

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



**TÊN ĐỀ TÀI TIỂU LUẬN:
HỆ THỐNG PHÁT HIỆN RÒ RỈ KHÍ GAS VỚI ESP32**

PHÁT TRIỂN ỨNG DỤNG IOT – 2024-2025.2.TIN4024.005

GIẢNG VIÊN HƯỚNG DẪN: VÕ VIỆT DŨNG

HUẾ, THÁNG 4 NĂM 2025

LỜI MỞ ĐẦU

Trong thời đại công nghiệp hóa và hiện đại hóa, việc đảm bảo an toàn trong sinh hoạt và sản xuất là một trong những yếu tố then chốt góp phần nâng cao chất lượng cuộc sống. Một trong những hiểm họa tiềm tàng và phổ biến hiện nay là tình trạng rò rỉ khí gas. Các vụ tai nạn do khí gas gây ra không chỉ ảnh hưởng nghiêm trọng đến tài sản mà còn đe dọa tính mạng con người. Chính vì vậy, việc phát triển những hệ thống phát hiện rò rỉ khí gas thông minh và hiệu quả là vô cùng cần thiết.

Với sự phát triển vượt bậc của công nghệ, đặc biệt là công nghệ IoT (Internet of Things - Internet vạn vật), việc ứng dụng các vi điều khiển như ESP32 để xây dựng các hệ thống cảnh báo an toàn trở nên dễ dàng và tiết kiệm chi phí hơn bao giờ hết. Đề tài này tập trung thiết kế và triển khai một Hệ thống phát hiện rò rỉ khí gas sử dụng ESP32, tích hợp các thành phần phần cứng như cảm biến khí MQ-6, màn hình OLED để hiển thị thông tin, còi và đèn LED để cảnh báo người dùng kịp thời.

Bài tiểu luận này sẽ trình bày quá trình từ cơ sở lý thuyết, thiết kế phần cứng, lập trình điều khiển, cho đến đánh giá hiệu quả và đề xuất hướng phát triển trong tương lai. Mong rằng sản phẩm này sẽ đóng góp một phần nhỏ vào việc đảm bảo an toàn cho hộ gia đình, nhà xưởng, và các môi trường dễ xảy ra sự cố rò rỉ khí gas.

MỤC LỤC

| | |
|---|----|
| LỜI MỞ ĐẦU | 1 |
| MỤC LỤC | 2 |
| 1. Giới thiệu chung | 4 |
| 1.1. Khái quát về Internet of Things (IoT)..... | 4 |
| 1.2. Giới thiệu về ESP32 | 4 |
| 1.3. Cảm biến khí gas MQ-6 | 4 |
| 1.4. Màn hình Oled SSD1306 | 5 |
| 1.5. Còi báo động (Buzzer) và đèn LED | 5 |
| 2. Thiết kế hệ thống | 6 |
| 2.1. Sơ đồ khối hệ thống | 6 |
| 2.2. Nguyên lý hoạt động..... | 6 |
| 2.3. Chu trình xử lý của ESP32 | 6 |
| 3. Mô hình thiết kế | 8 |
| 3.1. Bảng sơ đồ kết nối phần cứng..... | 8 |
| 3.2. Sơ đồ Wokwi | 8 |
| 3.3. Giao diện Blynk | 11 |
| 4. Lập trình cho hệ thống | 12 |
| 4.1. Chương trình tổng quát..... | 12 |
| 4.1.1. Mục tiêu của chương trình | 12 |
| 4.1.2. Cấu trúc chương trình tổng quát..... | 12 |
| 4.1.3. Luồng hoạt động của hệ thống | 13 |
| 4.1.4. Mô hình dữ liệu kết nối Blynk | 13 |
| 4.1.5. Ưu điểm của mô hình..... | 13 |

| | |
|---|----|
| 4.2. Khai báo thông tin Blynk | 14 |
| 4.3. Thư viện và cấu hình OLED. | 14 |
| 4.4. Khai báo chân phần cứng | 14 |
| 4.4. Khai báo ngưỡng khí Gas | 15 |
| 4.5. Thông tin Wi-Fi | 15 |
| 4.6. Hàm setup() | 15 |
| 4.7. Hàm loop() | 15 |
| 4.8. Hiển thị lên màn hình OLED | 16 |
| 4.9. Điều kiện cảnh báo | 16 |
| 4.10. Chu kỳ lặp. | 16 |
| 4.11. Tổng kết chức năng chính. | 17 |
| KẾT LUẬN | 17 |
| TÀI LIỆU THAM KHẢO | 20 |

1. Giới thiệu chung

1.1. Khái quát về Internet of Things (IoT)

Internet of Things (IoT) là khái niệm chỉ mạng lưới các thiết bị vật lý được kết nối với nhau thông qua internet để thu thập và trao đổi dữ liệu. Các thiết bị IoT có thể là cảm biến, vi điều khiển, thiết bị điện tử, phần mềm,... và có khả năng nhận dạng, định vị, giám sát và điều khiển trong thời gian thực. Ứng dụng của IoT ngày càng mở rộng trong nhiều lĩnh vực như:

- Nhà thông minh: điều khiển thiết bị điện tử từ xa, phát hiện rò rỉ nước, khí gas.
- Y tế: giám sát sức khỏe từ xa, hỗ trợ người già và bệnh nhân mãn tính.
- Nông nghiệp thông minh: theo dõi độ ẩm đất, nhiệt độ, tưới tiêu tự động.
- Giao thông thông minh: giám sát phương tiện, quản lý bãi đỗ xe.

Trong đề tài này, IoT sẽ đóng vai trò quan trọng trong việc giám sát khí gas và đưa ra cảnh báo sớm thông qua các cảm biến và thiết bị xử lý thông minh.

1.2. Giới thiệu về ESP32

ESP32 là một vi điều khiển tích hợp Wi-Fi và Bluetooth do công ty Espressif Systems phát triển. Đây là phiên bản nâng cấp của ESP8266 với nhiều tính năng mạnh mẽ hơn như:

- CPU lõi kép 32-bit, tốc độ lên đến 240 MHz.
- Tích hợp Wi-Fi chuẩn IEEE 802.11 b/g/n và Bluetooth v4.2.
- Hỗ trợ nhiều giao tiếp như UART, SPI, I2C, PWM, ADC, DAC.
- Có khả năng tiết kiệm năng lượng với nhiều chế độ ngủ.

ESP32 rất phù hợp trong các dự án IoT, tự động hóa, hệ thống cảnh báo, vì vừa mạnh mẽ, vừa tiết kiệm chi phí và dễ lập trình. Trong đề tài này, ESP32 giữ vai trò trung tâm trong việc thu thập dữ liệu từ cảm biến MQ-6, xử lý tín hiệu và điều khiển các thiết bị đầu ra như OLED, LED, Buzzer.

1.3. Cảm biến khí gas MQ-6

Cảm biến khí gas MQ-6 là một loại cảm biến hóa học được sử dụng để phát hiện sự hiện diện của khí gas dễ cháy như LPG, isobutane, propane,... MQ-6 hoạt động dựa trên nguyên lý thay đổi điện trở của vật liệu bán dẫn trong môi trường có khí gas.

- Thông số kỹ thuật:

- Điện áp hoạt động: 5V.
- Dòng tiêu thụ: ~150mA.
- Phát hiện LPG, butane, propane ở nồng độ 200~10000 ppm.

Cảm biến MQ-6 có thể xuất tín hiệu analog hoặc digital tùy cấu hình, giúp dễ dàng tích hợp với vi điều khiển. Trong hệ thống này, tín hiệu analog được lấy qua chân ADC của ESP32 để đánh giá nồng độ khí và kích hoạt cảnh báo khi vượt ngưỡng an toàn.

1.4. Màn hình Oled SSD1306

Màn hình OLED SSD1306 là loại màn hình nhỏ gọn với độ phân giải 128x64 pixel, giao tiếp qua giao thức I2C hoặc SPI. Màn hình OLED có ưu điểm:

- Kích thước nhỏ, tiêu thụ điện năng thấp.
- Hiển thị rõ ràng trong điều kiện thiếu sáng.
- Có thể hiển thị văn bản, đồ họa đơn giản.

Trong dự án này, màn hình OLED được dùng để hiển thị giá trị nồng độ khí đo được từ cảm biến MQ-6, giúp người dùng theo dõi dễ dàng hơn.

1.5. Còi báo động (Buzzer) và đèn LED

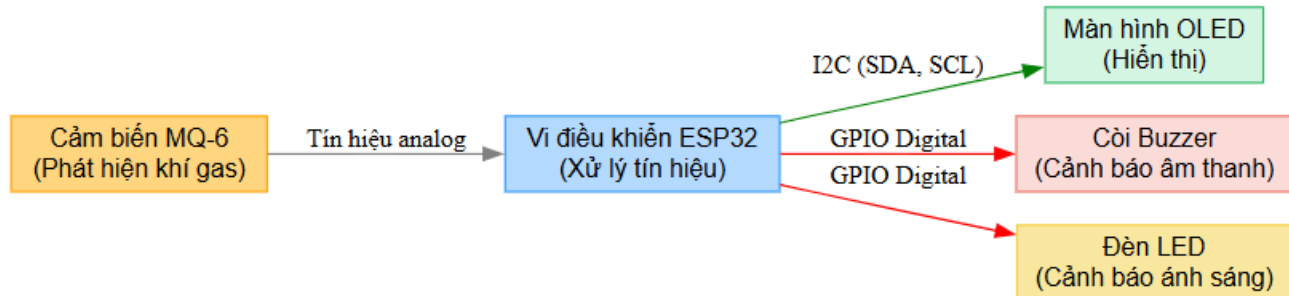
Còi báo động và đèn LED là các thiết bị cảnh báo đơn giản nhưng hiệu quả. Khi hệ thống phát hiện nồng độ khí gas vượt quá ngưỡng nguy hiểm, buzzer sẽ phát ra âm thanh và đèn LED sẽ nhấp nháy để báo hiệu khẩn cấp.

- Buzzer hoạt động như một thiết bị phát âm thanh đơn tần.
- Đèn LED đỏ dùng để cảnh báo trực quan.

Hai thiết bị này được điều khiển trực tiếp từ ESP32 bằng cách sử dụng tín hiệu digital. Tùy mức độ nghiêm trọng, hệ thống có thể cấu hình cảnh báo bằng âm thanh và ánh sáng liên tục hoặc ngắt quãng.

2. Thiết kế hệ thống

2.1. Sơ đồ khối hệ thống



2.2. Nguyên lý hoạt động

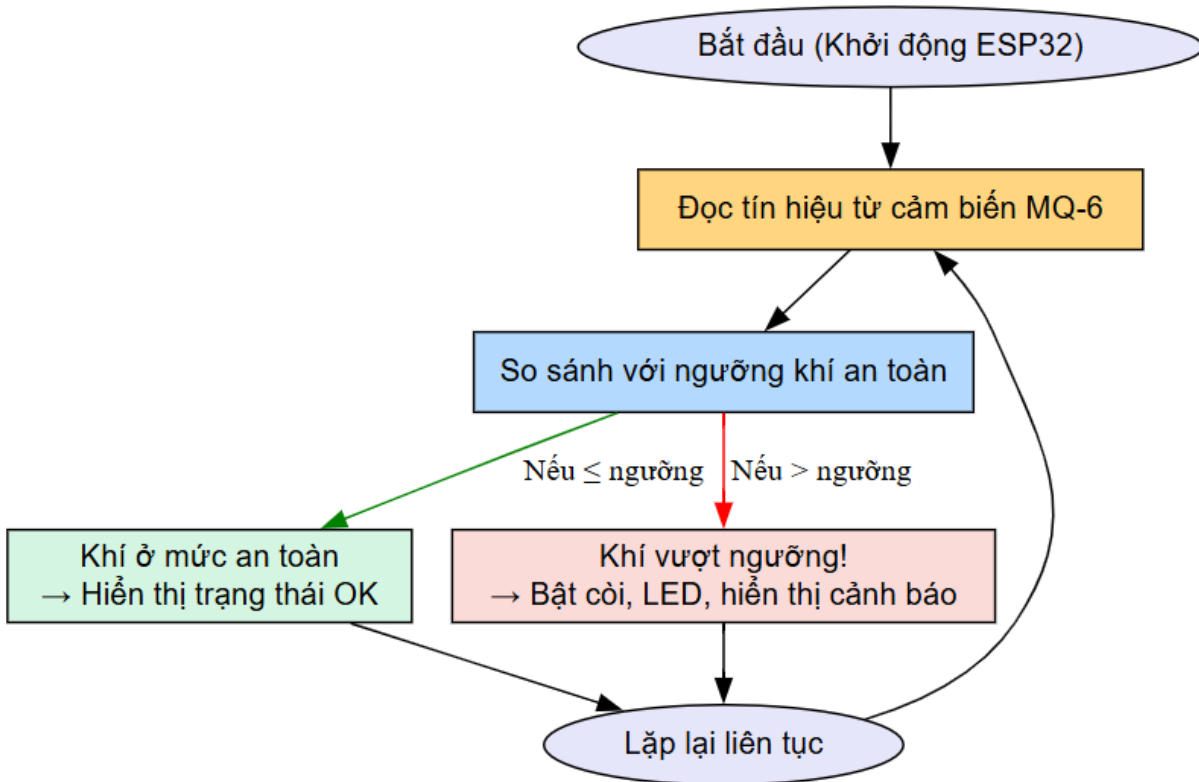
Cảm biến MQ-6 liên tục đo nồng độ khí gas trong môi trường và gửi tín hiệu analog về ESP32. Vi điều khiển ESP32 sẽ đọc tín hiệu này qua chân ADC, sau đó xử lý và so sánh với ngưỡng an toàn đã thiết lập. Nếu nồng độ khí vượt quá ngưỡng, hệ thống sẽ kích hoạt còi buzzer để phát âm thanh cảnh báo, đèn LED đỏ nhấp nháy để thu hút sự chú ý, đồng thời màn hình OLED hiển thị giá trị nồng độ khí kèm theo cảnh báo. Ngược lại, nếu nồng độ khí ở mức an toàn, OLED sẽ hiển thị trạng thái bình thường.

2.3. Chu trình xử lý của ESP32

ESP32 thực hiện chu trình xử lý chính theo các bước tuần hoàn liên tục, đảm bảo hệ thống hoạt động giám sát thời gian thực:

1. **Khởi tạo hệ thống:** Khởi tạo các chân I/O, thiết bị ngoại vi như màn hình OLED, cảm biến MQ-6, còi và LED. Thiết lập các tham số ngưỡng cảnh báo.
2. **Đọc tín hiệu cảm biến:** ESP32 đọc dữ liệu analog từ cảm biến MQ-6 thông qua chân ADC, chuyển đổi tín hiệu thành giá trị nồng độ khí gas (ppm).
3. **Xử lý và phân tích dữ liệu:** Giá trị đo được được so sánh với ngưỡng an toàn (ví dụ: 300 ppm). Nếu vượt ngưỡng, hệ thống sẽ chuyển sang chế độ cảnh báo.
4. **Cảnh báo người dùng:**

- Nếu **vượt ngưỡng**, ESP32 sẽ kích hoạt còi Buzzer và đèn LED, đồng thời cập nhật cảnh báo trên màn hình OLED.
 - Nếu **dưới ngưỡng**, hệ thống hiển thị trạng thái "An toàn" trên OLED và tắt thiết bị cảnh báo.
5. **Lặp lại:** Chu trình quay lại bước 2 sau một khoảng trễ nhỏ (ví dụ: 500ms), đảm bảo theo dõi liên tục.



3. Mô hình thiết kế

3.1. Bảng sơ đồ kết nối phần cứng

| Thiết bị | Chân thiết bị | Kết nối đến ESP32 (chân) | Ghi chú |
|----------|---------------|--------------------------|------------------------|
| MQ-6 | OUT | 3V3 | Cấp nguồn |
| | GND | GND | Nối đất |
| | VCC | GPIO35 | Đọc tín hiệu analog |
| OLED | VCC | 3V3 | Cấp nguồn |
| | GND | GND | Nối đất |
| | SCL | GPIO22 | Tín hiệu I2C – Clock |
| | SDA | GPIO21 | Tín hiệu I2C - Data |
| Buzzer | Chân 2 | GPIO26 | Điều khiển còi |
| | Chân 1 | GND | Nối đất |
| LED đỏ | Chân Anode | GPIO5 | Điều khiển bật/tắt đèn |
| | Chân Cathode | GND qua R 220 Ω | Có điện trở hạn dòng |

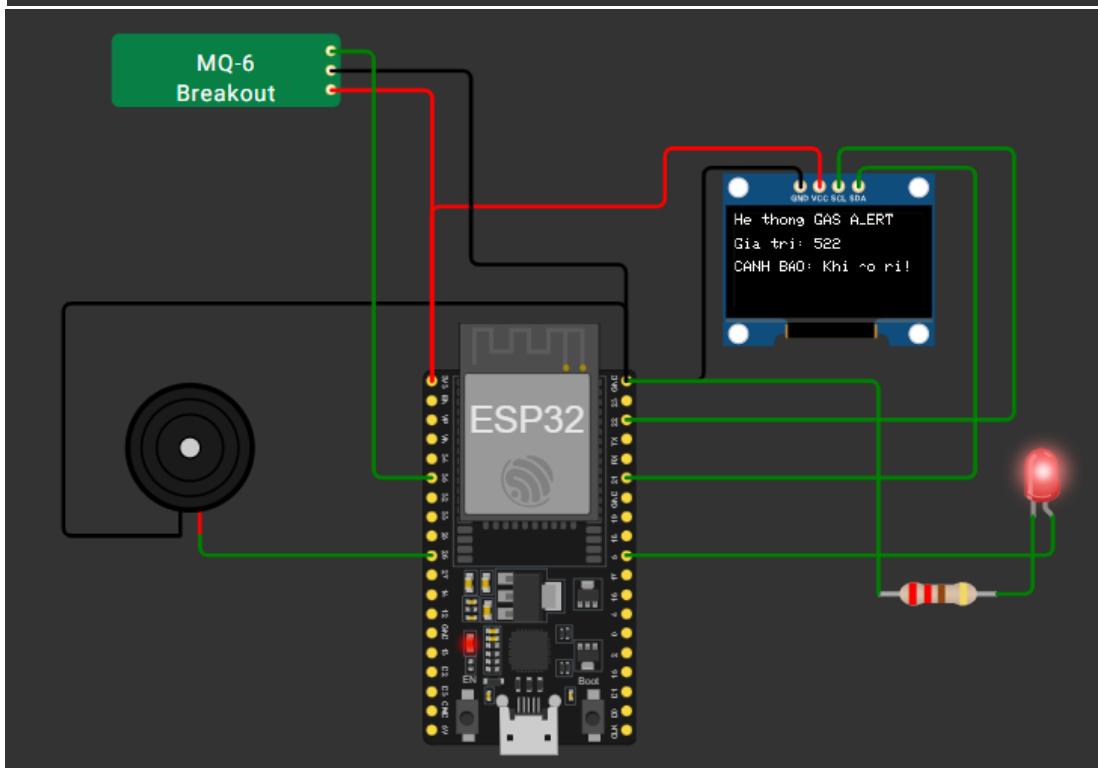
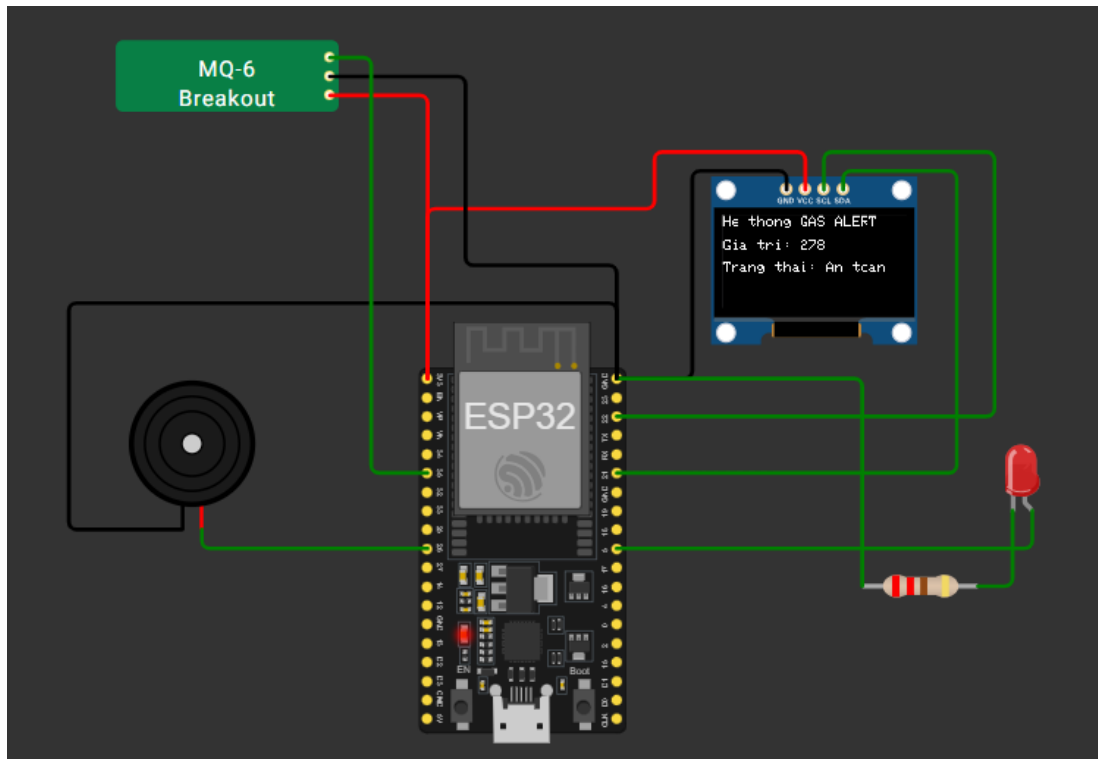
3.2. Sơ đồ Wokwi

```
{
  "version": 1,
  "author": "TKH",
  "editor": "wokwi",
  "parts": [
    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": -33.56,
      "attrs": {} },
    {
      "type": "board-ssd1306",
      "id": "oled1",
      "top": -73.66,
      "left": 115.43,
      "attrs": { "i2cAddress": "0x3c" }
    },
    {
      "type": "wokwi-buzzer",
      "id": "bz1",
      "top": 21.6,
      "left": -180.6,
      "attrs": { "volume": "0.1" }
    },
    { "type": "wokwi-led", "id": "led1", "top": 54, "left": 253.4, "attrs": {
      "color": "red" } },
    {
      "type": "wokwi-resistor",
```

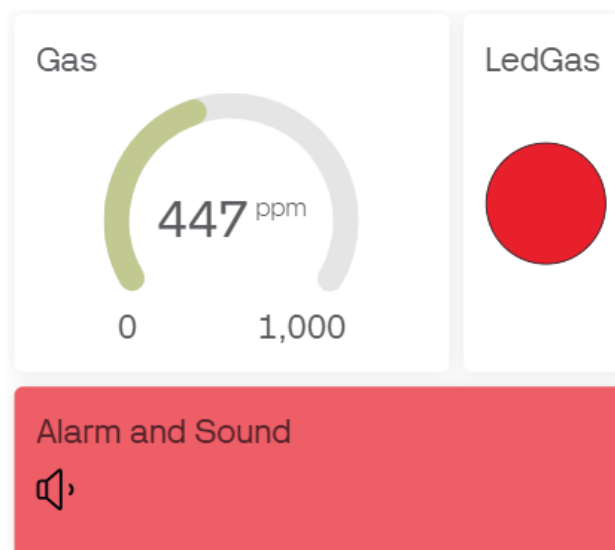
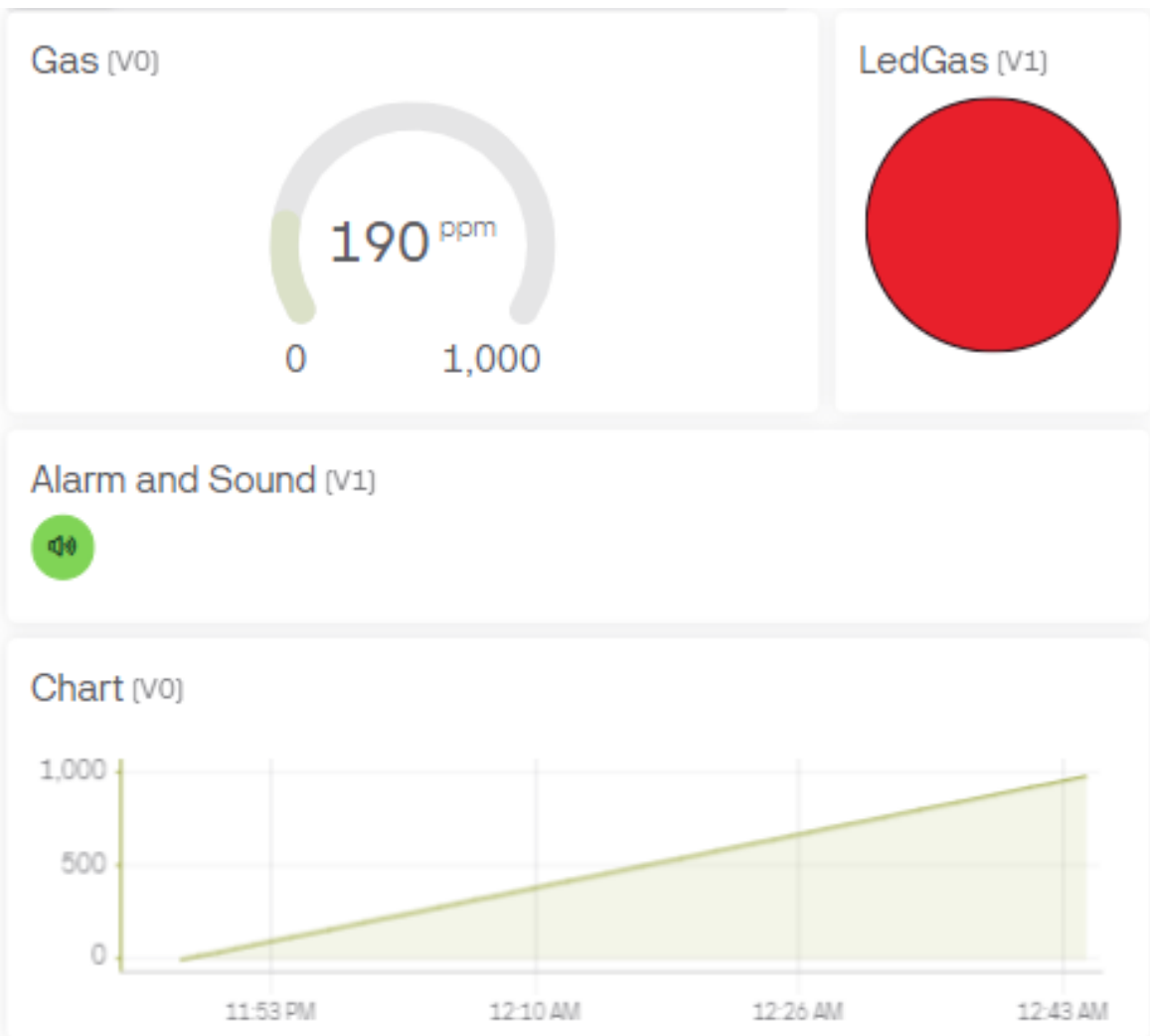
```

    "id": "r1",
    "top": 128.75,
    "left": 192,
    "attrs": { "value": "220" }
  },
  {
    "type": "chip-mq6",
    "id": "chip1",
    "top": -142.98,
    "left": -187.2,
    "attrs": { "simulateAnalogOutput": "true", "value": "300" }
  }
],
"connections": [
  [ "esp:TX", "$serialMonitor:RX", "", [ ] ],
  [ "esp:RX", "$serialMonitor:TX", "", [ ] ],
  [ "bz1:1", "esp:GND.2", "black", [ "h-57.6", "v-115.2", "h278.4" ] ],
  [ "bz1:2", "esp:26", "green", [ "v0" ] ],
  [ "oled1:GND", "esp:GND.2", "black", [ "v-9.6", "h-48", "v115.2" ] ],
  [ "oled1:VCC", "esp:3V3", "red", [ "v-19.2", "h-76.65", "v28.8", "h-115.35" ] ],
],
[
  [ "oled1:SCL", "esp:22", "green", [ "v-19.2", "h86.7", "v134.4" ] ],
  [ "oled1:SDA", "esp:21", "green", [ "v-9.6", "h57.67", "v153.6" ] ],
  [ "led1:C", "r1:2", "green", [ "v0" ] ],
  [ "r1:1", "esp:GND.2", "green", [ "v0" ] ],
  [ "led1:A", "esp:5", "green", [ "v0" ] ],
  [ "chip1:VCC", "esp:3V3", "red", [ "h49.81", "v144" ] ],
  [ "chip1:GND", "esp:GND.2", "black", [ "h69.01", "v96", "h76.8" ] ],
  [ "chip1:OUT", "esp:35", "green", [ "h21.01", "v211.2" ] ]
],
"dependencies": {}
}

```



3.3. Giao diện Blynk



4. Lập trình cho hệ thống

4.1. Chương trình tổng quát

Phần mềm được lập trình bằng Visual Code Studio với ngôn ngữ C/C++. Chương trình có nhiệm vụ:

- Khởi tạo các thiết bị (OLED, Buzzer, LED, cảm biến).
- Đọc dữ liệu analog từ cảm biến MQ-6.
- Hiển thị giá trị nồng độ khí gas lên màn hình OLED.
- So sánh nồng độ khí với ngưỡng nguy hiểm.
- Kích hoạt còi và đèn cảnh báo khi cần thiết.
- Gửi các giá trị lên ứng dụng Blynk.

4.1.1. Mục tiêu của chương trình

Xây dựng một hệ thống giám sát và cảnh báo rò rỉ khí gas dùng cảm biến MQ-6, có chức năng:

- Đọc nồng độ khí gas qua cảm biến.
- Hiển thị giá trị đo lên màn hình OLED.
- Phát cảnh báo (buzzer + đèn LED) khi vượt ngưỡng.
- Gửi dữ liệu thời gian thực lên Blynk để theo dõi từ xa.
- Cho phép bật/tắt cảnh báo thủ công từ điện thoại qua Blynk.

4.1.2. Cấu trúc chương trình tổng quát

| STT | Thành phần | Mô tả |
|-----|-------------------|---|
| 1 | Khai báo thư viện | Sử dụng thư viện cho OLED, WiFi, Blynk, cảm biến, v.v. |
| 2 | Biến cấu hình | Khai báo thông tin WiFi, Blynk, chân kết nối, ngưỡng cảnh báo. |
| 3 | Hàm setup() | Khởi tạo thiết bị: kết nối Blynk, OLED, cài đặt chân vào/ra. |
| 4 | Hàm loop() | Đọc cảm biến, xử lý cảnh báo, hiển thị OLED, gửi dữ liệu lên Blynk. |

| | | |
|---|-------------------|--|
| 5 | Hàm BLYNK_WRITE() | Nhận lệnh từ app Blynk (bật/tắt cảnh báo từ xa). |
|---|-------------------|--|

4.1.3. Luồng hoạt động của hệ thống

1. Khởi động

Kết nối WiFi → kết nối Blynk → khởi động OLED → hiển thị thông báo khởi động.

2. Đo nồng độ khí

- Dùng analogRead() từ MQ-6 để đo nồng độ khí.
- Giá trị thường nằm trong khoảng 0 – 1023 (ADC 10-bit).

3. Hiển thị giá trị lên OLED

Ghi “Nồng độ khí: xxx ppm” và trạng thái an toàn hay rò rỉ.

4. Xử lý cảnh báo

- Nếu nồng độ > ngưỡng (ví dụ 400) hoặc người dùng bật cảnh báo từ app → bật buzzer + đèn.
- Ngược lại, tắt cảnh báo.

5. Gửi dữ liệu lên app Blynk

- Gửi giá trị nồng độ lên V0.
- Nhận điều khiển bật/tắt từ V1.

4.1.4. Mô hình dữ liệu kết nối Blynk

| Tên thiết bị | Pin ảo | Kiểu | Vai trò |
|--------------|--------|---------|--|
| Gas | V0 | Double | Gửi dữ liệu nồng độ khí (sensor) |
| LedGas | V1 | Integer | Nhận điều khiển từ người dùng (switch) |

4.1.5. Ưu điểm của mô hình

- Tự động + thủ công: hệ thống có thể tự bật cảnh báo hoặc nhận lệnh điều khiển từ người dùng.
- Tính thời gian thực: truyền dữ liệu và điều khiển qua Internet tức thì.
- Giao diện dễ quan sát: OLED hiển thị trực tiếp tại thiết bị; Blynk hiển thị từ xa.

4.2. Khai báo thông tin Blynk

```
#define BLYNK_TEMPLATE_ID "TMPL6nzP1HY23"  
#define BLYNK_TEMPLATE_NAME "ESP32GasSensor"  
#define BLYNK_AUTH_TOKEN "G_P5lq6s4ipaVG0PxpTTDAYWSrVhmSCx"
```

Mục đích: Liên kết với nền tảng Blynk IoT.

Chi tiết:

- BLYNK_TEMPLATE_ID: ID mẫu từ Blynk.
- BLYNK_TEMPLATE_NAME: Tên thiết bị hiển thị trên ứng dụng.
- BLYNK_AUTH_TOKEN: Mã xác thực để kết nối thiết bị với tài khoản Blynk của bạn.

4.3. Thư viện và cấu hình OLED

```
#include <WiFi.h>  
#include <BlynkSimpleEsp32.h>  
#include <Wire.h>  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>
```

Mục đích:

- Kết nối WiFi + Blynk.
- Điều khiển màn hình OLED SSD1306.

Thiết lập độ phân giải màn hình OLED:

```
#define SCREEN_WIDTH 128  
#define SCREEN_HEIGHT 64  
#define OLED_RESET -1
```

Tạo đối tượng display để thao tác với màn hình:

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

4.4. Khai báo chân phần cứng

```
#define MQ6_PIN 35  
#define BUZZER_PIN 26  
#define LED_PIN 5
```

- MQ6_PIN: chân đọc tín hiệu khí gas từ cảm biến MQ-6.
- BUZZER_PIN: cảnh báo âm thanh.
- LED_PIN: cảnh báo ánh sáng.

4.4. Khai báo ngưỡng khí Gas

```
#define GAS_THRESHOLD 400
```

Ngưỡng này là giá trị analog từ MQ-6. Nếu vượt qua thì cảnh báo.

4.5. Thông tin Wi-Fi

```
const char* ssid = "Wokwi-GUEST";  
const char* password = "";
```

Kết nối WiFi của ESP32 trên mô phỏng Wokwi (không cần mật khẩu).

4.6. Hàm setup()

Khởi tạo Serial để in thông tin lên terminal (debug):

```
void setup() {  
    Serial.begin(115200);
```

Kết nối Blynk + Wi-Fi:

```
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
```

Cài đặt chế độ hoạt động của các chân: đọc tín hiệu, xuất cảnh báo:

```
pinMode(MQ6_PIN, INPUT);  
pinMode(BUZZER_PIN, OUTPUT);  
pinMode(LED_PIN, OUTPUT);
```

Khởi động màn hình OLED, địa chỉ 0x3C là địa chỉ mặc định I2C, nếu không tìm thấy OLED, dừng chương trình tại while(true):

```
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {  
    Serial.println(F("Không tìm thấy OLED!"));  
    while (true);  
}
```

Hiện thị thông báo khởi động hệ thống trong 1 giây:

```
display.println("Khoi dong he thong...");  
display.display();  
delay(1000);
```

4.7. Hàm loop()

Đọc giá trị khí gas (analog từ 0–4095 trên ESP32):

```
int gasValue = analogRead(MQ6_PIN);
```

Gửi giá trị đọc được lên Virtual Pin V0 trên Blynk để hiển thị:

```
Blynk.virtualWrite(V0, gasValue);
```


4.8. Hiển thị lên màn hình OLED

Xóa màn hình và in tiêu đề + giá trị khí gas hiện tại:

```
display.clearDisplay();
display.setCursor(0, 0);
display.println("He thong GAS ALERT");
display.setCursor(0, 16);
display.print("Nong do khi: ");
display.println(gasValue);
```

4.9. Điều kiện cảnh báo

Nếu giá trị khí gas vượt ngưỡng:

- Bật còi (buzzer).
- Bật đèn LED.
- Gửi tín hiệu bật LED ảo lên Blynk (V1 = 1).
- Hiển thị cảnh báo lên OLED.

```
if (gasValue >= GAS_THRESHOLD) {
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(LED_PIN, HIGH);
    display.setCursor(0, 32);
    display.setTextColor(SSD1306_WHITE);
    display.println("CANH BAO: Khi ro ri!");
    Blynk.virtualWrite(V1, 1);
}
```

Nếu giá trị an toàn:

- Tắt cảnh báo.
- Gửi tín hiệu tắt LED ảo (V1 = 0).
- Hiển thị "An toàn" lên OLED.

```
else {
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(LED_PIN, LOW);
    display.setCursor(0, 32);
    display.setTextColor(SSD1306_WHITE);
    display.println("Trang thai: An toan");
    Blynk.virtualWrite(V1, 0);
}
```

4.10. Chu kỳ lặp

Cập nhật nội dung OLED và lặp lại mỗi 0.5 giây:

```
display.display(); delay(500);
```

4.11. Tổng kết chức năng chính

| Chức năng | Mô tả |
|--------------------------------|-------------------------------------|
| Đọc giá trị từ MQ-6 | analogRead từ chân 35 |
| Hiển thị lên OLED | Sử dụng thư viện Adafruit SSD1306 |
| Cảnh báo LED + Buzzer | Khi vượt ngưỡng 400 |
| Gửi dữ liệu đến ứng dụng Blynk | Sử dụng V0 (giá trị), V1 (cảnh báo) |

- Tổng hợp code:

```
#define BLYNK_TEMPLATE_ID "TMPL6nzP1HY23"
#define BLYNK_TEMPLATE_NAME "ESP32GasSensor"
#define BLYNK_AUTH_TOKEN "G_P5lq6s4ipaVG0PxPTDAYWSrVhmSCx"
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
#define MQ6_PIN 35
#define BUZZER_PIN 26
#define LED_PIN 5
#define GAS_THRESHOLD 400
const char* ssid = "Wokwi-GUEST";
const char* password = "";
void setup() {
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  pinMode(MQ6_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Không tìm thấy OLED!"));
    while (true);
  }
  display.println("Khoi dong he thong...");
  display.display();
  delay(1000);
}
void loop(){
```

```

int gasValue = analogRead(MQ6_PIN);
Blynk.virtualWrite(V0, gasValue);
display.clearDisplay();
display.setCursor(0, 0);
display.println("He thong GAS ALERT");
display.setCursor(0, 16);
display.print("Nong do khi: ");
display.println(gasValue);
if (gasValue >= GAS_THRESHOLD) {
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(LED_PIN, HIGH);
    display.setCursor(0, 32);
    display.setTextColor(SSD1306_WHITE);
    display.println("CANH BAO: Khi ro ri!");
    Blynk.virtualWrite(V1, 10);
}
else {
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(LED_PIN, LOW);
    display.setCursor(0, 32);
    display.setTextColor(SSD1306_WHITE);
    display.println("Trang thai: An toan");
    Blynk.virtualWrite(V1, 1);
}
display.display(); delay(500);
}

```

KẾT LUẬN

Đề tài “Xây dựng hệ thống cảnh báo rò rỉ khí gas sử dụng ESP32 kết hợp Blynk và hiển thị OLED” đã triển khai thành công một mô hình ứng dụng IoT đơn giản nhưng hiệu quả, có khả năng phát hiện và cảnh báo khí gas rò rỉ theo thời gian thực. Hệ thống sử dụng cảm biến MQ-6 để thu thập dữ liệu nồng độ khí gas, ESP32 làm trung tâm xử lý, đồng thời truyền dữ liệu đến ứng dụng Blynk và hiển thị thông số trực tiếp trên màn hình OLED. Ngoài ra, LED và còi được sử dụng làm bộ phận cảnh báo tức thì tại chỗ, giúp tăng khả năng phản ứng nhanh với các sự cố nguy hiểm. Việc tích hợp phần cứng với nền tảng IoT như Blynk không những đảm bảo sự tiện lợi trong giám sát mà còn mở ra nhiều hướng phát triển mở rộng cho các hệ thống cảnh báo thông minh.

Kết quả đạt được cho thấy tiềm năng lớn của việc ứng dụng công nghệ IoT trong lĩnh vực an toàn và giám sát môi trường sống. Với thiết kế linh hoạt và chi phí hợp lý, hệ thống có thể được triển khai tại nhiều địa điểm như hộ gia đình, nhà hàng, khu công nghiệp, v.v. Trong tương lai, có thể tích hợp thêm các tính năng như gửi cảnh báo qua SMS, kết nối với hệ thống chữa cháy tự động, hay kết hợp nhiều loại cảm biến để mở rộng phạm vi giám sát. Đề tài là minh chứng rõ ràng cho vai trò thiết thực của công nghệ trong việc nâng cao chất lượng cuộc sống và bảo vệ con người trước các rủi ro tiềm ẩn từ môi trường.

TÀI LIỆU THAM KHẢO

1. Tài liệu học tập, source code tham khảo của thầy Võ Việt Dũng, <https://github.com/vvdung-husc/2024-2025.2.TIN4024.005>
2. Internet of Things – Những nền tảng cơ bản và ứng dụng, Nguyễn Việt Hùng (2019), NXB Bách Khoa Hà Nội.
3. Lập trình ESP32 ứng dụng trong IoT, Trần Văn Tới, Nguyễn Văn Thanh (2020), Tài liệu nội bộ Trường Đại học Bách Khoa TP.HCM.
4. Hướng dẫn lập trình vi điều khiển ESP32 với Arduino IDE, Võ Quốc Hưng (2021), NXB Đại học Quốc gia TP.HCM.
5. Giáo trình Kỹ thuật cảm biến, Đào Văn Tường (2020), NXB Khoa học và Kỹ thuật.
6. ESP32 Projects and Tutorials, Random Nerd Tutorials (2022), Truy cập từ: <https://randomnerdtutorials.com>
7. Learning Embedded Systems, Erik T. Ray (2013), O'Reilly Media.
8. Tài liệu kỹ thuật của cảm biến MQ-6: *MQ-6 Semiconductor Sensor for Gas Leakage Detection*, Hanwei Electronics Co., Ltd. Truy cập từ: <https://www.pololu.com/file/0J309/MQ6.pdf>
9. Build Web Servers with ESP32 and ESP8266, Rui Santos, Sara Santos (2021), Random Nerd Tutorials.
10. Blynk Documentation – Quickstart, Widgets and Templates, Blynk Inc. (2021), Truy cập từ: <https://docs.blynk.io>
11. Arduino Reference Guide, Arduino.cc (2023), Truy cập từ: <https://www.arduino.cc/reference/en/>