ được mở ra đồng thời kích hoạt quạt gió để thổi khí gas ra ngoài, đồng thời thông báo cũng được gửi lên web hiển thị thông báo bằng giao thức MQTT thành công. Sau khi khi gas giảm và không còn được phát hiện bởi cảm biến, vi điều khiển sẽ thay đổi trạng thái quạt gió dừng hoạt động và điều khiển đóng lại cửa sổ.

- **Kết quả mong đợi:** Cảm biến phát hiện ra khí gas và gửi tín hiệu cho vi điều khiển thực hiện các chế độ khác ngay lập tức, thông tin gửi lên dashboard đạt yêu cầu về thời gian thực.

**Kịch bản 4: Hệ thống phát hiện có ngọn lửa**

- **Mục tiêu kiểm tra:** Khi phát hiển có lửa từ cảm biến lửa, còi báo kêu lên và dữ liệu được gửi lên dashboard.

- **Công việc thực hiện:** Sử dụng bật lửa để vào vùng đo của cảm biến, theo dõi hoạt động sau đó và thông tin hiển thị trên dashboard.

- **Mô tả quá trình hoạt động:** Thử nghiệm bằng việc dùng bật lửa bật lên đưa sát vào phạm vi cảm biến phát hiện lửa, sau khi phát hiện có lửa còi báo kêu lên để thông báo, đồng thời dữ liệu từ cảm biến cũng được vi điều khiển gửi lên web thông báo qua MQTT lên dashboard.

- **Kết quả mong đợi:** Cảm biến phát hiện ra lửa và gửi tín hiệu cho vi điều khiển thực hiện các chế độ khác ngay lập tức, thông tin gửi lên dashboard đạt yêu cầu về thời gian thực.

**Kịch bản 5: Bật tắt các thiết bị như quạt, mở cửa, máy bơm bằng cách thủ công**

- Chuyển chế tự động sang điều khiển thủ công các thiết bị như quạt hút, cửa sổ, máy bơm thông qua dashboard web hoặc nút bấm, không cần điều kiện cảm biến.

- Khi người dùng thao tác bật/tắt, ESP32 nhận lệnh qua MQTT hoặc nút bấm, kích hoạt relay điều khiển thiết bị. Trạng thái thực tế được cập nhật ngược lên dashboard.

**- Kết quả mong đợi:** Thiết bị phản hồi đúng lệnh thủ công, trạng thái hiển thị chính xác, và vẫn đảm bảo ưu tiên tự động khi cần.

**4.2 Triển khai hệ thống thực tế và đánh giá ưu nhược điểm**

**Triển khai thực tế:**

Hệ thống được triển khai thực tế tại khu vực bếp của một ngôi nhà. Các cảm biến (MQ-2 để phát hiện khí gas và cảm biến flame để phát hiện lửa) được gắn cố định tại vị trí gần bếp nấu. ESP32 đóng vai trò là bộ điều khiển trung tâm, kết nối WiFi để gửi/nhận dữ liệu MQTT với server.

**Kiểm thử hệ thống được tiến hành trong nhiều tình huống:**

* Rò rỉ khí gas giả lập (từ bật lửa)
* Lửa giả lập (bật lửa)
* Mất kết nối mạng
* Thao tác thủ công bật/tắt thiết bị

**Ưu điểm:**

- **Phản ứng nhanh:** Hệ thống phản hồi tức thời khi có khí gas hoặc lửa.

- **Cảnh báo từ xa:** Gửi thông tin về dashboard giúp người dùng biết sự cố ngay cả khi không có mặt tại nhà.

- **Tự động và thủ công:** Có thể chuyển đổi linh hoạt giữa hai chế độ điều khiển.

- **Dễ mở rộng:** Có thể dễ dàng thêm các thiết bị khác như cảm biến nhiệt độ, camera, hoặc thiết bị chữa cháy.

- **Chi phí thấp:** Sử dụng linh kiện phổ thông như ESP32, relay, cảm biến MQ-2, giúp tiết kiệm chi phí mà vẫn hiệu quả.

**Nhược điểm:**

**- Phụ thuộc mạng:** Nếu mất kết nối WiFi, dashboard sẽ không cập nhật dữ liệu.

**- Thiết bị cơ học (cửa, quạt) có thể hỏng theo thời gian:** Cần kiểm tra và bảo trì định kỳ.

**- Không có nguồn dự phòng:** Khi mất điện, toàn hệ thống ngưng hoạt động nếu không có UPS.

**KẾT LUẬN**

Sản phẩm hệ thống cảnh báo khí gas trong nhà sử dụng công nghệ IoT đã đạt được một số kết quả đáng ghi nhận. Trước hết, hệ thống có khả năng phát hiện rò rỉ khí gas nhanh chóng và chính xác thông qua cảm biến khí, góp phần nâng cao mức độ an toàn cho người dùng. Khi phát hiện nồng độ khí vượt ngưỡng an toàn, hệ thống ngay lập tức kích hoạt còi báo động và gửi cảnh báo đến thiết bị di động qua mạng Wi-Fi, giúp người dùng phản ứng kịp thời, ngay cả khi không có mặt tại nhà. Bên cạnh đó, sản phẩm còn đạt được tính linh hoạt và dễ triển khai trong môi trường thực tế nhờ thiết kế nhỏ gọn, tiết kiệm năng lượng và chi phí thấp. Giao diện hiển thị dữ liệu trên ứng dụng đơn giản, dễ sử dụng, phù hợp với nhiều đối tượng người dùng.

Nhìn chung, sản phẩm đã đáp ứng được các yêu cầu cơ bản của một hệ thống cảnh báo khí gas thông minh, xử lý theo thời gian thực, mang lại hiệu quả thiết thực trong việc bảo vệ an toàn cho các hộ gia đình.

Việc đề xuất và thực hiện giải pháp IoT cảnh báo khí gas thể hiện tinh thần trách nhiệm đạo đức và nghề nghiệp cao của người thực hiện đối với cộng đồng và xã hội. Trước hết, sản phẩm hướng tới mục tiêu bảo vệ sức khỏe, tính mạng và tài sản của con người bằng cách phòng tránh các tai nạn cháy nổ do rò rỉ khí gas – một trong những nguy cơ thường gặp trong sinh hoạt gia đình. Đây là minh chứng rõ ràng cho việc ứng dụng kiến thức chuyên môn vào giải quyết các vấn đề thực tiễn, mang lại giá trị nhân văn và xã hội sâu sắc. Từ góc độ đạo đức nghề nghiệp, người thực hiện cần đảm bảo rằng thiết bị hoạt động ổn định, chính xác, và không gây ra cảnh báo sai gây hoang mang hoặc bỏ sót tình huống nguy hiểm. Việc đảm bảo quyền riêng tư của người dùng cũng cần được chú trọng, đặc biệt trong các hệ thống IoT có thu thập và truyền dữ liệu. Thiết kế hệ thống cần tuân thủ các tiêu chuẩn về an toàn điện, mạng và bảo mật thông tin. Ngoài ra, người phát triển cần trung thực trong quá trình nghiên cứu, thử nghiệm và báo cáo kết quả; không phóng đại hiệu quả sản phẩm và sẵn sàng chịu trách nhiệm trước những rủi ro có thể xảy ra khi hệ thống được đưa vào sử dụng thực tế. Điều này thể hiện phẩm chất đạo đức cần có của một kỹ sư, kỹ thuật viên trong thời đại công nghệ số.

**TÀI LIỆU THAM KHẢO**

[1] B. U. Toreyin, E. B. Soyer, O. Urfalioglu, and A. E. Cetin, “Flame detection using PIR sensors,” in *2008 IEEE 16th Signal Processing, Communication and Applications Conference*, 2008, pp. 1–4.

[2] “ESP32,” *Espressif.com*. [Online]. Available: https://www.espressif.com/en/products/socs/esp32. [Accessed: 07- May-2025].

[3] “Cảm Biến Khí Gas Là Gì? Nguyên Lý Hoạt động Và Cách ứng Dụng,” *Nhà cung cấp Máy tính công nghiệp IPC247*, 26-Jun-2023. [Online]. Available: https://ipc247.com/cam-bien-khi-gas-la-gi/. [Accessed: 07-May-2025].

[4] G. N. Sai, K. P. Sai, K. Ajay, and P. Nuthakki, “Smart LPG gas leakage detection and monitoring system,” in *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2023, pp. 571–576.

[5] N. Ibrahim, Ed., *Gas Leakage and Fire Detector Based on Internet of*  *Thing (IoT) Network*, vol. 1, no. 09. 2023.

[6] Marketing, “What is a Flame Detector and How Does it Work?,” *Crowcon Detection Instruments Limited*, 10-Jan-2024. [Online]. Available: <https://www.crowcon.com/article/what-is-a-flame> detector-and-how-does-it-work/. [Accessed: 07-May-2025].

[7] “Low-code programming for event-driven applications,” *Nodered.org*. [Online]. Available: https://nodered.org/. [Accessed: 07-May-2025].

[8] C. Rattanapoka, S. Chanthakit, A. Chimchai, and A. Sookkeaw, “An MQTT-based IoT cloud platform with flow design by node-RED,” in *2019 Research, Invention, and Innovation Congress (RI2C)*, 2019, pp. 1–6.