

**ĐẠI HỌC HUẾ**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**TIÊU LUẬN**  
**PHÁT TRIỂN ỨNG DỤNG IOT**

**Đề tài:**

**HỆ THỐNG PHÁT HIỆN RÒ RỈ KHÍ GAS VỚI  
ESP32**

Sinh viên thực hiện : **LÊ HỮU NHẬT**  
Khóa : **K45 – HỆ CHÍNH QUY**

**Huế, 04 – 2025**

**ĐẠI HỌC HUẾ**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**TIÊU LUẬN**  
**PHÁT TRIỂN ỨNG DỤNG IOT**

**Đề tài:**

**HỆ THỐNG PHÁT HIỆN RÒ RỈ KHÍ GAS VỚI  
ESP32**

Sinh viên thực hiện : **LÊ HỮU NHẬT**  
Khóa : **K45 – HỆ CHÍNH QUY**  
Giảng viên hướng dẫn : **ThS. Võ Việt Dũng**

**Huế, 04 – 2025**

# LỜI CẢM ƠN

Lời đầu tiên, em xin được bày tỏ lòng biết ơn đối với toàn thể Quý thầy cô Khoa Công nghệ thông tin trường Đại học Khoa học – Đại học Huế đã luôn nhiệt tình giảng dạy và đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Bên cạnh đó, Khoa còn tổ chức các buổi giao lưu giữa các doanh nghiệp và sinh viên thật sự bổ ích, giúp em có cơ hội tìm được một đơn vị thực tập tốt.

Đặc biệt, em xin chân thành cảm ơn sự hướng dẫn tận tình của thầy giáo ThS. Võ Việt Dũng, người đã tận tình hướng dẫn cho em trong suốt quá trình thực hiện đề tài tiểu luận để đạt được kết quả mong muôn.

Đề tài “**Hệ thống phát hiện rò rỉ khí gas với ESP32**” là nội dung mà em đã nghiên cứu và thực hiện Tiểu luận. Em xin cảm ơn Quý thầy cô Khoa Công nghệ thông tin đã giúp đỡ và góp ý để em hoàn thành.

Sau khi hoàn thành tiểu luận, em đã học hỏi và tích lũy được nhiều kiến thức hơn. Do kiến thức của bản thân còn hạn chế và thiếu kinh nghiệm, nên quá trình thực hiện, nếu có sai sót, xin quý thầy cô góp ý để đề tài của em được hoàn thiện hơn.

Cuối cùng, chúc Quý thầy cô lời chúc sức khỏe và luôn thành công trong sự nghiệp của mình.

Em xin chân thành cảm ơn!

Huế, tháng 5 năm 2023

Sinh viên thực hiện

Lê Hữu Nhật

# DANH MỤC CÁC TỪ VIẾT TẮT

- **IoT:** Internet of Things (Internet vạn vật)
- **ESP32:** Espressif Systems 32 (Tên module vi điều khiển)
- **MQ-2:** Một loại cảm biến khí thuộc dòng MQ
- **LPG:** Liquefied Petroleum Gas (Khí hóa lỏng)
- **ADC:** Analog-to-Digital Converter (Bộ chuyển đổi tín hiệu analog sang số)
- **GPIO:** General Purpose Input/Output (Chân vào/ra đa năng)
- **ppm:** Parts per million (Phần triệu)
- **I2C:** Inter-Integrated Circuit (Giao thức truyền thông)
- **LCD:** Liquid Crystal Display (Màn hình tinh thể lỏng)

# **PHỤ LỤC**

**Hình 1:** Truyền động, tương tác trong thế giới

**Hình 2:** Sự phát triển của các thiết bị

**Hình 3:** IoT ứng dụng vào những lĩnh vực

**Hình 4:** Kiến trúc hệ thống IoT

**Hình 5:** ESP32 DevKit-C V4

**Hình 6:** Cảm biến khí gas MQ-2

**Hình 7:** Màn hình LCD 16x2 (I2C)

**Hình 8:** Buzzer (Còi cảnh báo)

**Hình 9:** LED đỏ (Đèn báo động)

**Hình 10:** Điện trở 220 Ω

**Hình 11:** Web/Ứng dụng Blynk

**Hình 12:** Wokwi

**Hình 13:** Telegram web

**Hình 14:** Telegram app

**Hình 15:** Trạng thái cảnh báo khi vượt ngưỡng

**Hình 16:** Trạng thái bình thường khi dưới ngưỡng

**Hình 17:** Sơ đồ khái niệm phát hiện rò rỉ khí gas với ESP32 (Wokwi)

**Hình 18:** Kết quả trả về trạng thái bình thường trên Blynk Web

**Hình 19:** Kết quả trả về trạng thái bình thường trên Blynk App

**Hình 20:** Kết quả trả về trạng thái bình thường trên Telegram Web

**Hình 21:** Kết quả trả về trạng thái bình thường trên Telegram App

**Hình 22:** Kết quả trả về trạng thái cảnh báo trên Blynk Web

**Hình 23:** Kết quả trả về trạng thái cảnh báo trên Blynk App

**Hình 24:** Kết quả trả về trạng thái cảnh báo trên Telegram Web

**Hình 25:** Kết quả trả về trạng thái cảnh báo trên Telegram App

# MỤC LỤC

<b>PHẦN MỞ ĐẦU .....</b>	1
I. Lý do chọn đề tài .....	1
II. Mục tiêu đề tài.....	1
1. Mục tiêu tổng quát:.....	1
2. Mục tiêu cụ thể:.....	1
III. Đối tượng và phạm vi nghiên cứu .....	2
1. Đối tượng nghiên cứu:.....	2
2. Phạm vi nghiên cứu:.....	2
IV. Phương pháp nghiên cứu.....	2
1. Phương pháp thu thập thông tin: .....	2
2. Phương pháp phát triển hệ thống thông tin: .....	2
V. Các công trình nghiên cứu liên quan.....	3
<b>PHẦN NỘI DUNG .....</b>	4
I. Giới thiệu về Internet of Things.....	4
1. Internet of Things (IoT) là gì? .....	4
2. Ứng dụng của IoT .....	5
3. Các thành phần trong hệ thống IoT .....	6
II. Tổng quan về công nghệ.....	8
1. ESP32 DevKit-C V4.....	8
2. Cảm biến khí gas MQ-2 .....	9
3. Màn hình LCD 16x2 (I2C).....	10
4. Buzzer (Còi cảnh báo) .....	11
5. LED đỏ (Đèn báo động) .....	12
6. Điện trở 220Ω.....	13
7. Web/Ứng dụng Blynk .....	14

8. Wokwi .....	15
9. Telegram .....	16
III. Nguyên lý hoạt động .....	19
1. Thu thập dữ liệu từ cảm biến:.....	19
2. Xử lý dữ liệu:.....	19
3. Kích hoạt cảnh báo khi vượt ngưỡng: .....	19
4. Trạng thái bình thường khi dưới ngưỡng: .....	20
5. Chu kỳ hoạt động: .....	21
IV. Thiết kế hệ thống.....	21
1. Các thành phần cần có:.....	22
2. Kết nối linh kiện: .....	22
V. Lập trình.....	23
1. Phần mềm phát triển trên Arduino IDE:.....	23
2. Phần lập trình C++ trên VS .....	23
3. Trên Web/Ứng dụng Blynk và Web/Ứng dụng Telegram .....	27
VI. Ưu điểm và hạn chế.....	33
VII. Ứng dụng thực tiễn .....	35
<b>PHẦN KẾT LUẬN.....</b>	<b>36</b>
I. Kết quả đạt được .....	36
II. Hạn chế của đề tài.....	36
III. Hướng phát triển của đề tài .....	37
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>38</b>

# PHẦN MỞ ĐẦU

## I. Lý do chọn đề tài

Trong bối cảnh công nghệ Internet of Things (IoT) phát triển mạnh mẽ, các hệ thống an toàn thông minh ngày càng trở thành giải pháp thiết yếu để bảo vệ con người trước những nguy cơ tiềm ẩn. Một trong những mối đe dọa nghiêm trọng là rò rỉ khí gas, đặc biệt là khí LPG (khí hóa lỏng), có thể gây ra cháy nổ, ngộ độc hoặc thiệt hại lớn về người và tài sản nếu không được phát hiện và xử lý kịp thời. Thực tế cho thấy, nhiều vụ tai nạn liên quan đến rò rỉ khí gas vẫn xảy ra do thiếu các hệ thống cảnh báo hiệu quả, chi phí thấp và dễ triển khai. Trước nhu cầu cấp thiết đó, đề tài “Hệ thống phát hiện rò rỉ khí gas với ESP32” được lựa chọn nhằm xây dựng một giải pháp cảnh báo nhanh chóng, tiết kiệm chi phí, và tận dụng tiềm năng của IoT để nâng cao an toàn trong đời sống.

## II. Mục tiêu đề tài

Đề tài hướng đến việc thiết kế và phát triển một hệ thống phát hiện rò rỉ khí gas sử dụng vi điều khiển ESP32 DevKit-C V4, tích hợp các công nghệ hiện đại để đảm bảo tính hiệu quả và tiện lợi.

### 1. Mục tiêu tổng quát:

Xây dựng một hệ thống IoT có khả năng phát hiện rò rỉ khí gas, cảnh báo người dùng qua tín hiệu âm thanh, ánh sáng và thông báo từ xa, đồng thời cung cấp giải pháp an toàn với chi phí hợp lý.

### 2. Mục tiêu cụ thể:

- Tích hợp cảm biến khí MQ-2, màn hình LCD 16x2 I2C, buzzer, LED đỏ và ứng dụng Blynk để phát hiện và cảnh báo rò rỉ khí gas.
- Mô phỏng hệ thống trên công cụ Wokwi để kiểm tra tính khả thi trước khi triển khai thực tế.

- Đánh giá hiệu quả của hệ thống và đề xuất các cải tiến để ứng dụng trong thực tiễn.

### **III. Đối tượng và phạm vi nghiên cứu**

#### **1. Đối tượng nghiên cứu:**

Hệ thống phát hiện rò rỉ khí gas sử dụng vi điều khiển ESP32, tập trung vào việc phát hiện khí LPG với cảm biến MQ-2 và tích hợp các thành phần phần cứng/phần mềm như LCD, buzzer, LED, và ứng dụng Blynk.

#### **2. Phạm vi nghiên cứu:**

Đề tài giới hạn trong việc thiết kế, lập trình và mô phỏng hệ thống trên nền tảng Wokwi, với các thành phần phần cứng bao gồm ESP32 DevKit-C V4, cảm biến MQ-2, LCD 16x2 I2C, buzzer, LED đỏ, và ứng dụng Blynk để gửi thông báo từ xa. Nghiên cứu không bao gồm việc triển khai thực tế trên thiết bị vật lý mà tập trung vào mô phỏng và phân tích hiệu quả hệ thống.

### **IV. Phương pháp nghiên cứu**

#### **1. Phương pháp thu thập thông tin:**

- Thu thập tài liệu và nghiên cứu các công trình liên quan về hệ thống phát hiện rò rỉ khí gas, ứng dụng IoT trong an toàn, và cách sử dụng vi điều khiển ESP32 cùng cảm biến MQ-2.
- Tham khảo tài liệu kỹ thuật của ESP32 DevKit-C V4, cảm biến MQ-2, và ứng dụng Blynk để hiểu rõ thông số và cách tích hợp.

#### **2. Phương pháp phát triển hệ thống thông tin:**

- Sử dụng phương pháp phát triển phần mềm theo mô hình Agile, bao gồm các bước: phân tích yêu cầu, thiết kế hệ thống, lập trình, mô phỏng và kiểm thử trên Wokwi.

- Tích hợp phần cứng và phần mềm: lập trình ESP32 để đọc dữ liệu từ cảm biến MQ-2, hiển thị trên LCD, kích hoạt buzzer/LED khi phát hiện rò rỉ, và gửi thông báo qua Blynk.

## V. Các công trình nghiên cứu liên quan

Nhiều nghiên cứu trước đây đã tập trung vào việc phát triển các hệ thống phát hiện rò rỉ khí gas sử dụng các vi điều khiển như Arduino hoặc Raspberry Pi, kết hợp với cảm biến khí và các phương thức cảnh báo khác nhau. Tuy nhiên, việc sử dụng ESP32 – một nền tảng tích hợp Wi-Fi và Bluetooth – cùng với ứng dụng Blynk để gửi thông báo từ xa vẫn còn là một hướng đi mới mẻ, đặc biệt khi kết hợp mô phỏng trên Wokwi để giảm chi phí thử nghiệm. Đề tài này kế thừa các nghiên cứu trước về cảm biến MQ-2 và IoT, đồng thời đóng góp một giải pháp toàn diện hơn với chi phí thấp và khả năng triển khai linh hoạt.

# PHẦN NỘI DUNG

## I. Giới thiệu về Internet of Things

### 1. Internet of Things (IoT) là gì?

- Thuật ngữ 'Internet of Things' được Kevin Ashton đặt ra vào năm 1999, để chỉ việc kết nối Internet với các thiết bị nhỏ gọn, có khả năng đo đạc, giám sát điều kiện môi trường thông qua các cảm biến, sau đó gửi dữ liệu này về một máy chủ tập trung (server). Kể từ đó, thuật ngữ này đã được sử dụng để mô tả bất kỳ thiết bị nào tương tác với thế giới vật lý xung quanh nó, bằng cách thu thập dữ liệu từ cảm biến (sensor) hoặc cung cấp các tương tác trong thế giới thực thông qua bộ truyền động (actuator - thiết bị thực hiện một số thao tác như bật công tắc hoặc bật đèn LED). Các thiết bị này sẽ được bổ sung thêm khả năng kết nối với mạng Internet, để truyền tải dữ liệu lên điện toán đám mây (cloud server).



**Hình 1.** Truyền động, tương tác trong thế giới

- Cảm biến thu thập thông tin từ thế giới xung quanh, chẳng hạn như đo nhiệt độ, ánh sáng, tốc độ hoặc vị trí... Thiết bị truyền động chuyển đổi

tín hiệu điện thành các tương tác trong thế giới thực như kích hoạt công tắc, bật đèn, tạo âm thanh hoặc gửi tín hiệu điều khiển đến phần cứng khác, chẳng hạn như để bật ổ cắm điện. IoT là một lĩnh vực công nghệ đang phát triển nhanh chóng. Nhờ việc tối ưu kích thước và giá thành của các cảm biến cũng như thiết bị truyền động, mà các ứng dụng IoT có thể dễ dàng được hiện thực và tích hợp. Ước tính đến cuối năm 2020, 50 tỷ thiết bị IoT đã được triển khai và kết nối vào mạng Internet. Nhìn về tương lai, ước tính đến năm 2025, các thiết bị IoT sẽ thu thập được gần 80 zettabyte dữ liệu hoặc 80 nghìn tỷ gigabyte. Đó là rất nhiều dữ liệu, là nguồn thông tin quan trọng cho các công nghệ phía sau IoT, chẳng hạn như phân tích dữ liệu, khoa học dữ liệu hay các bộ não nhân tạo.



*Hình 2. Sự phát triển của các thiết bị*

- Sự phát triển của các thiết bị kết nối vạn vật: Hơn 50 tỉ thiết bị vào cuối năm 2020 (Kết quả tham khảo từ Cisco).

## 2. Ứng dụng của IoT

Công nghệ IoT có thể được ứng dụng trong rất nhiều lĩnh vực và có thể chia thành 1 số nhóm lớn như:

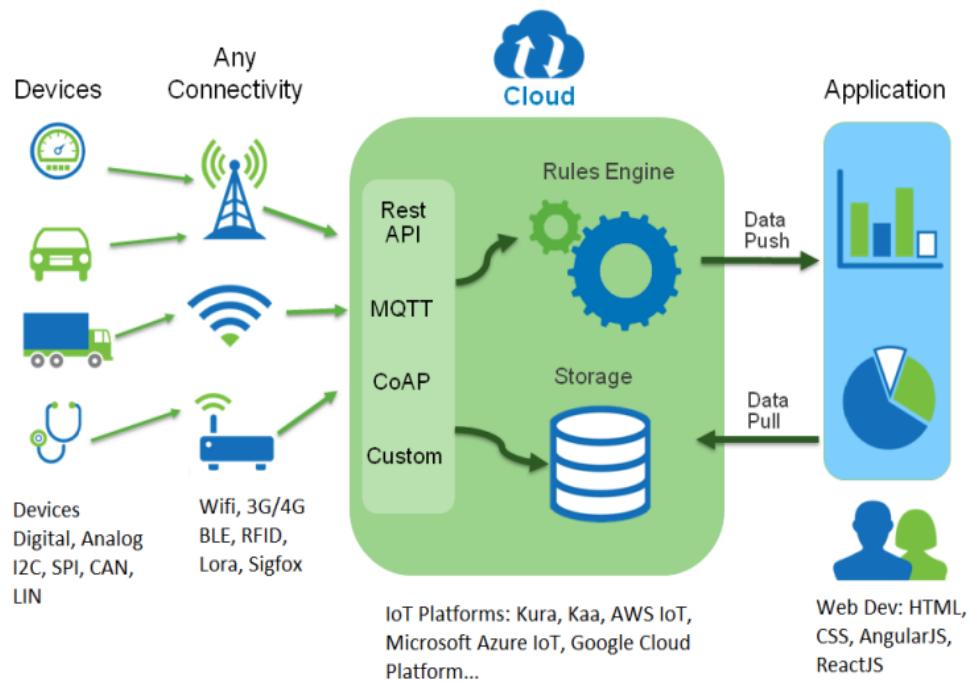
- Giải pháp tiêu dùng (smart home, wearable)
- Giải pháp thương mại, doanh nghiệp (smart tracking, building management system...)
- Công nghiệp (smart factory, smart agriculture, smart healthcare...)
- Cơ sở hạ tầng (smart city, smart grid...)



*Hình 3. IoT ứng dụng vào những lĩnh vực*

### 3. Các thành phần trong hệ thống IoT

Dưới đây là kiến trúc một hệ thống IoT tiêu biểu:



**Hình 4. Kiến trúc hệ thống IoT**

### Thiết bị (Things):

- Là các thiết bị được tích hợp cảm biến, bộ vi điều khiển và có khả năng kết nối ra Internet hoặc các gateway trong hệ thống.
- Có nhiệm vụ thu thập dữ liệu về môi trường xung quanh như nhiệt độ, độ ẩm, áp suất, chuyển động, v.v.
- Giao tiếp và chia sẻ dữ liệu với các thiết bị khác qua mạng Internet.
- Ví dụ: cảm biến nhiệt độ, camera thông minh, đồng hồ thông minh, v.v.

### IoT Gateway:

- Hỗ trợ và là trung gian kết nối giữa các thiết bị IoT và mạng Internet.
- Hỗ trợ nhiều giao thức kết nối khác nhau như Wi-Fi, Bluetooth, Zigbee, v.v.
- Giúp quản lý và bảo mật dữ liệu truyền tải giữa các thiết bị.
- Giúp xử lý các dữ liệu thu thập được ở cục bộ trước khi đưa lên Internet để tăng hiệu suất và giảm chi phí xử lý (Edge computing).

### **IoT Platform:**

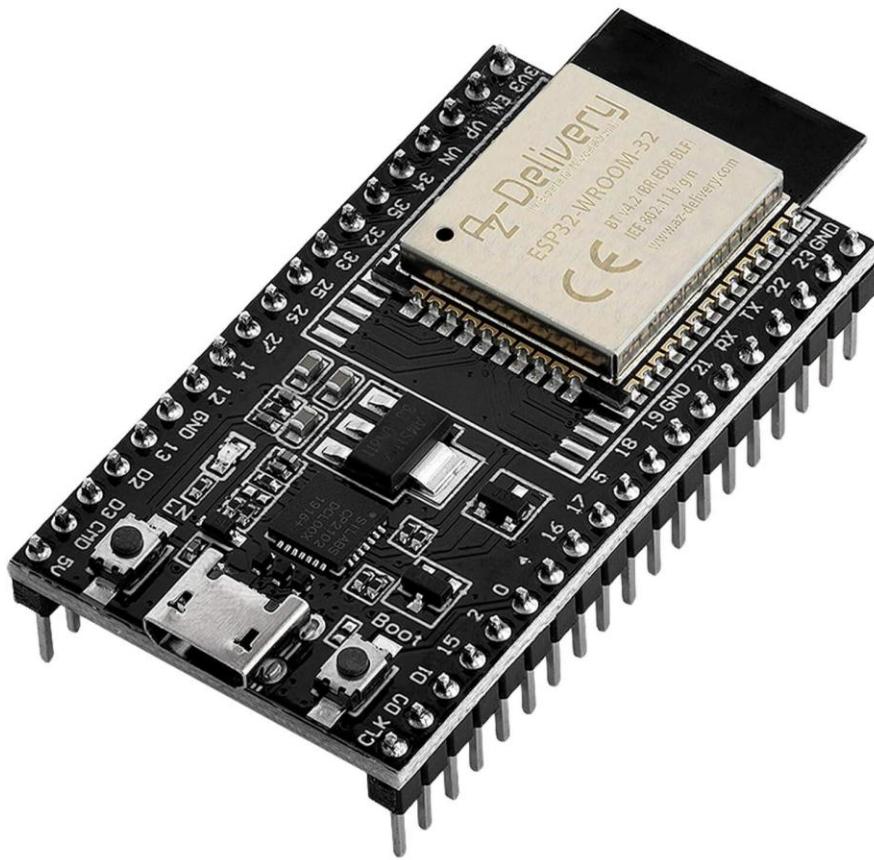
- Nhận và lưu trữ các dữ liệu thu thập được từ các thiết bị IoT.
- Phân tích dữ liệu thu thập được từ các thiết bị IoT.
- Hỗ trợ ra quyết định và thực hiện các hành động tự động theo kịch bản được cấu hình sẵn (Rules Engine).
- Thường được triển khai trên các dịch vụ đám mây

### **Giao diện người dùng (User Interface):**

- Cung cấp giao diện cho người dùng tương tác với hệ thống IoT.
- Hiển thị dữ liệu được thu thập và phân tích từ các thiết bị IoT.
- Cho phép người dùng điều khiển và quản lý các thiết bị IoT.
- Bao gồm ứng dụng di động, trang web, bảng điều khiển, v.v.
- Ví dụ: ứng dụng Smart Home, bảng điều khiển điều khiển tòa nhà, v.v. Ngoài ra, hệ thống IoT có thể bao gồm các thành phần khác như:
  - Hệ thống lưu trữ: lưu trữ dữ liệu thu thập được từ các thiết bị IoT.
  - Hệ thống bảo mật: bảo vệ hệ thống IoT khỏi các mối đe dọa an ninh mạng.
  - Hệ thống quản lý: quản lý và điều phối các hoạt động của hệ thống IoT. Tóm lại, một hệ thống IoT bao gồm nhiều thành phần khác nhau hoạt động cùng nhau để thu thập, truyền tải, phân tích và xử lý dữ liệu từ các thiết bị IoT. Nhờ đó, hệ thống IoT giúp con người đưa ra quyết định sáng suốt hơn, tự động hóa các quy trình và cải thiện chất lượng cuộc sống.

## **II. Tổng quan về công nghệ**

### **1. ESP32 DevKit-C V4**



**Hình 5.** *ESP32 DevKit-C V4*

- **Khái niệm:** ESP32 DevKit-C V4 là phiên bản phát triển của vi điều khiển ESP32 do Espressif Systems thiết kế, tích hợp Wi-Fi và Bluetooth, phù hợp cho các ứng dụng IoT [1].
- **Thông số kỹ thuật:**
  - **CPU:** Xtensa dual-core 32-bit LX6, 160-240 MHz.
  - **RAM:** 520 KB SRAM.
  - **Kết nối:** Wi-Fi 802.11 b/g/n, Bluetooth v4.2.
  - **GPIO:** 36 chân (ADC, DAC, PWM, I2C, SPI).
  - **Nguồn:** 5V qua USB hoặc 3.3V qua pin, dòng tiêu thụ 80 mA.

## 2. Cảm biến khí gas MQ-2



**Hình 6.** Cảm biến khí gas MQ-2

- **Khái niệm:** MQ-2 là cảm biến khí thuộc dòng MQ, sử dụng oxit thiếc ( $\text{SnO}_2$ ) làm vật liệu nhạy khí, phát hiện LPG, CO, khói và các khí dễ cháy khác [2].
- **Thông số kỹ thuật:**
  - **Phạm vi phát hiện:** 300-10.000 ppm.
  - **Điện áp:**  $5\text{V} \pm 0.1\text{V}$ .
  - **Dòng tiêu thụ:** 150 mA.
  - **Thời gian làm nóng:** 24-48 giờ.
  - **Đầu ra:** Analog (0-5V).

### 3. Màn hình LCD 16x2 (I2C)



**Hình 7.** Màn hình LCD 16x2 (I2C)

- **Khái niệm:** LCD 16x2 là màn hình tinh thể lỏng hiển thị 16 ký tự trên 2 dòng, tích hợp module I2C để giao tiếp với vi điều khiển qua giao thức I2C [3].
- **Thông số kỹ thuật:**
  - **Kích thước:** 16 ký tự x 2 dòng.
  - **Điện áp:** 5V.
  - **Giao thức:** I2C (SDA, SCL).
  - **Địa chỉ I2C:** 0x27 (có thể thay đổi).
  - **Dòng tiêu thụ:** 20-40 mA.

#### 4. Buzzer (Còi cảnh báo)



**Hình 8. Buzzer (Còi cảnh báo)**

- **Khái niệm:** Buzzer là thiết bị âm thanh nhỏ gọn, hoạt động dựa trên dao động điện từ hoặc áp điện để tạo âm thanh cảnh báo [4].
- **Thông số kỹ thuật:**
  - **Điện áp:** 3.3-5V.
  - **Tần số:** 2-4 kHz.
  - **Độ ồn:** 85-100 dB.
  - **Dòng tiêu thụ:** 15-30 mA.
  - **Kích thước:** Đường kính 12 mm.

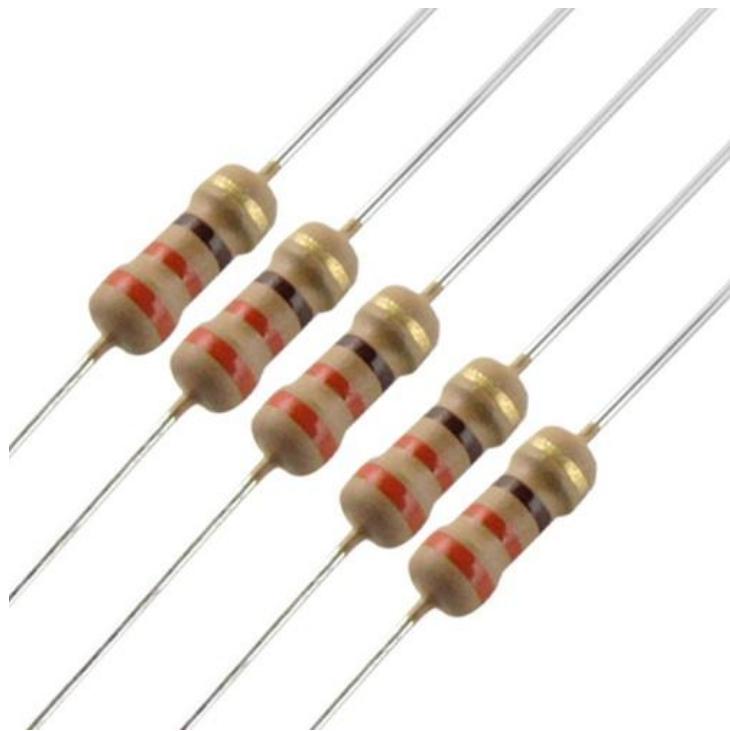
## 5. LED đỏ (Đèn báo động)



**Hình 9. LED đỏ (Đèn báo động)**

- **Khái niệm:** LED đỏ là diode phát quang, sử dụng để báo hiệu bằng ánh sáng khi có sự kiện xảy ra [5].
- **Thông số kỹ thuật:**
  - **Điện áp:** 1.8-2.2V.
  - **Dòng điện:** 10-20 mA.
  - **Màu sắc:** Đỏ.
  - **Độ sáng:** 100-200 mcd.

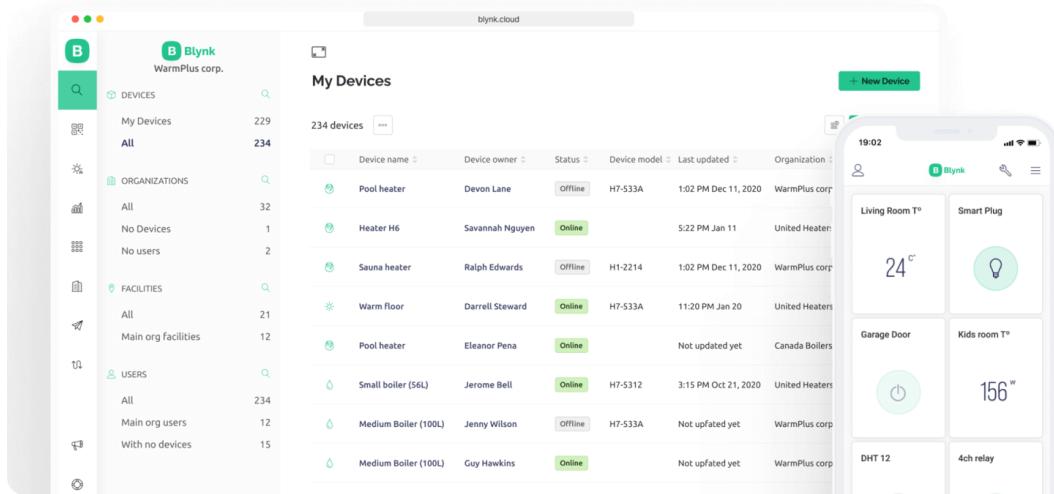
## 6. Điện trở $220\Omega$



**Hình 10. Điện trở 220 Ω**

- **Khái niệm:** Điện trở 220Ω là linh kiện thụ động, dùng để hạn dòng cho LED, đảm bảo hoạt động an toàn [5].
- **Thông số kỹ thuật:**
  - **Giá trị:** 220 Ohm  $\pm 5\%$ .
  - **Công suất:** 0.25W.
  - **Kích thước:** Chuẩn 1/4W.

## 7. Web/Ứng dụng Blynk



**Hình 11.** Web/Ứng dụng Blynk

- **Giới thiệu:** Blynk được phát triển bởi Blynk Inc., ra mắt năm 2014, là nền tảng IoT phổ biến giúp thiết kế giao diện điều khiển và giám sát thiết bị qua điện thoại [6].
- **Khái niệm:** Blynk là nền tảng IoT cho phép giám sát và điều khiển thiết bị từ xa qua ứng dụng di động [6].
- **Đặc tả dịch vụ:**
  - **Hỗ trợ:** Wi-Fi, Bluetooth, Ethernet.
  - **Giao thức:** TCP/IP, HTTP, WebSocket.
  - **Nền tảng:** iOS, Android, Web.
  - **Dung lượng:** 100 thiết bị miễn phí.
  - **Thời gian phản hồi:** 1-3 giây.

## 8. Wokwi



*Hình 12. Wokwi*

- **Giới thiệu:** Wokwi là công cụ mô phỏng phần cứng trực tuyến, ra mắt năm 2020 bởi nhóm CodeCraft, hỗ trợ thử nghiệm các dự án IoT [7].
- **Khái niệm:** Wokwi là nền tảng mô phỏng trực tuyến cho phép thiết kế, lập trình và kiểm tra mạch điện tử với vi điều khiển như ESP32 [7].
- **Đặc tả dịch vụ:**
  - **Hỗ trợ:** ESP32, Arduino, Raspberry Pi Pico.
  - **Ngôn ngữ:** C/C++, MicroPython.
  - **Giao diện:** Trình chỉnh sửa mã và sơ đồ mạch.
  - **Định dạng:** Hỗ trợ JSON.
  - **Truy cập:** Web-based.

## 9. Telegram



**Hình 13.** Telegram web



**Hình 14.** Telegram app

- **Giới thiệu:** Telegram là ứng dụng nhắn tin tức thời được phát triển bởi Telegram FZ-LLC, ra mắt năm 2013, nổi tiếng với tính bảo mật cao và khả năng tích hợp bot để tự động hóa các tác vụ [8].
- **Khái niệm:** Telegram là nền tảng nhắn tin đa năng, hỗ trợ gửi thông báo từ xa và tích hợp với các hệ thống IoT thông qua Telegram Bot API, cho phép người dùng nhận cảnh báo hoặc điều khiển thiết bị từ xa [8].
- **Đặc tả dịch vụ:**
  - Hỗ trợ: iOS, Android, Windows, macOS, Linux, Web.
  - Giao thức: MTProto (giao thức mã hóa riêng của Telegram).
  - Tính năng: Hỗ trợ bot, gửi tin nhắn, hình ảnh, tệp (lên đến 2GB).
  - Bảo mật: Mã hóa đầu cuối cho các cuộc trò chuyện bí mật.
  - Thời gian phản hồi: Dưới 1 giây (tùy thuộc vào kết nối mạng).

- Dung lượng: Không giới hạn số lượng người dùng trong nhóm (lên đến 200.000 thành viên).

### **III. Nguyên lý hoạt động**

Hệ thống phát hiện rò rỉ khí gas với ESP32 DevKit-C V4 hoạt động dựa trên sự phối hợp giữa các linh kiện phần cứng và phần mềm, được chia thành các giai đoạn cụ thể như sau:

#### **1. Thu thập dữ liệu từ cảm biến:**

- Cảm biến MQ-2 liên tục đo nồng độ khí gas trong không khí. Khi phát hiện khí LPG, CO hoặc khói, MQ-2 chuyển đổi tín hiệu thành giá trị điện áp analog (từ 0-5V) và gửi đến ESP32 qua chân GPIO 34. Giá trị này tỷ lệ thuận với nồng độ khí, được quy đổi thành đơn vị ppm (parts per million).

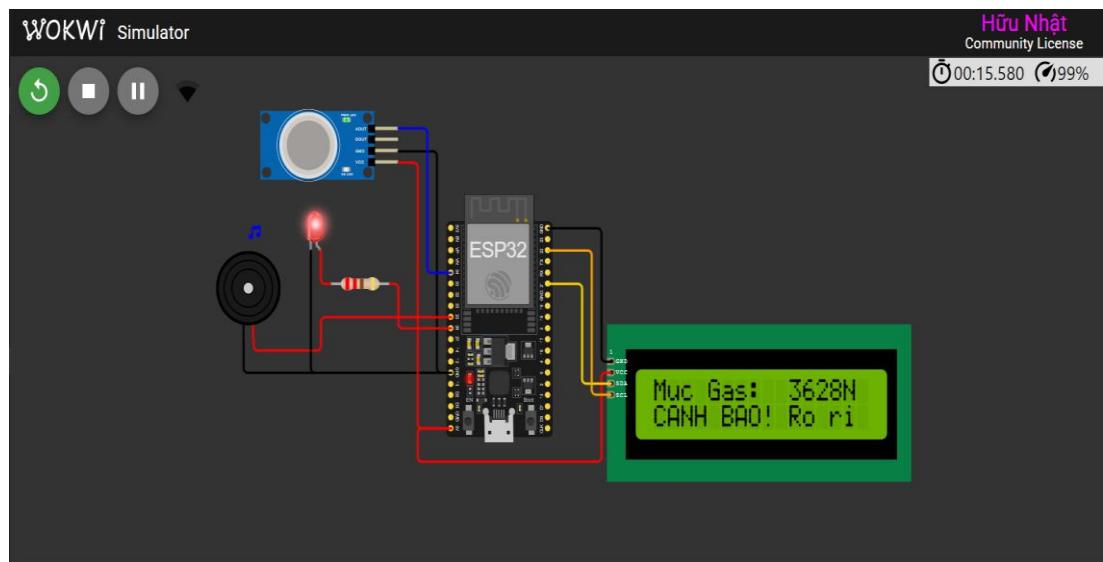
#### **2. Xử lý dữ liệu:**

- ESP32 nhận tín hiệu analog từ MQ-2 thông qua bộ chuyển đổi ADC tích hợp. Module này đọc giá trị và so sánh với ngưỡng an toàn được thiết lập là 1500 ppm. Ngưỡng này được chọn dựa trên mức độ nguy hiểm của khí gas trong môi trường sống, đảm bảo cảnh báo kịp thời trước khi đạt đến mức gây nguy cơ cháy nổ hoặc ngộ độc.

#### **3. Kích hoạt cảnh báo khi vượt ngưỡng:**

- Nếu nồng độ khí vượt quá 1500 ppm:
  - Buzzer: ESP32 gửi tín hiệu HIGH đến chân GPIO 25, kích hoạt buzzer phát ra âm thanh cảnh báo với tần số 2-4 kHz, thông báo sự cố tại chỗ.
  - LED đỏ: ESP32 gửi tín hiệu HIGH đến chân GPIO 26, qua điện trở  $220\Omega$ , làm LED đỏ sáng để báo hiệu trực quan.
  - LCD 16x2: Qua giao thức I2C (GPIO 21 - SDA, GPIO 22 - SCL), ESP32 gửi lệnh hiển thị thông điệp “Gas Detected” trên màn hình LCD, cung cấp thông tin chi tiết cho người dùng.

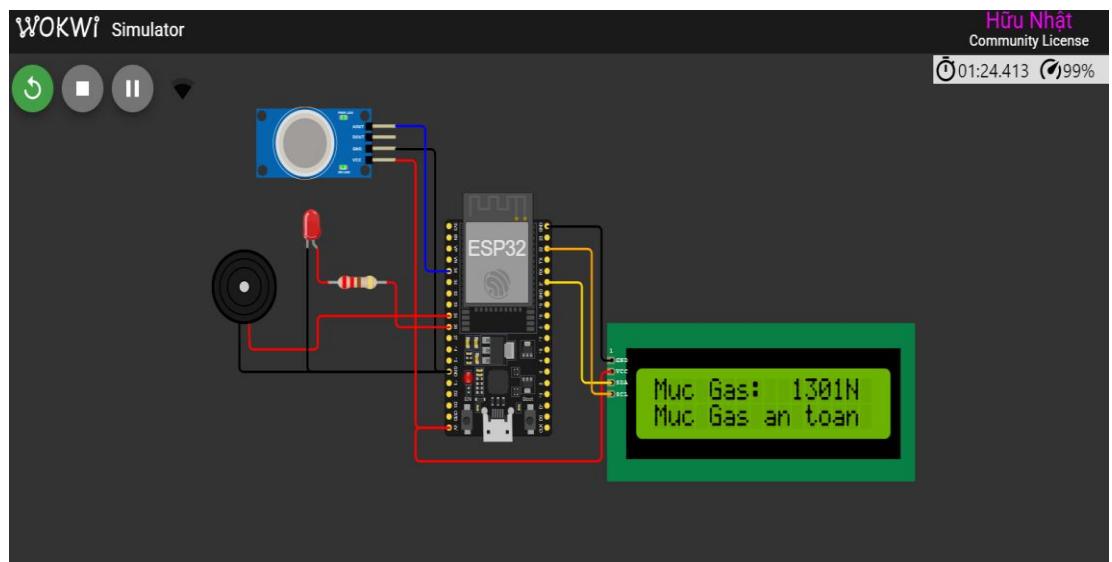
- Blynk: ESP32 sử dụng kết nối Wi-Fi để gửi thông báo “Cảnh báo: Phát hiện rò rỉ khí gas!” đến ứng dụng Blynk trên điện thoại, cho phép giám sát từ xa.
- Telegram: ESP32 sử dụng kết nối Wi-Fi để gửi thông báo “Cảnh báo: Phát hiện rò rỉ khí gas!” đến ứng dụng Telegram trên điện thoại, cho phép giám sát từ xa.
- Các cảnh báo này hoạt động đồng thời để đảm bảo thông tin được truyền tải qua nhiều kênh (âm thanh, ánh sáng, màn hình, và ứng dụng).



**Hình 15.** Trạng thái cảnh báo khi vượt ngưỡng

#### 4. Trạng thái bình thường khi dưới ngưỡng:

- Nếu nồng độ khí dưới 1500 ppm, ESP32 duy trì trạng thái an toàn:
  - Buzzer và LED đỏ được tắt (tín hiệu LOW tại GPIO 25 và GPIO 26).
  - LCD hiển thị “Safe” để xác nhận không có nguy cơ.
  - Blynk: ESP32 sử dụng kết nối Wi-Fi để gửi thông báo “Mức Gas an toàn” đến ứng dụng Blynk trên điện thoại, cho phép giám sát từ xa.
  - Telegram: ESP32 sử dụng kết nối Wi-Fi để gửi thông báo “Mức Gas an toàn” đến ứng dụng Telegram trên điện thoại, cho phép giám sát từ xa.



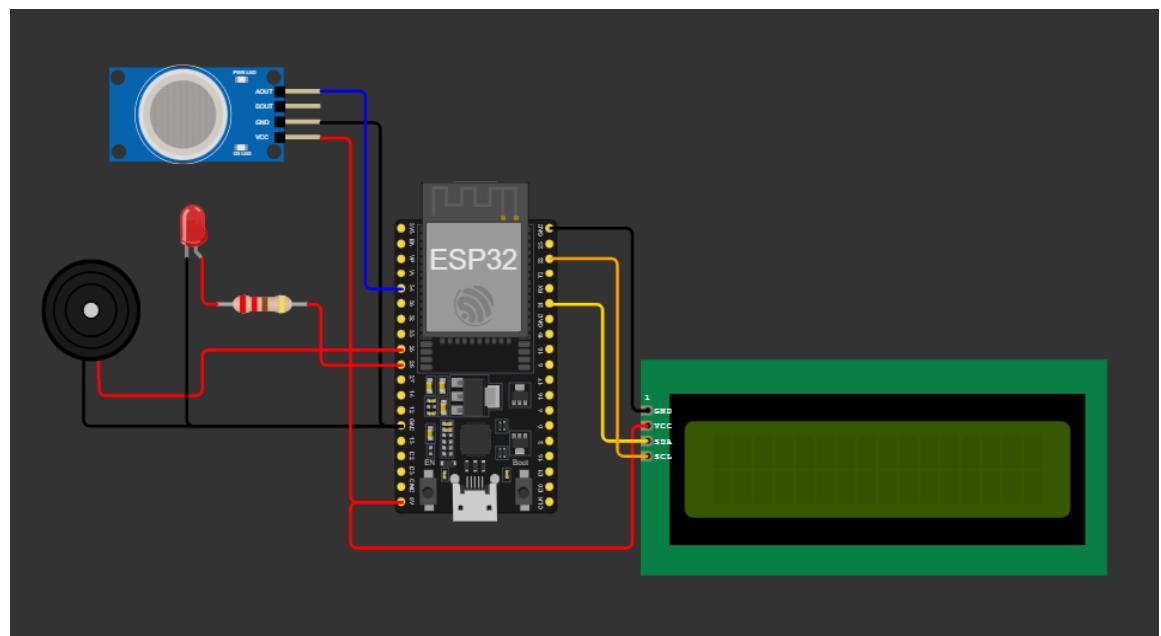
**Hình 16.** Trạng thái bình thường khi dưới ngưỡng

### 5. Chu kỳ hoạt động:

- Hệ thống hoạt động theo chu kỳ lặp lại, với ESP32 liên tục đọc dữ liệu từ MQ-2 mỗi 1 giây (có thể điều chỉnh trong phần mềm), đảm bảo phát hiện kịp thời mọi thay đổi trong môi trường.

Nguyên lý này tận dụng khả năng xử lý nhanh của ESP32 và sự kết hợp đa dạng của các thiết bị đầu ra, mang lại hiệu quả cảnh báo cao cả tại chỗ và từ xa.

## IV. Thiết kế hệ thống



## **Hình 17. Sơ đồ khái niệm hệ thống phát hiện rò rỉ khí gas với ESP32 (Wokwi)**

### **1. Các thành phần cần có:**

- ESP32 Dev Module
- Cảm biến khí gas MQ-2
- Màn hình LCD 16x2 (I2C - PCF8574)
- Buzzer (còi cảnh báo)
- LED (đèn báo động)
- Điện trở  $220\Omega$  (cho LED)
- Dây nối (jumper)

### **2. Kết nối linh kiện:**

- MQ-2 Sensor ↔ ESP32:

- VCC → 3.3V hoặc 5V
- GND → GND
- A0 → GPIO34

- LCD 16x2 (I2C) ↔ ESP32:

- VCC → 5V
- GND → GND
- SDA → GPIO21
- SCL → GPIO22

- Buzzer ↔ ESP32:

- Cực dương (+) → GPIO25
- Cực âm (-) → GND

- LED (Cảnh báo) ↔ ESP32:

- Cực dương (+) → GPIO26 (qua điện trở 220 Ω)
- Cực âm (-) → GND

## V. Lập trình

### 1. Phần mềm phát triển trên Arduino IDE:

- Thư viện: BlynkSimpleEsp32.h, LiquidCrystal\_I2C.h.
- Đọc dữ liệu: analogRead(34) từ MQ-2.
- Điều khiển: digitalWrite(25, HIGH/LOW) cho buzzer, digitalWrite(26, HIGH/LOW) cho LED, lcd.print() hiển thị trên LCD.
- Thông báo: Blynk.notify() gửi cảnh báo qua Wi-Fi.
- Cấu hình: Kết nối Wi-Fi (ssid: "Wokwi-GUEST", pass: ""), Blynk (Auth Token), I2C (địa chỉ 0x27).

### 2. Phần lập trình C++ trên VS

Mã lập trình C++:

```

1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <LiquidCrystal_I2C.h>
4 #include <WiFi.h>
5 #include <WiFiClientSecure.h>
6 #include <UniversalTelegramBot.h>
7 #include <ArduinoJson.h>
8
9
10#define MQ2_PIN 34           // Cảm biến MQ-2
11#define BUZZER_PIN 25        // Còi cảnh báo
12#define LED_PIN 26           // Đèn báo
13
14#define BLYNK_TEMPLATE_ID "TMPL6j8Bmba8G"
15#define BLYNK_TEMPLATE_NAME "ESP32_Gas"
16#define BLYNK_AUTH_TOKEN "b7gAD5ZMm6HABv1rbmaarySvPch1055c"
17

```

```

18 #include <WiFiClient.h>
19 #include <BlynkSimpleEsp32.h>
20 // Thông tin WiFi
21 char ssid[] = "Wokwi-GUEST";
22 char pass[] = "";
23
24 // Thông tin Telegram
25 #define BOT_TOKEN "8066719718:AAE9Pi7EVp4hRwUz7bW7_tmsmuQTq84iD6M"
26 #define CHAT_ID "-4657079728"
27 WiFiClientSecure client;
28 UniversalTelegramBot bot(BOT_TOKEN, client);
29
30 LiquidCrystal_I2C lcd(0x27, 16, 2);
31 bool buzzerActive = false;
32 unsigned long lastAlertTime = 0;           // Thời gian gửi thông báo cuối
33 unsigned long lastTelegramTime = 0;         // Thời gian gửi Telegram cuối
34 unsigned long lastBuzzerToggle = 0;         // Thời gian bật/tắt còi
35 bool alertSent = false;
36 const int alertInterval = 2000;            // Gửi thông báo mỗi 2 giây
37 const int buzzerInterval = 500;             // Chu kỳ bật/tắt còi
38
39 void setup() {
40     Serial.begin(115200);
41
42     pinMode(MQ2_PIN, INPUT);
43     pinMode(BUZZER_PIN, OUTPUT);
44     pinMode(LED_PIN, OUTPUT);
45     digitalWrite(BUZZER_PIN, LOW); // Đảm bảo còi tắt lúc đầu
46
47     lcd.init();
48     lcd.backlight();
49     lcd.setCursor(0, 0);
50     lcd.print("Gas Detector ON");
51
52     // Kết nối WiFi
53     WiFi.begin(ssid, pass);
54     while (WiFi.status() != WL_CONNECTED) {
55         delay(500);
56         Serial.print(".");

```

```

57     }
58     Serial.println("\nWiFi Connected!");
59
60     client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Chứng chỉ Telegram
61     Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
62 }
63
64 void updateLCD(int gasValue, bool isAlert) {
65     lcd.setCursor(0, 0);
66     lcd.print("Muc Gas:      ");
67     lcd.setCursor(10, 0);
68     lcd.print("      ");
69     lcd.setCursor(10, 0);
70     lcd.print(gasValue);
71
72     lcd.setCursor(0, 1);
73     if (isAlert) {
74         lcd.print("CẢNH BÁO! Rò rỉ ");
75     } else {
76         lcd.print("Muc Gas an toàn ");
77     }
78 }
79
80 void handleBuzzer() {
81     if (buzzerActive && (millis() - lastBuzzerToggle >= buzzerInterval)) {
82         digitalWrite(BUZZER_PIN, !digitalRead(BUZZER_PIN)); // Đổi trạng
83 thái còi
84         lastBuzzerToggle = millis();
85     }
86 }
87
88 void sendNotifications(int gasValue, bool isAlert) {
89     unsigned long currentTime = millis();
90
91     if (currentTime - lastAlertTime >= alertInterval) {
92         Blynk.virtualWrite(V0, gasValue); // Gửi giá trị gas
93         if (isAlert) {
94             Blynk.virtualWrite(V1, "CẢNH BÁO: Rò rỉ khí gas!");
95             if (currentTime - lastTelegramTime >= alertInterval) {

```

```

96                 bot.sendMessage(CHAT_ID, "⚠️ CẢNH BÁO: Rò rỉ khí gas! Giá
97 trị: " + String(gasValue));
98                 lastTelegramTime = currentTime;
99             }
100         } else {
101             Blynk.virtualWrite(V1, "Mức gas an toàn");
102             if (!alertSent && currentTime - lastTelegramTime >=
103 alertInterval) {
104                 bot.sendMessage(CHAT_ID, "☑️ Mức gas an toàn. Giá trị: "
105 + String(gasValue));
106                 lastTelegramTime = currentTime;
107                 alertSent = true; // Chỉ gửi thông báo an toàn một lần
108             }
109         }
110     lastAlertTime = currentTime;
111 }
112 }
113
114 void loop() {
115     Blynk.run(); // Chạy Blynk
116
117     int gasValue = analogRead(MQ2_PIN);
118     Serial.print("Giá trị Gas: ");
119     Serial.println(gasValue);
120
121     if (gasValue > 1500) {
122         digitalWrite(LED_PIN, HIGH);
123         updateLCD(gasValue, true);
124         Serial.println("CẢNH BÁO! Rò rỉ khí gas!");
125
126         if (!buzzerActive) {
127             buzzerActive = true;
128             digitalWrite(BUZZER_PIN, HIGH); // Bật còi
129         }
130         alertSent = false; // Reset để gửi thông báo an toàn sau khi hết
131 cảnh báo
132         sendNotifications(gasValue, true);
133     } else {
134

```

```

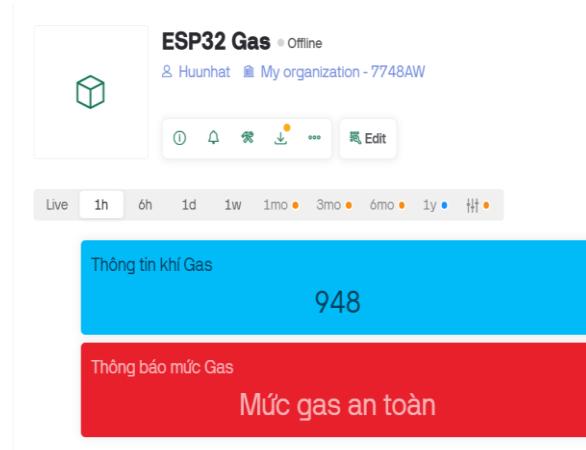
135     digitalWrite(LED_PIN, LOW);
136     updateLCD(gasValue, false);
137     Serial.println("Mức gas an toàn");
138
139     if (buzzerActive) {
140         buzzerActive = false;
141         digitalWrite(BUZZER_PIN, LOW); // Tắt còi
142     }
143
144     sendNotifications(gasValue, false);
145 }

handleBuzzer(); // Xử lý bật/tắt còi không chấn
}

```

### 3. Trên Web/Ứng dụng Blynk và Web/Ứng dụng Telegram

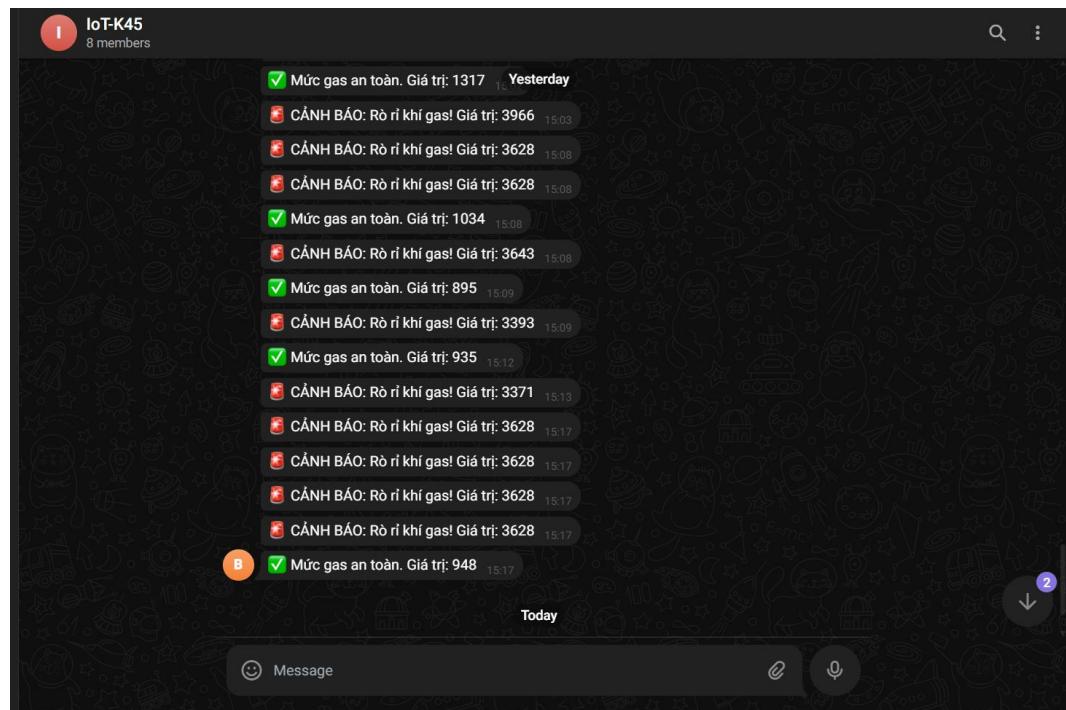
**Trạng thái bình thường khi dưới ngưỡng:**



**Hình 18.** Kết quả trả về trạng thái bình thường trên Blynk Web



**Hình 19.** Kết quả trả về trạng thái bình thường trên Blynk App

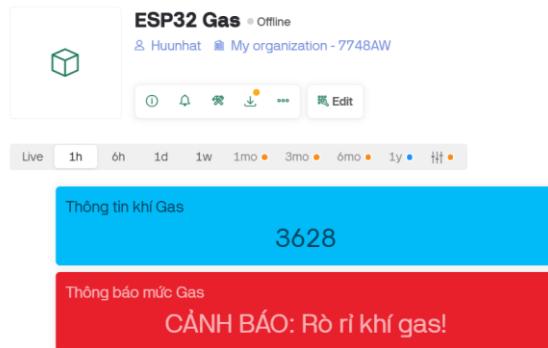


**Hình 20.** Kết quả trả về trạng thái bình thường trên Telegram Web

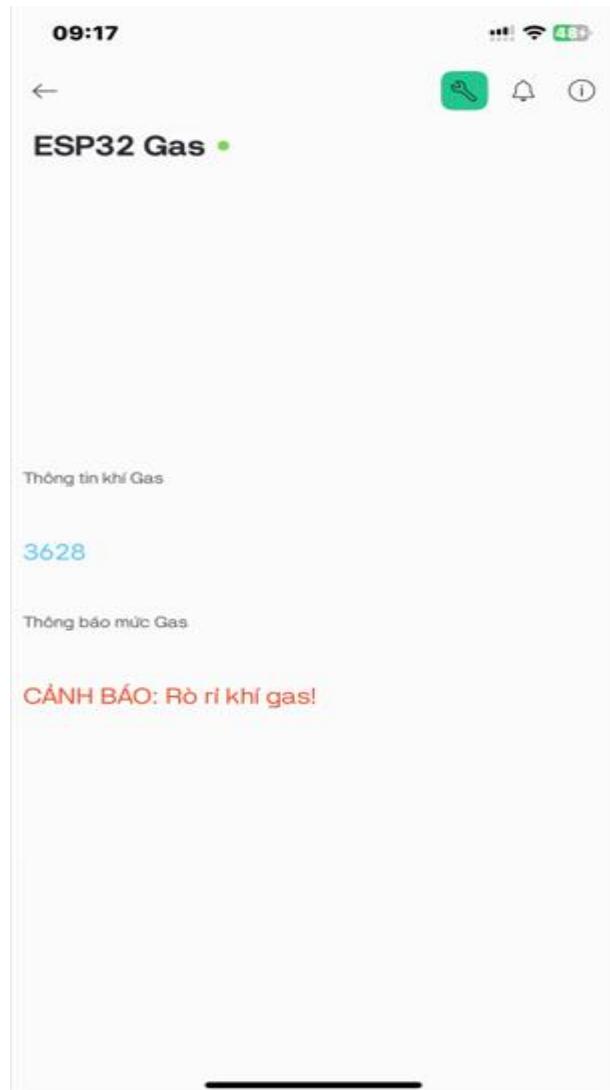


**Hình 21.** Kết quả trả về trạng thái bình thường trên Telegram App

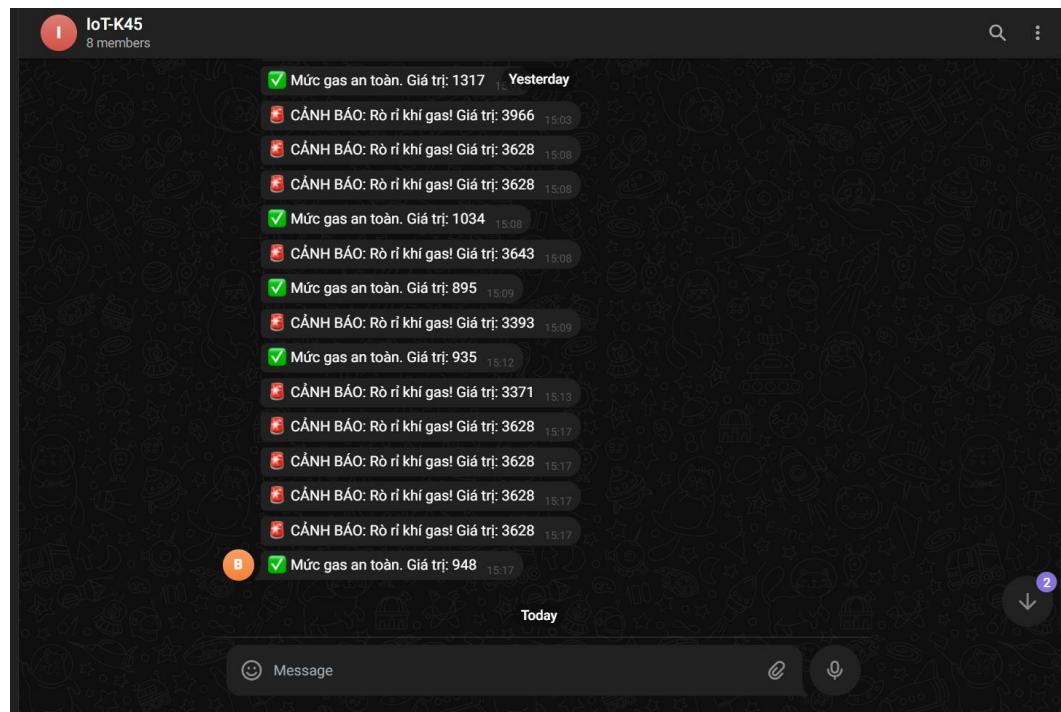
Trạng thái cảnh báo khi vượt ngưỡng:



**Hình 22.** Kết quả trả về trạng thái cảnh báo trên Blynk Web



**Hình 23.** Kết quả trả về trạng thái cảnh báo trên Blynk App



**Hình 24.** Kết quả trả về trạng thái cảnh báo trên Telegram Web



**Hình 25.** Kết quả trả về trạng thái cảnh báo trên Telegram App

## VI. Ưu điểm và hạn chế

- **Ưu điểm:**
  - **Đa dạng phương thức cảnh báo:**

Hệ thống tích hợp nhiều hình thức cảnh báo như âm thanh (buzzer), ánh sáng (LED đỏ), hiển thị trực quan (LCD 16x2), và thông báo từ xa (Blynk). Điều

này đảm bảo người dùng nhận được thông tin kịp thời qua nhiều kênh, cả tại chỗ và từ xa, tăng tính hiệu quả trong việc phát hiện rò rỉ khí gas.

- **Chi phí thấp và dễ triển khai:**

Các linh kiện như ESP32 DevKit-C V4, MQ-2, LCD I2C, buzzer, và LED đều có giá thành phải chăng, dễ mua trên thị trường. Việc kết nối đơn giản qua breadboard và lập trình trên Arduino IDE giúp người dùng, kể cả sinh viên hoặc người mới học IoT, dễ dàng triển khai hệ thống mà không cần thiết bị chuyên dụng hay kỹ năng cao cấp.

- **Tính linh hoạt và khả năng mở rộng:**

Nhờ tích hợp IoT qua Blynk, hệ thống có thể giám sát từ xa qua điện thoại hoặc web, phù hợp với xu hướng nhà thông minh. Ngoài ra, ESP32 hỗ trợ nhiều chân GPIO, cho phép mở rộng thêm các thiết bị như cảm biến khác hoặc relay để điều khiển van gas. Công cụ Wokwi hỗ trợ mô phỏng trước khi triển khai thực tế, giảm rủi ro sai sót.

- **Hạn chế:**

- **Độ chính xác của cảm biến MQ-2:**

MQ-2 nhạy với nhiều loại khí (LPG, CO, khói), dẫn đến khả năng báo động sai nếu có khí khác ngoài LPG trong môi trường. Để tập trung phát hiện LPG, cần hiệu chỉnh cảm biến định kỳ hoặc kết hợp thuật toán lọc tín hiệu, làm tăng độ phức tạp của hệ thống.

- **Phụ thuộc vào kết nối Wi-Fi:**

Tính năng thông báo qua Blynk yêu cầu kết nối Wi-Fi ổn định. Nếu mạng bị gián đoạn hoặc không có kết nối Internet, hệ thống mất khả năng gửi cảnh báo từ xa, chỉ còn hoạt động tại chỗ qua buzzer, LED và LCD, giảm hiệu quả trong các tình huống cần giám sát từ xa.

- **Phạm vi phát hiện hạn chế:**

MQ-2 chỉ phát hiện khí trong khoảng cách 5-10 mét vuông, tùy thuộc vào luồng không khí và vị trí đặt cảm biến. Trong không gian lớn hoặc khu vực thông thoáng, hệ thống có thể không phát hiện kịp thời rò rỉ khí ở xa, đòi hỏi bố trí nhiều cảm biến hơn để mở rộng phạm vi.

## VII. Ứng dụng thực tiễn

### - **Hộ gia đình:**

Hệ thống được lắp đặt trong bếp để phát hiện rò rỉ từ bình gas hoặc ống dẫn.

Khi nồng độ khí vượt ngưỡng, buzzer và LED cảnh báo tại chỗ, đồng thời thông báo qua Blynk giúp người dùng kịp thời xử lý, đặc biệt hữu ích khi không có ai ở nhà.

### - **Công nghiệp hóa chất:**

Trong các nhà máy sản xuất hoặc lưu trữ khí dễ cháy, hệ thống giám sát nồng độ khí liên tục qua LCD và gửi dữ liệu lên Blynk. Điều này hỗ trợ nhân viên kỹ thuật theo dõi từ xa, kết hợp với các biện pháp an toàn như ngắt nguồn khí tự động, giảm thiểu nguy cơ cháy nổ.

### - **Nhà thông minh:**

Tích hợp với hệ sinh thái nhà thông minh, hệ thống có thể kết nối với relay để tự động ngắt van gas khi phát hiện rò rỉ, hoặc kích hoạt quạt thông gió. Thông báo qua Blynk cho phép điều khiển từ xa, nâng cao tính an toàn và tiện nghi cho người dùng hiện đại.

# **PHẦN KẾT LUẬN**

## **I. Kết quả đạt được**

Đề tài “Hệ thống phát hiện rò rỉ khí gas với ESP32” đã thành công trong việc thiết kế và mô phỏng một giải pháp IoT hiệu quả, tận dụng vi điều khiển ESP32 DevKit-C V4 kết hợp với cảm biến MQ-2, màn hình LCD 16x2 I2C, buzzer, LED đỏ, ứng dụng Blynk và Telegram để đảm bảo an toàn trước nguy cơ rò rỉ khí gas. Hệ thống có khả năng phát hiện khí LPG trong phạm vi từ 300 đến 10.000 ppm, hiển thị thông tin trên màn hình LCD, phát tín hiệu cảnh báo tại chỗ qua buzzer và LED, đồng thời gửi thông báo từ xa qua Blynk và Telegram. Việc mô phỏng trên công cụ Wokwi đã giúp kiểm tra tính khả thi của hệ thống với chi phí thấp, thiết kế phần cứng đơn giản và phần mềm linh hoạt, dễ dàng triển khai trong các ứng dụng thực tế như hộ gia đình, nhà thông minh hoặc môi trường công nghiệp.

## **II. Hạn chế của đề tài**

Mặc dù đạt được những kết quả khả quan, hệ thống vẫn tồn tại một số hạn chế cần khắc phục:

- Độ chính xác của cảm biến MQ-2 có thể bị ảnh hưởng bởi các yếu tố môi trường như nhiệt độ, độ ẩm, hoặc sự hiện diện của các khí khác, dẫn đến khả năng xảy ra cảnh báo sai.
- Hệ thống phụ thuộc vào kết nối Wi-Fi để gửi thông báo từ xa qua Blynk và Telegram, do đó có thể không hoạt động hiệu quả trong môi trường không có mạng hoặc mạng không ổn định.
- Phạm vi phát hiện của cảm biến MQ-2 còn giới hạn, không phù hợp cho các không gian quá rộng hoặc phức tạp.
- Việc mô phỏng trên Wokwi tuy hiệu quả nhưng chưa phản ánh hoàn toàn các điều kiện thực tế, cần triển khai trên phần cứng vật lý để đánh giá chính xác hơn.

### **III. Hướng phát triển của đề tài**

Để nâng cao hiệu quả và tính ứng dụng của hệ thống, một số hướng phát triển trong tương lai có thể được xem xét:

- Tích hợp các cảm biến khí hiện đại hơn với độ nhạy và độ chính xác cao hơn, đồng thời bổ sung khả năng phân biệt các loại khí để giảm thiểu cảnh báo sai.
- Phát triển cơ chế hoạt động độc lập với Wi-Fi, chặng hạn sử dụng mạng GSM hoặc LoRa để gửi thông báo trong trường hợp không có kết nối Internet.
- Mở rộng phạm vi phát hiện bằng cách tích hợp nhiều cảm biến MQ-2 hoặc các cảm biến khác, kết hợp với thuật toán định vị để xác định chính xác vị trí rò rỉ khí gas trong không gian lớn.
- Ứng dụng trí tuệ nhân tạo (AI) và học máy (Machine Learning) để phân tích dữ liệu từ cảm biến, dự đoán nguy cơ rò rỉ khí gas trước khi xảy ra, từ đó nâng cao tính chủ động và độ tin cậy của hệ thống.
- Triển khai hệ thống trên phần cứng thực tế và thử nghiệm trong các môi trường thực để đánh giá hiệu suất, từ đó tối ưu hóa thiết kế và phần mềm trước khi ứng dụng rộng rãi.

# TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Nguyễn Văn Tân (2020). Ứng dụng IoT trong đời sống hiện đại, Tạp chí Khoa học Công nghệ, tập 5, số 02, tr. 15-20.
- [2] Trần Thị Minh (2018). Cảm biến khí và ứng dụng trong an toàn công nghiệp, Nxb Khoa học và Kỹ thuật, Hà Nội.

Tiếng Anh

- [3] Jones, R. (2019). LCD Displays in Embedded Systems, Embedded World, vol. 15, pp. 30-35.
  - [4] Smith, J. (2019). Introduction to Buzzers and Sound Devices, Electronics Weekly, vol. 12, pp. 45-50.
  - [5] Brown, P. (2020). Basic Electronics Components, McGraw-Hill, London.
  - [6] Brown, T. (2021). IoT Platforms: A Comprehensive Guide, Springer, New York.
  - [7] CodeCraft Team (2022). Wokwi: Online Electronics Simulator, <https://wokwi.com/>.
- Truy cập ngày 26/3/2025.