

TRƯỜNG ĐẠI HỌC KHOA HỌC

KHOA

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

Học viên thực hiện :

Mã học viên :

Chuyên ngành : Công Nghệ Phần Mềm

Khóa học : K45

HUẾ, THÁNG 4 NĂM 2025

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Giải nghĩa
ESP32	Espressif Systems 32-bit – Vi điều khiển tích hợp Wi-Fi và Bluetooth.
MQ-6	Tên loại cảm biến khí gas thuộc dòng MQ (không viết tắt cụ thể, ký hiệu sản phẩm).
IoT	Internet of Things – Mạng lưới thiết bị kết nối Internet.
ADC	Analog to Digital Converter – Bộ chuyển đổi tín hiệu tương tự sang số.
GPIO	General Purpose Input/Output – Chân vào/ra đa dụng.
VS Code	Visual Studio Code – Trình soạn thảo mã nguồn của Microsoft.
API	Application Programming Interface – Giao diện lập trình ứng dụng.
UPS	Uninterruptible Power Supply – Bộ lưu điện.
LPG	Liquefied Petroleum Gas – Khí dầu mỏ hóa lỏng.

MỤC LỤC

PHẦN MỞ ĐẦU.....	1
I. TÍNH CẤP THIẾT CỦA ĐỀ TÀI	1
1. Thống kê về các vụ rò rỉ khí gas, bao gồm cháy nổ, thương vong và thiệt hại.	2
2. Mục Tiêu Nghiên Cứu.....	3
PHẦN NỘI DUNG	4
I. Cơ sở lý thuyết.....	4
1. Cảm biến khí gas MQ-6	4
2. Vi điều khiển ESP32	5
3. Còi báo động (Buzzer)	6
4. Mô phỏng trên Wokwi	7
5. Telegram Bot API.....	7
II. Nguyên lý tổng thể hoạt động của hệ thống.....	8
III. Cài đặt phần mềm	9
1. Môi trường phát triển và công cụ sử dụng.....	9
2. Cài đặt phần mềm	10
3. Kết nối thiết bị với ESP32	11
IV. Triển khai hệ thống.....	11
2. Cách Wokwi Mô phỏng các Thành phần Chính	11
3. Quy trình Thực hiện Mô phỏng và Kiểm thử trên Wokwi	12
V. Kết quả thực hiện.....	13
1. Hình ảnh mô phỏng trên Wokwi	13
2. Hình ảnh tin nhắn gửi qua Telegram.....	14
VI. Đánh giá và hướng phát triển	14
1. Hiệu quả của hệ thống.....	14
2. Hạn chế và hướng phát triển trong tương lai	15
PHẦN KẾT LUẬN.....	17
Tài Liệu Tham Khảo	18

PHẦN MỞ ĐẦU

Trong cuộc sống hiện đại, việc sử dụng khí gas trong sinh hoạt và công nghiệp ngày càng trở nên phổ biến nhờ tính tiện dụng và hiệu quả. Tuy nhiên, bên cạnh những lợi ích, khí gas cũng tiềm ẩn nhiều nguy cơ mất an toàn, đặc biệt là nguy cơ rò rỉ khí dẫn đến cháy nổ, gây thiệt hại nghiêm trọng về người và tài sản. Vì vậy, việc xây dựng một hệ thống phát hiện và cảnh báo rò rỉ khí gas là hết sức cần thiết nhằm đảm bảo an toàn cho người sử dụng.

Trên nền tảng của cuộc cách mạng công nghiệp 4.0, các thiết bị IoT (Internet of Things) ngày càng được ứng dụng rộng rãi trong giám sát và tự động hóa. Dự án “Hệ thống phát hiện rò rỉ khí gas sử dụng ESP32, cảm biến MQ-6, còi báo động và gửi thông báo ” là một ví dụ điển hình. Hệ thống này có khả năng giám sát nồng độ khí gas trong không khí theo thời gian thực, phát hiện sớm các dấu hiệu rò rỉ, kích hoạt còi báo động để cảnh báo tại chỗ, đồng thời gửi thông báo đến người dùng thông qua ứng dụng Telegram dù họ đang ở bất kỳ đâu.

Tiểu luận này sẽ trình bày quá trình xây dựng hệ thống, bao gồm lựa chọn linh kiện phần cứng, thiết kế mạch, lập trình cho ESP32, kết nối với Telegram, và kiểm tra hoạt động thực tế. Qua đó, đề tài không chỉ góp phần nâng cao nhận thức về an toàn khí gas mà còn mở rộng khả năng ứng dụng công nghệ IoT trong cuộc sống hàng ngày.

I. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Trong những năm gần đây, các vụ tai nạn cháy nổ do rò rỉ khí gas đã trở thành một vấn đề đáng lo ngại, đặc biệt tại các khu dân cư, nhà hàng, quán ăn, và các cơ sở sản xuất công nghiệp. Nguyên nhân chủ yếu đến từ việc rò rỉ khí gas nhưng không được phát hiện kịp thời, dẫn đến tích tụ khí trong không gian kín và bùng phát khi gặp nguồn nhiệt hoặc tia lửa điện. Hậu quả để lại không chỉ là thiệt hại về tài sản mà còn đe dọa trực tiếp đến tính mạng con người.

Trong khi đó, thực tế cho thấy nhiều hộ gia đình và cơ sở vẫn chưa được trang bị các thiết bị phát hiện rò rỉ khí chuyên dụng. Các phương pháp kiểm tra thủ công như dùng nước xà phòng hoặc cảm nhận mùi bằng giác quan không đủ tin cậy và không mang lại hiệu quả trong các tình huống khẩn cấp. Việc xây dựng một hệ thống phát hiện rò rỉ khí gas tự động, hoạt động liên tục và cảnh báo tức thời là vô cùng cần thiết trong bối cảnh hiện nay.

Không chỉ giúp phát hiện sớm rò rỉ khí gas, hệ thống còn tích hợp công nghệ IoT để gửi cảnh báo từ xa qua mạng Internet, giúp người dùng phản ứng kịp thời ngay cả khi không có mặt tại hiện trường. Việc áp dụng các thiết bị giá rẻ, dễ lập trình như ESP32 và cảm biến MQ-6 giúp hệ thống có thể triển khai rộng rãi với chi phí thấp, phù hợp với hộ gia đình và các cơ sở nhỏ.

Từ đó có thể thấy, đề tài không chỉ mang ý nghĩa thực tiễn cao mà còn phù hợp với xu hướng phát triển của công nghệ hiện đại, đặc biệt trong việc nâng cao an toàn và chất lượng cuộc sống cho cộng đồng.

1. Thống kê về các vụ rò rỉ khí gas, bao gồm cháy nổ, thương vong và thiệt hại.

Thông kê toàn cầu:

Theo ước tính của Cơ quan Năng lượng Quốc tế (IEA), lượng khí methane thải ra từ lĩnh vực năng lượng vẫn ở mức gần kỷ lục vào năm 2023, với khoảng 120 triệu tấn từ sản xuất và sử dụng nhiên liệu hóa thạch, và thêm 10 triệu tấn từ năng lượng sinh học. Các sự kiện rò rỉ methane lớn được phát hiện bởi vệ tinh đã tăng hơn 50% trong năm 2023, cho thấy vấn đề rò rỉ khí gas vẫn còn rất nghiêm trọng. Mặc dù đã có những nỗ lực giảm phát thải, tổng lượng khí thải vẫn còn quá cao so với mục tiêu khí hậu toàn cầu.[3]

Một ví dụ điển hình về sự cố rò rỉ khí gas quy mô lớn là vụ vỡ đường ống dẫn khí đốt Nord Stream ở châu Âu vào năm 2022, giải phóng tới 485.000 tấn khí methane, lượng khí thải do con người gây ra lớn nhất trên hành tinh tại thời

điểm đó . Sự cố này đã góp phần vào sự nóng lên toàn cầu tương đương với lượng khí thải từ 8 triệu ô tô chạy trong một năm .[4]

Thông kê Việt Nam:

Việt Nam có trữ lượng khí tự nhiên đáng kể, ước tính hơn 700 tỷ m³, đứng thứ ba ở Đông Nam Á . Điều này cho thấy sự phụ thuộc vào khí tự nhiên và tầm quan trọng của việc đảm bảo an toàn trong quá trình sử dụng.

Mặc dù số liệu thống kê chi tiết về các vụ rò rỉ khí gas tại Việt Nam còn hạn chế, đã có những báo cáo về các sự cố nghiêm trọng. Vào tháng 7 năm 2024, một vụ nổ do rò rỉ khí gas ở tỉnh Vĩnh Phúc đã làm bị thương 8 người, bao gồm cả người nước ngoài . Nguyên nhân ban đầu cho thấy có thể đã có cảnh báo về rò rỉ khí gas trước đó nhưng không được chú ý . Vào tháng 4 năm 2024, một vụ nổ khí methane tại một mỏ than ở tỉnh Quảng Ninh đã khiến 4 thợ mỏ thiệt mạng và 7 người khác bị thương . Sự cố này cho thấy nguy cơ rò rỉ và nổ khí methane trong môi trường công nghiệp khai thác mỏ ở Việt Nam. Trước đó, cũng đã có những vụ nổ khí methane gây chết người tại các mỏ than khác ở Quảng Ninh .[1,2]

Việt Nam cũng nằm trong số các quốc gia có lượng phát thải khí nhà kính bình quân đầu người tăng nhanh trên thế giới , cho thấy tầm quan trọng của việc kiểm soát tất cả các nguồn phát thải khí, bao gồm cả rò rỉ khí gas.

2. Mục Tiêu Nghiên Cứu

- Xây dựng hệ thống phát hiện khí gas rò rỉ dùng ESP32 và cảm biến MQ-6.
- Gửi cảnh báo real-time qua Telegram và phát còi báo động.
- Đánh giá hiệu quả hệ thống và đề xuất hướng phát triển.

PHẦN NỘI DUNG

I. Cơ sở lý thuyết

1. Cảm biến khí gas MQ-6

Khái niệm: Cảm biến MQ-6 là một cảm biến bán dẫn chuyên dụng để phát hiện khí gas dễ cháy như LPG (liquefied petroleum gas), butan, propan và metan. MQ-6 thuộc dòng cảm biến khí MQ-series, nổi bật bởi độ nhạy cao với LPG và tính ứng dụng rộng rãi trong các thiết bị an toàn công nghiệp và gia đình.

Nguyên lý hoạt động:

- MQ-6 sử dụng phần tử cảm biến làm từ thiếc oxit (SnO_2). Bình thường, trong không khí sạch, điện trở bề mặt cao.
- Khi khí dễ cháy xuất hiện, khí sẽ tương tác với bề mặt thiếc oxit, làm giảm điện trở.
- Sự thay đổi điện trở này được chuyển thành tín hiệu điện áp ra dưới dạng analog, gửi tới bộ vi điều khiển để xử lý.
- MQ-6 mất khoảng 24-48 giờ để "nung nóng lần đầu" (preheat) để hoạt động ổn định, nhưng trong mô phỏng Wokwi quá trình này được mô phỏng ngay lập tức.

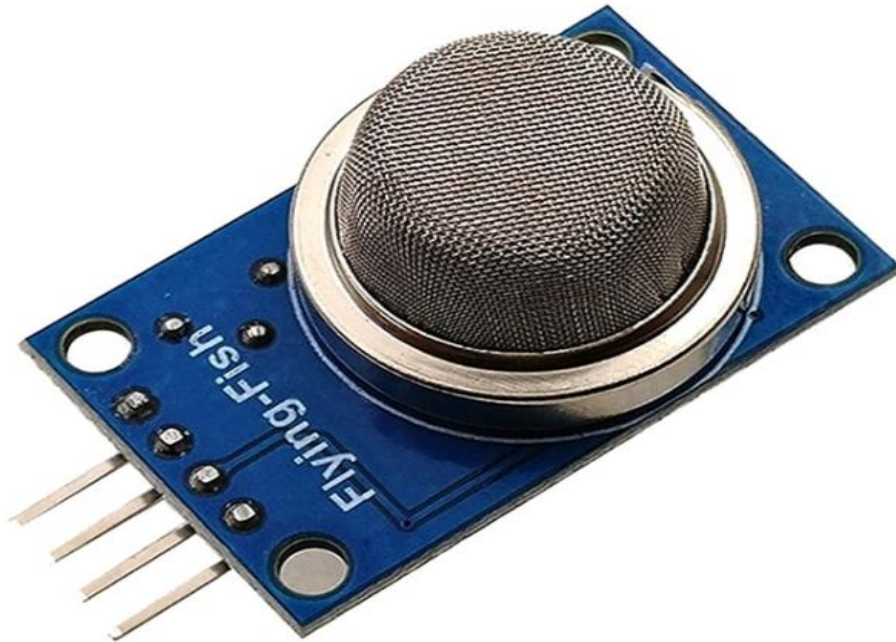
Ưu điểm:

- Giá thành thấp, dễ mua.
- Hoạt động ổn định trong môi trường thông thường.
- Dễ tích hợp với vi điều khiển như ESP32.

Nhược điểm:

- Không phân biệt loại khí (chỉ biết nồng độ tổng hợp).
- Bị ảnh hưởng bởi nhiệt độ và độ ẩm môi trường.

- Cần hiệu chỉnh định kỳ để đảm bảo độ chính xác.[5]



Hình ảnh cảm biến khí Gas MQ-6

2. Vi điều khiển ESP32

Khái niệm: ESP32 là vi điều khiển mạnh mẽ tích hợp Wi-Fi và Bluetooth, thuộc dòng sản phẩm ESPressif. Chip này rất phổ biến trong các dự án IoT nhờ hiệu suất xử lý cao, giá thành thấp và khả năng mở rộng tốt.

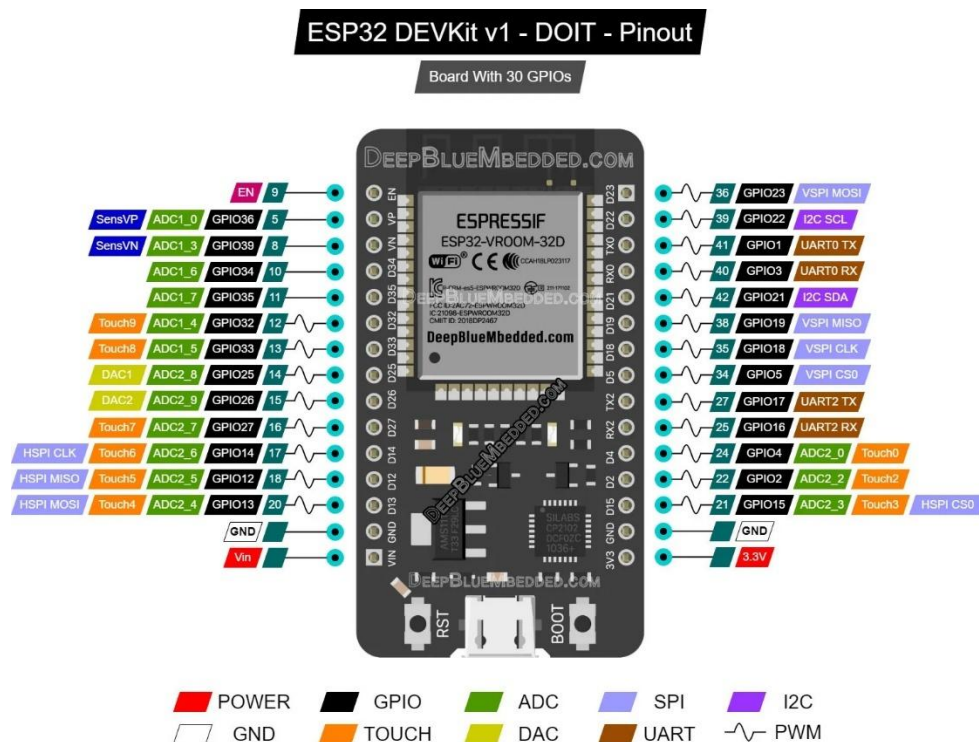
Chức năng trong hệ thống:

- Đọc tín hiệu analog từ MQ-6 qua ADC (Analog to Digital Converter).
- Xử lý dữ liệu: So sánh giá trị đọc với mức ngưỡng an toàn đã đặt trước.
- Điều khiển buzzer: Kích hoạt cảnh báo khi phát hiện khí gas vượt mức cho phép.
- Giao tiếp Serial: Gửi thông tin đo lường lên cửa sổ Serial Monitor để người dùng dễ dàng theo dõi.

Ưu điểm khi dùng ESP32

- Xử lý tín hiệu nhanh, đáp ứng thời gian thực.

- Tiêu thụ điện năng thấp.
- Hỗ trợ nhiều chuẩn giao tiếp mở rộng cho các thiết bị ngoại vi khác.[6]



Hình ảnh ESP32 DevKit V1

3. Còi báo động (Buzzer)

Khái niệm: Buzzer là thiết bị phát âm thanh cảnh báo dạng điện tử, giúp người sử dụng nhận biết tình trạng khẩn cấp qua tín hiệu âm thanh lớn và rõ ràng.

Nguyên lý hoạt động trong hệ thống:

- Khi ESP32 phát hiện tín hiệu analog từ MQ-6 vượt ngưỡng cài đặt, vi điều khiển sẽ xuất tín hiệu digital HIGH đến chân điều khiển buzzer.
- Buzzer nhận điện và phát ra âm thanh cảnh báo người dùng về nguy cơ cháy nổ do khí gas rò rỉ.
- Khi nồng độ khí trở về mức an toàn, tín hiệu điều khiển từ ESP32 sẽ là LOW, tắt buzzer.

4. Mô phỏng trên Wokwi

Giới thiệu: Wokwi là nền tảng mô phỏng phần cứng trực tuyến mạnh mẽ, hỗ trợ nhiều loại vi điều khiển và cảm biến như ESP32, ESP8266, Arduino UNO, cảm biến khí MQ-series, màn hình OLED, LCD, v.v.

Lợi ích khi mô phỏng:

- Không cần phần cứng thật, tiết kiệm chi phí và thời gian.
- Dễ dàng điều chỉnh tham số, giúp kiểm tra phản ứng của hệ thống trong nhiều tình huống khí gas khác nhau.
- Theo dõi dữ liệu trực tiếp qua Serial Monitor.
- Giao diện trực quan, dễ dàng kéo thả, nối dây và lập trình.

Cách mô phỏng cảm biến MQ-6 trên Wokwi:

- MQ-6 trên Wokwi có thanh điều chỉnh "Gas Level" cho phép mô phỏng thay đổi nồng độ khí gas.
- Khi tăng "Gas Level", giá trị tín hiệu analog từ cảm biến cũng tăng theo.
- Điều này giúp mô phỏng được tình huống rò rỉ khí gas và kiểm tra khả năng phản ứng của hệ thống cảnh báo.[7]

5. Telegram Bot API

Giới Thiệu: Telegram là một nền tảng nhắn tin đa nền tảng miễn phí với tốc độ cao, mã hóa mạnh mẽ và hỗ trợ đa dạng các hệ điều hành như Android, iOS, Windows, Linux và web. Một trong những tính năng nổi bật của Telegram là hỗ trợ **Bot** – các ứng dụng tự động hóa được điều khiển thông qua **Telegram Bot API**. Các bot này có thể gửi, nhận tin nhắn, phản hồi người dùng, truy xuất dữ liệu từ web hoặc giao tiếp với các thiết bị IoT.

Telegram Bot API là một giao diện lập trình ứng dụng (API) do Telegram cung cấp, cho phép các lập trình viên và hệ thống IoT dễ dàng tích hợp chức năng gửi và nhận tin nhắn tự động. Trong bối cảnh của đề tài này, Telegram Bot đóng

vai trò là kênh cảnh báo từ xa, giúp người dùng nhận được thông báo ngay lập tức khi phát hiện khí gas rò rỉ từ hệ thống cảm biến.

Lợi ích khi mô phỏng:

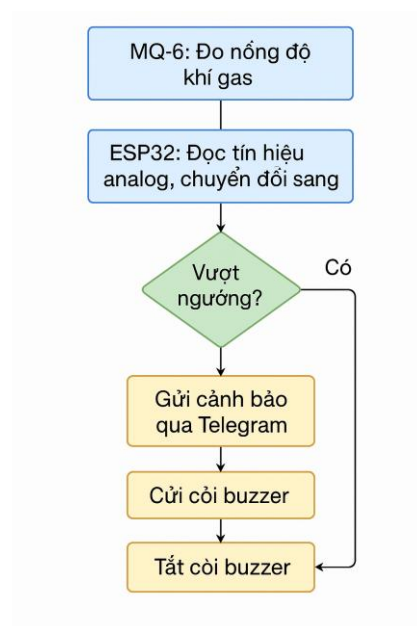
Việc tích hợp Telegram Bot API trong hệ thống phát hiện rò rỉ khí gas mang lại nhiều lợi ích vượt trội so với các phương pháp cảnh báo truyền thống như còi báo:

- Cảnh báo thời gian thực từ xa: Cho phép người dùng nhận được thông báo bất kể họ đang ở đâu, miễn là có kết nối Internet.
- Miễn phí và dễ sử dụng: Không cần xây dựng ứng dụng riêng hoặc thuê máy chủ. Telegram cung cấp nền tảng hoàn chỉnh, ổn định và miễn phí.
- Khả năng mở rộng cao: Telegram Bot có thể dễ dàng tích hợp thêm các tính năng như nhật ký cảnh báo, điều khiển thiết bị từ xa, xác thực người dùng, tích hợp AI...
- Tính bảo mật cao: Các tin nhắn gửi qua Telegram được mã hóa, đảm bảo tính riêng tư và an toàn dữ liệu cho người sử dụng.
- Mô phỏng hiệu quả: Khi mô phỏng hệ thống trên nền tảng như Wokwi, việc tích hợp Telegram giúp kiểm thử chức năng gửi cảnh báo thực tế mà không cần phần cứng thật. Người dùng có thể kiểm tra tin nhắn gửi đến Telegram thật để xác minh hoạt động của hệ thống.

II. Nguyên lý tổng thể hoạt động của hệ thống

- Cảm biến MQ-6 liên tục đo nồng độ khí gas trong không khí và xuất tín hiệu analog.
- ESP32 đọc tín hiệu từ cảm biến qua chân ADC và chuyển đổi thành giá trị số.
- ESP32 so sánh giá trị này với ngưỡng an toàn đã định.
- Nếu vượt ngưỡng:

- Cảnh báo tại chỗ: ESP32 ngay lập tức gửi tín hiệu đến chân GPIO đang điều khiển còi buzzer.
- Thông báo từ xa: Sau khi kích hoạt còi, ESP32 sử dụng kết nối Wi-Fi đã thiết lập và thư viện để gửi một tin nhắn cảnh báo (ví dụ: "PHÁT HIỆN RÒ RỈ KHÍ GAS!") đến Chat ID của người dùng hoặc nhóm đã được cấu hình.
- Nếu dưới ngưỡng:
 - Còi buzzer được giữ ở trạng thái tắt.
 - Không có tin nhắn Telegram nào được gửi đi.
 - Hệ thống quay lại bước đọc cảm biến tiếp theo trong vòng lặp.
- Người dùng có thể thay đổi giá trị "Gas Level" trên Wokwi để kiểm tra phản ứng hệ thống.



Sơ đồ hệ thống

III. Cài đặt phần mềm

1. Môi trường phát triển và công cụ sử dụng

Để phát triển phần mềm cho hệ thống, nhóm sử dụng các công cụ sau:

- Visual Studio Code (VS Code): là trình soạn thảo mã nguồn mạnh mẽ, hỗ trợ đa nền tảng và dễ sử dụng.
- PlatformIO: là một extension của VS Code, hỗ trợ lập trình các vi điều khiển như ESP32.
- Wokwi: nền tảng mô phỏng trực tuyến cho phép mô phỏng các dự án phân cứng như ESP32, cảm biến MQ-6, buzzer, v.v.
- Telegram Bot API: dùng để gửi tin nhắn cảnh báo khi phát hiện rò rỉ khí gas.

2. Cài đặt phần mềm

- Cài đặt VS Code: Tải và cài đặt từ <https://code.visualstudio.com>
- Cài đặt PlatformIO: Mở VS Code, vào tab Extensions (Ctrl + Shift + X), tìm "PlatformIO IDE" và cài đặt.
- Cài đặt Wokwi Extension: Tìm và cài "Wokwi for VS Code"
- Tạo dự án PlatformIO mới: Vào PlatformIO Home > New Project

Tên: ESP32_RoRiGas => Board: nodemcu-32s => Framework: Arduino

- Cài thư viện cần thiết trong platformio.ini:

```
[env:nodemcu-32]
```

```
platform = espressif32
```

```
board = nodemcu-32s
```

```
framework = arduino
```

```
monitor_speed = 115200
```

```
lib_deps =
```

```
bblanchon/ArduinoJson@^7.3.1[9]
```

```
witnessmenow/UniversalTelegramBot@^1.3.0[10]
```

3. Kết nối thiết bị với ESP32

Sử dụng sơ đồ sau để kết nối:

- MQ-6 AO → GPIO34 (ESP32)
- MQ-6 VCC → 5V (ESP32)
- MQ-6 GND → GND
- Buzzer chân + → GPIO18
- Buzzer chân - → GND

IV. Triển khai hệ thống

1. Thiết lập môi trường mô phỏng trên Wokwi

- Chọn Linh kiện: Bạn sẽ vào thư viện linh kiện của Wokwi, tìm và kéo thả board ESP32 (ví dụ: ESP32 DevKit V1), cảm biến "Gas Sensor (MQ-6)", và "Buzzer" vào khu vực làm việc.
- Kết nối Mạch ảo: Sử dụng chuột để vẽ các đường nối dây ảo giữa các chân linh kiện, tuân thủ đúng sơ đồ kết nối bạn đã thiết kế trong Phần III (ví dụ: chân AO của MQ-6 nối với GPIO34 của ESP32, chân VCC của MQ-6 nối với chân 5V của ESP32, chân GND nối với GND, chân dương (+) của Buzzer nối với GPIO18, chân âm (-) nối với GND).
- Tích hợp Code: Bạn sẽ dán (copy-paste) toàn bộ mã nguồn (code) mà bạn đã viết cho ESP32 (thường là từ VS Code + PlatformIO hoặc Arduino IDE) vào cửa sổ soạn thảo code (sketch.ino) của Wokwi.

2. Cách Wokwi Mô phỏng các Thành phần Chính

ESP32: Wokwi chạy một trình giả lập thực thi mã C++/Arduino của bạn. Nó mô phỏng hoạt động của CPU Tensilica LX6, bộ nhớ, các chân GPIO (Input/Output), bộ chuyển đổi ADC, và quan trọng nhất là mô-đun Wi-Fi ảo.

Cảm biến MQ-6: Đây là điểm mấu chốt của mô phỏng:

- Wokwi không mô phỏng phản ứng hóa học thực tế, thay vào đó, nó cung cấp một thanh trượt (slider) tương tác có nhãn "Gas Level" trên biểu tượng cảm biến MQ-6.
- Khi bạn kéo thanh trượt này, Wokwi sẽ thay đổi giá trị điện áp ảo tại chân đầu ra Analog (AO) của cảm biến MQ-6 mô phỏng.
- Code trên ESP32 ảo sẽ đọc giá trị điện áp này thông qua chân ADC (GPIO34), giống như cách nó đọc từ cảm biến thật. Giá trị ADC đọc được sẽ tăng/giảm tương ứng với vị trí thanh trượt.
- Điều này cho phép bạn giả lập các kịch bản khác nhau: không khí sạch (slider ở mức thấp), có rò rỉ nhẹ (slider ở mức giữa), rò rỉ nghiêm trọng (slider ở mức cao). Bạn có thể dễ dàng kiểm tra xem hệ thống có phản ứng đúng tại ngưỡng bạn cài đặt trong code hay không.

Còi Buzzer: Khi code trên ESP32 ảo thực hiện lệnh `digitalWrite(BUZZER_PIN, HIGH)` (với `BUZZER_PIN` là `GPIO18`), Wokwi sẽ thay đổi hình ảnh của còi buzzer ảo (ví dụ: có thêm các vòng sóng âm thanh) để cho biết nó đang "kêu". Khi lệnh là `digitalWrite(BUZZER_PIN, LOW)`, hình ảnh còi sẽ trở lại bình thường.

Kết nối Internet và Telegram: Khi code của bạn gọi các hàm để kết nối Wi-Fi (`WiFi.begin(...)`), Wokwi sẽ sử dụng kết nối Internet của máy tính bạn (thông qua trình duyệt) để cấp cho ESP32 ảo một kết nối mạng. Khi code sử dụng thư viện Telegram Bot để gửi tin nhắn (`bot.sendMessage(...)`), Wokwi sẽ cho phép yêu cầu đó đi qua Internet ảo đến máy chủ thật của Telegram. Nếu Bot Token và Chat ID trong code của bạn là đúng, bạn sẽ nhận được tin nhắn cảnh báo thật trên ứng dụng Telegram của mình.

3. Quy trình Thực hiện Mô phỏng và Kiểm thử trên Wokwi

Bước 1: Mở dự án Wokwi đã thiết lập mạch và code.

Bước 2: Nhấn nút "Start the simulation" (biểu tượng Play màu xanh).

Bước 3: Quan sát cửa sổ "Serial Monitor" ảo: Kiểm tra các thông báo log bạn đã viết trong code (ví dụ: "Connecting to WiFi...", "WiFi connected", "Gas Value: xxx", "Gas Leak Detected!").

Bước 4: Tương tác với mô phỏng: Dùng chuột kéo thanh trượt "Gas Level" trên cảm biến MQ-6 lên/xuống qua mức ngưỡng bạn đã định nghĩa trong code.

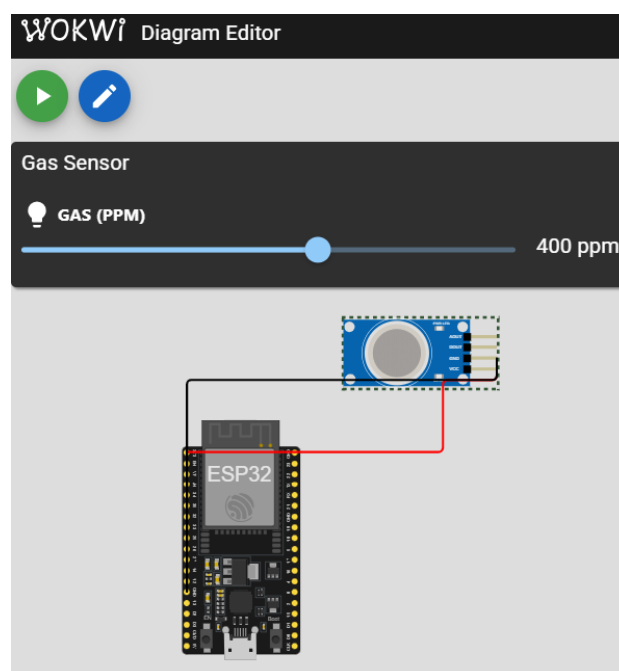
Bước 5: Quan sát kết quả:

- Còi Buzzer ảo có thay đổi trạng thái (kêu/im lặng) tương ứng không?
- Kiểm tra điện thoại/máy tính của bạn xem có nhận được tin nhắn từ Telegram Bot không? Nội dung tin nhắn có đúng không?

Bước 6: Lặp lại Bước 4 và 5 với các mức "Gas Level" khác nhau để đảm bảo hệ thống hoạt động ổn định và chính xác theo logic đã lập trình.

V. Kết quả thực hiện

1. Hình ảnh mô phỏng trên Wokwi



2. Hình ảnh tin nhắn gửi qua Telegram



VI. Đánh giá và hướng phát triển

1. Hiệu quả của hệ thống

Khả năng phát hiện sớm và kịp thời:

- Hệ thống sử dụng cảm biến MQ-6 có độ nhạy cao với khí gas dễ cháy (như LPG, butan, methane). Khi nồng độ khí vượt ngưỡng an toàn, cảm biến gửi tín hiệu ngay lập tức tới vi điều khiển ESP32.
- Còi buzzer phát tín hiệu cảnh báo trực tiếp tại chỗ giúp người gần khu vực rò rỉ có thể xử lý ngay lập tức.

Cảnh báo từ xa qua Telegram Bot:

- Với sự tích hợp Telegram Bot, người dùng nhận được cảnh báo tức thì qua điện thoại dù đang ở bất kỳ đâu, miễn là có kết nối internet.

- Không cần xây dựng server hoặc giao diện web phức tạp, tiết kiệm chi phí và thời gian phát triển.

Telegram có độ bảo mật tốt và hỗ trợ đa nền tảng (Android, iOS, Desktop).

- Mô phỏng dễ dàng và kiểm thử linh hoạt:
 - Thông qua Wokwi, hệ thống có thể được mô phỏng đầy đủ trước khi triển khai thực tế.
 - Kết hợp với Visual Studio Code và PlatformIO giúp lập trình dễ dàng, kiểm tra và sửa lỗi nhanh chóng.
- Chi phí thấp, tính ứng dụng cao:
 - ESP32 và MQ-6 là các thiết bị giá thành rẻ nhưng hiệu quả, giúp giảm chi phí triển khai.
 - Hệ thống phù hợp sử dụng trong gia đình, nhà kho, trạm gas, hoặc các khu vực dễ cháy nổ.

2. Hạn chế và hướng phát triển trong tương lai

Hạn chế:

Độ chính xác cảm biến MQ-6:

- MQ-6 cần thời gian làm nóng ban đầu và có thể bị ảnh hưởng bởi nhiệt độ môi trường hoặc độ ẩm, dẫn tới sai số nhẹ trong một số điều kiện.
- Phụ thuộc vào mạng Internet:
- Hệ thống sử dụng Telegram Bot nên cần kết nối internet ổn định. Nếu mất mạng, cảnh báo từ xa sẽ không hoạt động.
- Chỉ cảnh báo đơn giản:
- Hệ thống hiện tại mới chỉ gửi tin nhắn văn bản. Chưa có báo cáo lịch sử dữ liệu, thống kê nồng độ khí, hay điều khiển nâng cao.

Hướng phát triển trong tương lai:

- Phát triển lưu trữ dữ liệu và phân tích xu hướng:
 - Tích hợp thêm module lưu trữ dữ liệu vào thẻ nhớ hoặc gửi dữ liệu lên cloud để phân tích xu hướng rõ ràng, lập báo cáo định kỳ.
- Nâng cấp giao diện hiển thị:
 - Ngoài Telegram, có thể phát triển thêm giao diện web hoặc ứng dụng di động để giám sát dữ liệu thời gian thực.
- Mở rộng loại cảm biến:
 - Bổ sung thêm các cảm biến khác như cảm biến khói, nhiệt độ, hoặc khí CO để tăng cường an toàn cho toàn khu vực giám sát.
- Tự động hoá phản ứng:
 - Khi phát hiện rò rỉ, hệ thống có thể điều khiển tự động tắt van gas hoặc kích hoạt hệ thống thông gió để giảm nguy cơ cháy nổ.
- Cải tiến về nguồn điện:
 - Tích hợp bộ lưu điện (UPS) hoặc pin dự phòng giúp hệ thống hoạt động ổn định ngay cả khi mất điện.

PHẦN KẾT LUẬN

Đề tài "Hệ thống phát hiện rò rỉ khí gas sử dụng ESP32, cảm biến MQ-6, còi báo động và gửi thông báo" đã được nghiên cứu và triển khai một cách toàn diện, từ lý thuyết đến thực hành mô phỏng. Hệ thống đã chứng minh được tính hiệu quả trong việc phát hiện sớm các dấu hiệu rò rỉ khí gas, cảnh báo kịp thời cả tại chỗ lẫn từ xa thông qua nền tảng Telegram.

Qua quá trình thực hiện, tiểu luận đã khẳng định được tiềm năng ứng dụng của các công nghệ hiện đại như vi điều khiển ESP32 và nền tảng IoT vào các vấn đề thực tiễn trong cuộc sống, đặc biệt là trong lĩnh vực an toàn cháy nổ. Không chỉ giúp giảm thiểu rủi ro về con người và tài sản, hệ thống còn có khả năng mở rộng để đáp ứng các nhu cầu giám sát phức tạp hơn trong tương lai.

Tuy vẫn còn tồn tại một số hạn chế như phụ thuộc vào kết nối Internet hoặc độ chính xác cảm biến, đề tài đã mở ra nhiều hướng phát triển tích cực, như tích hợp thêm chức năng điều khiển tự động, lưu trữ và phân tích dữ liệu, hoặc kết nối với các hệ thống cảnh báo khác.

Tiểu luận không chỉ là một bài học về kỹ thuật lập trình và mô phỏng phần cứng, mà còn thể hiện tư duy sáng tạo và ứng dụng thực tiễn trong giải quyết các vấn đề an toàn. Đây là tiền đề quan trọng để phát triển các dự án IoT có ý nghĩa xã hội cao trong thời gian tới.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] VnExpress. (2024, tháng 7). Vụ nổ khí gas tại Vĩnh Phúc làm 8 người bị thương. Truy cập từ <https://vnexpress.net>

[2] Tuổi Trẻ Online. (2024, tháng 4). Vụ nổ khí methane tại mỏ than ở Quảng Ninh khiến 4 người tử vong. Truy cập từ <https://tuoitre.vn>

Tiếng Anh

[3] Cơ quan Năng lượng Quốc tế (IEA). (2023). Global Methane Tracker 2023. Truy cập từ <https://www.iea.org/reports/global-methane-tracker-2023>

[4] Vụ nổ đường ống giải phóng lượng khí mê-tan kỷ lục: Nghiên cứu của UNEP <https://www.unep.org/news-and-stories/story/pipeline-blasts-released-record-shattering-amount-methane-unep-study>

[5] https://www.pololu.com/file/download/mq6.pdf?file_id=0j312

[6] <https://vi.wikipedia.org/wiki/ESP32>

[7] Wokwi. (n.d.). Wokwi IoT Simulator Documentation. Truy cập từ <https://docs.wokwi.com>

[8] Telegram. (n.d.). Telegram Bot API. Truy cập từ <https://core.telegram.org/bots/api>

[9] bblanchon. (n.d.). ArduinoJson Library. Truy cập từ <https://arduinojson.org>

[10] WitnessMeNow. (n.d.). UniversalTelegramBot Library for Arduino. Truy cập từ <https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot>