

**BÁO CÁO BÀI TẬP LỚN**

**MÔN: KỸ THUẬT VI XỬ LÝ**

**ĐỀ TÀI:**

**MÔ HÌNH HỆ THỐNG GIÁM SÁT CHẤT LƯỢNG KHÔNG KHÍ DỰA TRÊN CÔNG NGHỆ IOT**

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

TRƯỜNG ĐIỆN - ĐIỆN TỬ

-----□□&□□-----



***Hà Nội, tháng 7 năm 2023***

Giảng viên hướng dẫn: **Ts. Hàn Huy Dũng**

Mã lớp: **142050**

Nhóm sinh viên thực hiện: **Nhóm 14**

Lê Văn Quyết 20200519

Lê Công Thành 20203584

Vũ Hồng Diệu 20203360

Phùng Quang Bắc 20203321

MỤC LỤC

[MỞ ĐẦU 2](#_Toc142478505)

[CHƯƠNG 1: TỔNG QUAN CHỈ SỐ CHẤT LƯỢNG KHÔNG KHÍ (AQI) 3](#_Toc142478506)

[1.1. Khảo sát hiện trạng, mục đích dự án 3](#_Toc142478507)

[1.2. Khái niệm 4](#_Toc142478508)

[1.3. Tính toán chỉ số chất lượng không khí Việt Nam. 4](#_Toc142478509)

[1.4. Tính toán giá trị VN\_AQI 5](#_Toc142478510)

[CHƯƠNG 2: XÂY DỰNG MÔ HÌNH 9](#_Toc142478511)

[CHƯƠNG 3: CÁC CẢM BIẾN SỬ DỤNG ĐO CHẤT LƯỢNG KHÔNG KHÍ 10](#_Toc142478512)

[3.1. Vi điều khiển ESP32 10](#_Toc142478513)

[3.2 MQ-7 Mạch Cảm Biến Khí (CO) 16](#_Toc142478514)

[3.3. Cảm Biến Chất Lượng Không Khí MQ-135 19](#_Toc142478515)

[CHƯƠNG 4: PHẦN CODE 21](#_Toc142478516)

[4.1 Lập trình esp32 21](#_Toc142478517)

[4.2. Kết quả kiểm thử 29](#_Toc142478518)

[TÀI LIỆU THAM KHẢO 31](#_Toc142478519)

# MỞ ĐẦU

Hiện nay, cùng với sự phát triển mạnh mẽ của khoa học công nghệ, hoạt động sản xuất công nghiệp, xe cộ, … ô nhiễm không khí ngày càng trở nên nghiêm trọng. Theo Tổ chức Y tế Thế giới (WHO), ô nhiễm không khí gây ra 7 triệu ca tử vong sớm mỗi năm[5]. Số liệu thống kê của Liên hợp quốc cũng cho thấy, mỗi giờ có khoảng 800 ca tử vong do ô nhiễm không khí, trung bình 13 người tử vong mỗi phút, gấp ba lần số tử vong vì sốt rét, bệnh lao và AIDS mỗi năm. Cũng theo WHO, Việt Nam có 34.232 người tử vong sớm có liên quan đến ô nhiễm không khí[5]. Ở nước ta hiện nay, tại những thành phố lớn như Hà Nội, thành phố Hồ Chí Minh, và những thành phố công nghiệp mức độ ô nhiễm những năm gần đây đã nhiều lúc lên ngưỡng báo động. Cùng với đó nhiều thành phố ở Việt Nam từng bước phát triển để trở thành các đô thị thông minh. Trong các tiêu chí phát triển thành phố xanh và hiện đại, xây dựng một môi trường sống trong lành và an toàn là việc làm rất cần thiết nhằm đảm bảo sức khỏe, nâng cao chất lượng cuộc sống con người[4]. Từ đó, việc đo lường và giám sát môi trường không khí ngày càng trở nên cấp thiết.

Tuy nhiên việc theo dõi, giám sát chất lượng không khí chưa thực sự đáp ứng với tình hình hiện nay, do giới hạn về chi phí, cùng với thiếu hụt các hệ thống quan trắc và sự xuống cấp của hệ thống theo thời gian.

Chi phí là 1 trong những rào cản rất lớn trong việc chúng ta mở rộng giám sát chất lượng không khí, với sự xuất hiện của các cảm biến với chi phí thấp kết hợp cùng với “Internet of Thing” (IoT), là giải pháp xem kẽ những hệ thống lớn để tăng độ tin cậy, và độ chính xác của hệ thống giám sát. Chính vì những vấn đề vậy, trong bài cáo báo này chúng em nghiêm cứu về Mô hình hệ thống giám sát chất lượng không khí (PM2.5, CO, ...) dựa trên trên công nghệ IoT.

Với mục tiêu thiết kế, chế tạo thiết bị đo nồng độ ô nhiễm trong không khí có giá thành hợp lý cùng khả năng ứng dụng thực tiễn, báo cáo chúng em sử dụng kết hợp các cảm biến với vi điều khiển trên nền IoT, chúng em hy vọng có thể đóng góp vào việc giám sát chất lượng không khí và tạo ra thông tin hữu ích để người dùng có thể thực hiện các biện pháp cần thiết để bảo vệ sức khỏe và môi trường xung quanh.

Chúng em xin chân thành cảm ơn!

# CHƯƠNG 1: TỔNG QUAN CHỈ SỐ CHẤT LƯỢNG KHÔNG KHÍ (AQI)

## Khảo sát hiện trạng, mục đích dự án

* Khảo sát hiện trạng:

Ước tính đầu năm 2018 cho thấy rằng 9/10 người dân phải hít thở không khí với hàm lượng ô nhiễm cực kì cao. Ô nhiễm không khí cả ở bên ngoài và trong nhà gây ra khoảng 7 triệu ca tử vong mỗi năm trên toàn cầu và chỉ tính riêng trên khu vực Tây Thái Bình Dương thì đã có khoản 2.2 triệu ca tử vong. Ở Việt Nam số ca tử vong trung bình là 60000 người mỗi năm liên quan đến vấn đề ô nhiễm không khí.

Ô nhiễm môi trường đang trở thành vấn đề nghiêm trọng của Hà Nội. Nồng độ bụi mịn PM2.5 trung bình ở Hà Nội là 46.9µg/m3 – nghiêm trọng hơn so với năm 2018 (40.8µg/m3) cao gấp 5 lần khuyến cáo của WHO và gấp 2 lần quy định tiêu chuẩn của Việt Nam. Một số nguyên nhân dẫn đến tình trạng này có thể kể đến như quá trình đô thị hóa ở Việt Nam, sự phát triển của các khu công nghiệp với số lượng nhà máy tăng nhanh. Thêm vào đó, sự phát triển cơ sở hạ tầng kéo theo việc gia tăng số lượng phương tiện tham gia giao thông hàng ngày khiến một lượng lớn khí thải được xả ra môi trường. Chất lượng không khí xấu ảnh hưởng nghiêm trọng đến sức khỏe của chính những người dân trong khu vực, gây ra những bệnh nguy hiểm về đường hô hấp do khí hít vào tích tụ trong phổi như hen suyễn, dị ứng, nghiêm trọng hơn là ung thư phổi. Nguy hiểm hơn nữa, theo Cơ quan bảo vệ môi trường Mỹ (EPA) khi mức độ ô nhiễm không khí trong nhà có thể cao hơn 100 lần so với ngoài trời.

Tuy nhiên, chất lượng không khí thể biết được chỉ qua cảm quan của con người. Điều này càng làm người dân khó có thể nhận biết được mức độ nguy hại của không khí để từ đó có những biện pháp thích hợp.

Để giải quyết vấn đề này, việc có thể theo dõi thông tin về các chỉ số trong không khí kịp thời và chính xác là rất cần thiết. Bên cạnh đó, dữ liệu sau khi được thu sẽ được đẩy lên server để theo dõi. Ứng dụng IoT trong việc giám sát, cảnh báo mức độ ô nhiễm không khí sẽ giúp cho người dân sẽ nắm được thông tin về tình trạng môi trường không khí trong khu vực sinh sống và làm việc một cách nhanh chóng và chính xác nhất từ đó giúp họ đưa ra những quyết định phù hợp. Bên cạnh đó, khi các chỉ số môi trường không khí vượt ngưỡng cho phép, gây hại cho sức khỏe con người thì hệ thống IoT sẽ ngay lập tức phân tích xử lý và đưa ra các cảnh báo để người dân kịp thời ứng phó.

* Mục đích dự án:

Nhóm mong muốn tập trung xây dựng một sản phẩm có khả năng cung cấp cho người dùng thông tin về chỉ số chất lượng chất lượng không khí một cách đầy đủ, chi tiết và chính xác. Các thông số hiển thị sẽ bao gồm nhiệt độ, độ ẩm, nồng độ khí NH3, khói, CO2 và các loại bụi như PM1.0, PM2.5, PM10. Các thông số được hiển thị thông qua giao diện trực quan giúp người dùng dễ đưa ra những biện pháp cần thiết. Ngoài ra, dữ liệu thu thập được từ cảm biến sẽ được chuyển đến server nhằm mục đích theo dõi tình hình không khí.

## Khái niệm

Chỉ số chất lượng không khí (viết tắt AQI – Air Quality Index) là một chỉ số báo cáo chất lượng không khí hàng ngày[3]. Đây được coi là một thước đo đơn giản hóa mức độ ô nhiễm không khí, cho biết không khí xung quanh ta là sạch hay ô nhiễm, ô nhiễm đến mức độ nào. Rủi ro đối với sức khỏe cộng đồng càng cao khi chỉ số AQI càng lớn. Chỉ số AQI tập trung vào sự ảnh hưởng tới sức khỏe người dân có thể gặp trong vòng vài giờ hoặc vài ngày sau khi hít thở không khí ô nhiễm.

Chỉ số chất lượng không khí được tính theo thang điểm (khoảng giá trị AQI) tương ứng với biểu tượng và các màu sắc để cảnh báo chất lượng không khí và mức độ ảnh hưởng tới sức khỏe con người, cụ thể như sau:

*Bảng 1: Khoảng giá trị AQI và đánh giá chất lượng không khí*

|  |  |  |  |
| --- | --- | --- | --- |
| **Khoảng giá trị AQI** | **Chất lượng không khí** | **Màu sắc** | **Mã màu RBG** |
| 0- 50 | Tốt | Xanh | 0;228;0 |
| 51- 100 | Trung bình | Vàng | 255;255;0 |
| 101 - 150 | Kém | Da cam | 255;126;0 |
| 151 - 200 | Xấu | Đỏ | 255;0;0 |
| 201 - 300 | Rất xấu | Tím | 143;63;151 |
| 301-500 | Nguy hại | Nâu | 126;0;35 |

## Tính toán chỉ số chất lượng không khí Việt Nam.

Chỉ số chất lượng không khí Việt Nam (viết tắt VN\_AQI)

* VN\_AQI được tính toán bao gồm AQI giờ và AQI ngày. Số liệu sử dụng để tính toán VN\_AQI là giá trị quan trắc trung bình 1 giờ, trung bình 8 giờ và trung bình 24 giờ.
* VN\_AQI được tính toán cho dữ liệu của từng trạm quan trắc không khí tự động liên tục đối với môi trường không khí xung quanh.
* Đối với mỗi trạm quan trắc, AQIx được tính toán cho từng thông số quan trắc, giá trị AQI cuối cùng là giá trị lớn nhất trong các giá trị AQIx của mỗi thông số.
* Các thông số được sử dụng để tính VN\_AQI bao gồm: SO2, CO, NO2, O3, PM10, và PM2.5.
* Phương pháp tính toán VN\_AQI yêu cầu bắt buộc phải có tối thiểu 01 trong 02 thông số PM10, PM2.5 trong công thức tính.

## Tính toán giá trị VN\_AQI

**Tính toán giá trị AQI giờ (AQIh)**

Số liệu tính toán AQI giờ là giá trị quan trắc trung bình 1 giờ.

1. ***Tính giá trị Nowcast đối với thông số PM2.5 và PM10***

Gọi c1, c2, …c12 là 12 giá trị quan trắc trung bình 1 giờ (với c1 là giá trị quan trắc trung bình 1 giờ hiện tại, c12 là giá trị quan trắc trung bình 1 giờ cách 12 giờ so với hiện tại).

Tính giá trị trọng số:

Trong đó là giá trị nhỏ nhất trong số 12 giá trị trung bình 1 giờ

là giá trị lớn nhất trong số 12 giá trị trung bình 1 giờ.

Nếu thì lấy

Nếu thì lấy

Trong trường hợp thì giá trị

Trong trường hợp thì

Lưu ý: Cần có ít nhất 2 trong 3 giá trị c1, c2, c3 có dữ liệu thì mới tính được giá trị Nowcast, ngược lại nếu ít hơn không tính được giá trị

1. ***Tính giá trị AQIh của từng thông số (AQIx)***

Giá trị AQIh của các thông số SO2, CO, O3 được tính toán theo công thức 1

Giá trị AQIh của các thông số PM10, PM2.5 được tính toán theo công thức 2

Trong đó:

* AQIx: Giá trị AQI thông số của thông số x
* : Nồng độ giới hạn dưới của giá trị thông số quan trắc được quy định trong Bảng 2 tương ứng với mức i
* : Nồng độ giới hạn trên của giá trị thông số quan trắc được quy định
* trong Bảng 2 tương ứng với mức i+1
* Ii: Giá trị AQI ở mức i đã cho trong bảng tương ứng với giá trị i
* Ii+1: Giá trị AQI ở mức i+1 cho trong bảng tương ứng với giá trị
* Cx: Giá trị quan trắc trung bình 1 giờ của thông số x.
* Nowcastx: Giá trị Nowcast được tính toán ở phần a

*Bảng 2: Các giá trị đối với các thông số*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **i** | **Giá trị BPi quy định đối với từng thông số** (*Đơn vị: µg/m3)* | | | | | | |
| **O3(1h)** | **O3(8h)** | **CO** | **O2** | **O2** | **M10** | **M2.5** |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 50 | 160 | 100 | 10.000 | 125 | 100 | 50 | 25 |
| 3 | 100 | 200 | 120 | 30.000 | 350 | 200 | 150 | 50 |
| 4 | 150 | 300 | 170 | 45.000 | 550 | 700 | 250 | 80 |
| 5 | 200 | 400 | 210 | 60.000 | 800 | 1.200 | 350 | 150 |
| 6 | 300 | 800 | 400 | 90.000 | 1.600 | 2.350 | 420 | 250 |
| 7 | 400 | 1.000 | - | 120.000 | 2.100 | 3.100 | 500 | 350 |
| 8 | 500 | ≥ 1.200 | - | ≥ 150.000 | ≥ 2.630 | ≥ 3.850 | ≥ 600 | ≥ 500 |

1. ***Giá trị AQI giờ tổng hợp***

Sau khi đã có giá trị ***AQIx của mỗi thông số***, chọn giá trị AQI lớn nhất của các thông số để lấy làm giá trị ***AQI giờ tổng hợp***.

AQIh = max(AQIx)

Giá trị AQI giờ được làm tròn thành số nguyên

1. ***Giá trị AQI ngày tổng hợp***

Tương tự AQI giờ, ta dụa trên các giá trị

- Thông số PM2.5 và PM10: giá trị trung bình 24 giờ.

- Thông số O3: giá trị trung bình 1 giờ lớn nhất trong ngày và giá trị trung bình 8 giờ lớn nhất trong ngày.

- Thông số SO2, NO2 và CO: giá trị trung bình 1 giờ lớn nhất trong ngày

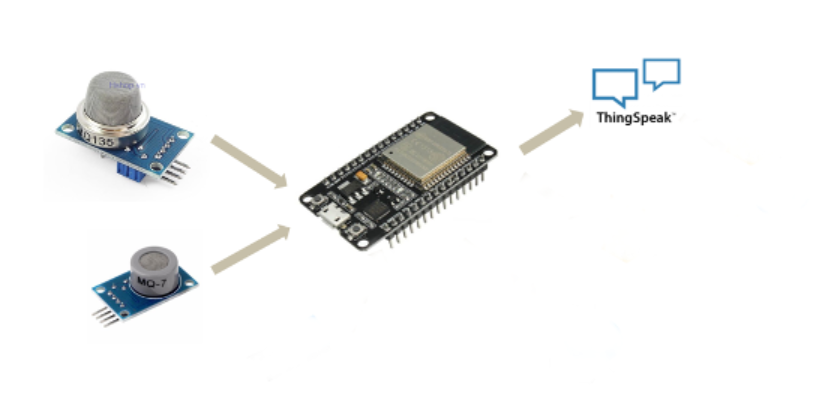
Tương tự, ta áp dụng công thức 1 ở trên để tính giá trị AQIx ngày của mỗi thông số, chọn giá trị AQI lớn nhất của các thông số để lấy làm giá trị AQI ngày tổng hợp.

AQId = max(AQIx)

Sau khi tính toán được chỉ số chất lượng không khí, sử dụng bảng xác định giá trị AQI tương ứng với mức cảnh boá chất lượng không khí và mức độ ảnh hưởng đến sức khoẻ con người để so sánh, đánh giá:

|  |  |
| --- | --- |
| Khoảng giá trị AQI | Ảnh hưởng tới sức khoẻ con người |
| 0 – 50  (Tốt) | Chất lượng không khí tốt, không ảnh hưởng tới sức khỏe |
| 51 – 100  (Trung bình) | Chất lượng không khí ở mức chấp nhận được. Tuy nhiên, đối với những người nhạy cảm (người già, trẻ em, người mắc các bệnh hô hấp, tim mạch…) có thể chịu những tác động nhất định tới sức khỏe |
| 101 – 150  (Kém) | Những người nhạy cảm gặp phải các vấn đề về sức khỏe, những người bình thường ít ảnh hưởng. |
| 151 – 200  (Xấu) | Những người bình thường bắt đầu có các ảnh hưởng tới sức khỏe, nhóm người nhạy cảm có thể gặp những vấn đề sức khỏe nghiêm trọng hơn. |
| 201 – 300  (Rất xấu) | Cảnh báo hưởng tới sức khỏe: mọi người bị ảnh hưởng tới sức khỏe nghiêm trọng hơn. |
| 301 – 500  (Nguy hiểm) | Cảnh báo khẩn cấp về sức khỏe: Toàn bộ dân số bị ảnh hưởng tới sức khỏe tới mức nghiêm trọng. |

# CHƯƠNG 2: XÂY DỰNG MÔ HÌNH



Tổng quan hệ thống

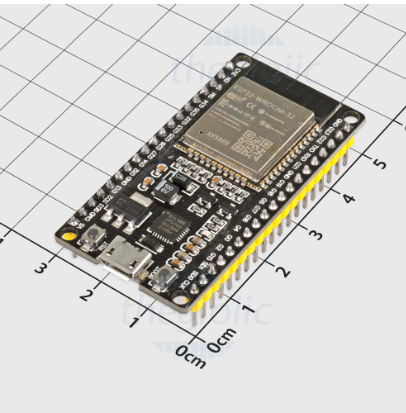
Hệ thống chúng em chủ yếu gồm khối xử lý trung tâm, khối giao tiếp, khối hiển thị

Khối xử lý trung tâm có nhiệm vụ đảm bảo vai trò đọc kết quả từ cảm biến, tính toán để gửi kết quả lên thinkspeak

Khối cảm biến gồm các cảm biến khác nhau như: cảm biến khí CO (MQ-7), cảm biến chất lượng không khí (MQ-135).

# CHƯƠNG 3: CÁC CẢM BIẾN SỬ DỤNG ĐO CHẤT LƯỢNG KHÔNG KHÍ

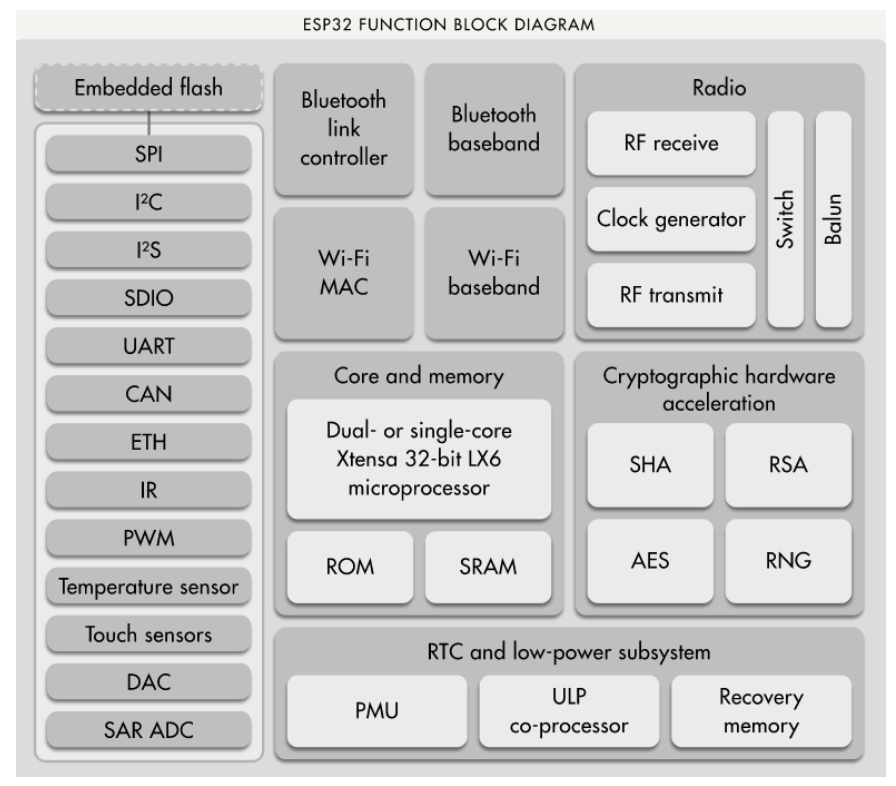
## 3.1. Vi điều khiển ESP32



ESP32 NodeMCU LuaNode32 là một module phát triển tiên tiến dựa trên vi điều khiển ESP32. Nó cung cấp khả năng kết nối WiFi và Bluetooth, cũng như nhiều chức năng khác như GPIO, UART, SPI, I2C và ADC.

Module này có 38 chân GPIO, cho phép bạn kết nối và điều khiển nhiều loại thiết bị và cảm biến. Nó cũng có chế độ ứng dụng phát sóng Wi-Fi, giúp bạn xây dựng các ứng dụng IoT

LuaNode32 cung cấp một môi trường lập trình Lua tương thích, cho phép bạn nhanh chóng và dễ dàng phát triển các ứng dụng. Bên cạnh đó, nó cũng hỗ trợ các phiên bản phát triển khác như Arduino IDE và MicroPython.**Cấu hình của ESP32**



**CPU**

* CPU: Xtensa Dual-Core LX6 microprocessor.
* Chạy hệ 32 bit
* Tốc độ xử lý từ 160 MHz đến 240 MHz
* ROM: 448 Kb
* Tốc độ xung nhịp từ 40 Mhz ÷ 80 Mhz (có thể tùy chỉnh khi lập trình)
* RAM: 520 Kb SRAM liền chip. Trong đó 8 Kb RAM RTC tốc độ cao – 8 Kb RAM RTC tốc độ thấp (dùng ở chế độ DeepSleep).

**Hỗ trợ 2 giao tiếp không dây**

* Wi-Fi: 802.11 b/g/n/e/I (Wifi 2.4 GHz)
* Station mode (Wi-Fi client hay STA). ESP32 có khả năng kết nối đến các điểm truy cập.
* Trở thành 1 điểm truy cập (Access Point mode hay Soft-AP). Lúc thành ESP trở thành trung tâm kết nối liên kết thông tin. Các Station kết nối với Access-Point chính là ESP32 tạo nên
* AP-STA mode ESP32 có thể đồng thời là điểm truy cập và có thể truy cập đến địa chỉ khác
* Bluetooth: v4.2 BR/EDR và BLE

Việc hỗ trợ cả bluetooth khiến tăng tính tương tác cho ESP32. Chúng có thể kết nối với các thiết bị như là chuột, bàn phím, hay các thiết bị thông minh như điện thoại, laptop khi mà không có Wi-Fi. Bạn có thể lựa chọn tùy biến chức năng này là BLE hay Bluetooth Classic. Tùy theo các yêu cầu về tốc độ và năng lượng mà project cần thiết

**Hỗ trợ tất cả các loại giao tiếp**

* 2 bộ chuyển đổi số sang tương tự (DAC) 8 bit
* 18 kênh bộ chuyển đổi tương tự sang số (ADC) 12 bit.
* 2 cổng giao tiếp I²C
* 3 cổng giao tiếp UART
* 3 cổng giao tiếp SPI (1 cổng cho chip FLASH )
* 2 cổng giao tiếp I²S
* 10 kênh ngõ ra điều chế độ rộng xung (PWM)
* SD card/SDIO/MMC host
* Ethernet MAC hỗ trợ chuẩn: DMA và IEEE 1588
* CAN bus 2.0
* IR (TX/RX)

**Cảm biến tích hợp trên chip ESP32**

* 1 cảm biến Hall (cảm biến từ trường)
* 1 cảm biến đo nhiệt độ
* Cảm biến chạm (điện dung) với 10 đầu vào khác nhau.

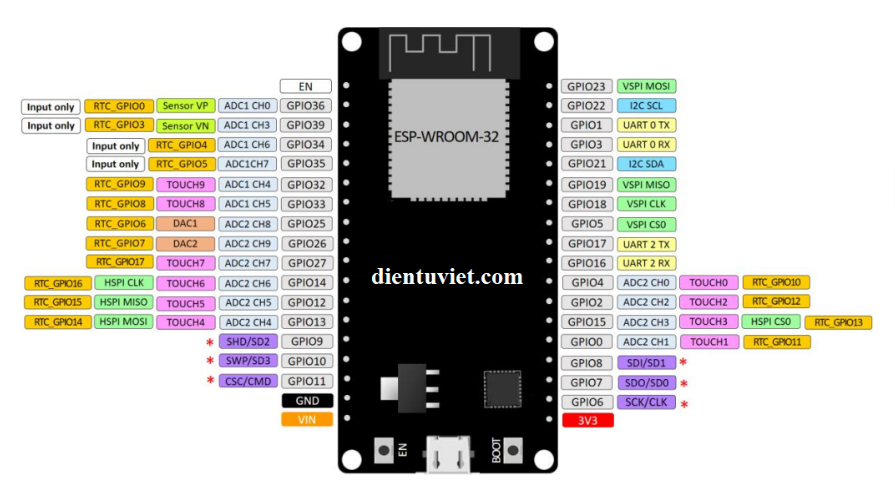
**Bảo mật**

* Hỗ trợ tất cả các tính năng bảo mật chuẩn IEEE 802.11, bao gồm WFA, WPA/WPA2 và WAPI
* Khởi động an toàn (Secure boot)
* Mã hóa flash (Flash encryption)
* 1024-bit OTP, lên đến 768-bit cho khách hàng
* Tăng tốc phần cứng mật mã: AES, SHA-2, RSA, mật mã đường cong elliptic (ECC – elliptic curve cryptography), bộ tạo số ngẫu nhiên (RNG – random number generator)

**Nguồn điện hoạt động**

* Điện áp hoạt động: 2,2V ÷ 3,6V
* Nhiệt độ hoạt động: -40oC ÷ + 85oC
* Số cổng GPIO: 36

**Sơ đồ chân của ESP32**



**Chân Input Only**

GPIO từ 34 đến 39 là các chân chỉ đầu vào. Các chân này không có điện trở kéo lên hoặc kéo xuống bên trong. Chúng không thể được sử dụng làm đầu ra, vì vậy chỉ sử dụng các chân này làm đầu vào:

* GPIO34
* GPIO35
* GPIO36
* GPIO39

**Chân tích hợp Flash trên ESP32**

GPIO 6 đến GPIO 11 dùng để kết nối Flash SPI trên chip ESP-WROOM-32, không khuyến khích sử dụng cho các mục đích sử dụng khác.

* GPIO6 (SCK/CLK)
* GPIO7 (SDO/SD0)
* GPIO8 (SDI/SD1)
* GPIO9 (SHD/SD2)
* GPIO10 (SWP/SD3)
* GPIO11 (CSC/CMD)

**Chân cảm biến điện dung**

ESP32 có 10 cảm biến điện dung bên trong. Các cảm biến này có thể phát hiện được sự thay đổi về điện áp cảm ứng trên các chân GPIO. Các chân cảm ứng điện dung cũng có thể được sử dụng để đánh thức ESP32 khỏi chế độ ngủ sâu (deep sleep).

Các chân ESP32 này có chức năng như 1 nút nhấn cảm ứng, có thể phát hiện sự thay đổi về điện áp cảm ứng trên chân.

Các cảm biến cảm ứng bên trong đó được kết nối với các GPIO sau:

* TOUCH0 (GPIO4)
* TOUCH1 (GPIO0)
* TOUCH2 (GPIO2)
* TOUCH3 (GPIO15)
* TOUCH4 (GPIO13)
* TOUCH5 (GPIO12)
* TOUCH6 (GPIO14)
* TOUCH7 (GPIO27)
* TOUCH8 (GPIO33)
* TOUCH9 (GPIO32)

**Bộ chuyển đổi tương tự sang số ADC (Analog to Digital Converter)**

ESP32 có 18 kênh đầu vào ADC 12 bit (trong khi ESP8266 chỉ có 1 kênh ADC 10 bit). Đây là các GPIO có thể được sử dụng làm ADC và các kênh tương ứng:

* ADC1\_CH0 (GPIO36)
* ADC1\_CH1 (GPIO37)
* ADC1\_CH2 (GPIO38)
* ADC1\_CH3 (GPIO39)
* ADC1\_CH4 (GPIO32)
* ADC1\_CH5 (GPIO33)
* ADC1\_CH6 (GPIO34)
* ADC1\_CH7 (GPIO35)
* ADC2\_CH0 (GPIO4)
* ADC2\_CH1 (GPIO0)
* ADC2\_CH2 (GPIO2)
* ADC2\_CH3 (GPIO15)
* ADC2\_CH4 (GPIO13)
* ADC2\_CH5 (GPIO12)
* ADC2\_CH6 (GPIO14)
* ADC2\_CH7 (GPIO27)
* ADC2\_CH8 (GPIO25)
* ADC2\_CH9 (GPIO26)

Các kênh đầu vào ADC có độ phân giải 12 bit. Điều này có nghĩa là chúng ta có thể nhận được các giá trị tương tự từ 0 đến 4095, trong đó 0 tương ứng với 0V và 4095 đến 3,3V. Ta cũng có thể thiết lập độ phân giải cho các kênh thông qua chương trình (code).

**Bộ chuyển đổi số sang tương tự DAC (Digital to Analog Converter)**

Có 2 kênh DAC 8 bit trên ESP32 để chuyển đổi tín hiệu số sang tương tự. Các kênh này chỉ có độ phân giải 8 bit, nghĩa là có giá trị từ 0 ÷ 255 tương ứng với 0 ÷ 3.3V

Đây là các kênh DAC:

* DAC1 (GPIO25)
* DAC2 (GPIO26)

**Các chân thời gian thực RTC**

Các chân này có tác dụng đánh thức ESP32 khi trong chế độ ngủ sâu (Low Power Mode). Sử dụng như 1 chân ngắt ngoài.

Các chân RTC:

* RTC\_GPIO0 (GPIO36)
* RTC\_GPIO3 (GPIO39)
* RTC\_GPIO4 (GPIO34)
* RTC\_GPIO5 (GPIO35)
* RTC\_GPIO6 (GPIO25)
* RTC\_GPIO7 (GPIO26)
* RTC\_GPIO8 (GPIO33)
* RTC\_GPIO9 (GPIO32)
* RTC\_GPIO10 (GPIO4)
* RTC\_GPIO11 (GPIO0)
* RTC\_GPIO12 (GPIO2)
* RTC\_GPIO13 (GPIO15)
* RTC\_GPIO14 (GPIO13)
* RTC\_GPIO15 (GPIO12)
* RTC\_GPIO16 (GPIO14)
* RTC\_GPIO17 (GPIO27)

**Chân PWM**

ESP32 LED PWM có 16 kênh độc lập có thể được cấu hình để tạo tín hiệu PWM với các thuộc tính khác nhau. Tất cả các chân có thể hoạt động như đầu ra đều có thể được sử dụng làm chân PWM (GPIO từ 34 đến 39 không thể tạo PWM).

Để xuất PWM, bạn cần định nghĩa các thông số này trong code:

* Tần số tín hiệu
* Chu kỳ làm việc
* Kênh PWM
* Chân GPIO xuất tín hiệu ra

**Chân I2C**

ESP32 có hai kênh I2C và bất kỳ chân nào cũng có thể được đặt làm SDA hoặc SCL. Khi sử dụng ESP32 với Arduino IDE, các chân I2C mặc định là:

* GPIO21 (SDA)
* GPIO22 (SCL)

Nếu chúng ta muốn sử dụng chân khác cho việc điều khiển I2C có thể sử dụng câu lệnh:

Wire.begin(SDA, SCL);

**Chân SPI**

Theo mặc định, ánh xạ cho chân SPI là:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SPI | MOSI | MISO | CLK | CS |
| VSPI | GPIO 23 | GPIO 19 | GPIO 18 | GPIO 5 |
| HSPI | GPIO 13 | GPIO 12 | GPIO 14 | GPIO 15 |

**Chân ngắt ngoài**

Tất cả các chân ESP đều có thể sử dụng ngắt ngoài.

**Ứng dụng**

* Module được dùng nhiều trong các ứng dụng thu thập dữ liệu và điều khiển thiết bị qua WiFi, Bluetooth.
* Sử dụng cho các ứng dụng tiết kiệm năng lượng, điều khiển mạng lưới cảm biến, mã hóa hoặc xử lí tiếng nói, xử lí Analog-Digital trong các ứng dụng phát nhạc, hoặc với các file MP3…

## 3.2 MQ-7 Mạch Cảm Biến Khí (CO)

MQ-7 là một loại cảm biến khí CO (carbon monoxide) được sử dụng để phát hiện mức độ khí CO trong môi trường. MQ-7 sử dụng công nghệ cảm biến điện hóa để phản ứng với khí CO và tạo ra một tín hiệu điện tử tương ứng.

Cảm biến MQ-7 thường được sử dụng trong các ứng dụng như hệ thống báo động khí CO, hệ thống điều khiển an toàn trong các phòng ngủ, nhà ở, nhà máy, xe hơi, và các ứng dụng liên quan đến giám sát môi trường.

Cảm biến khi CO – MQ7

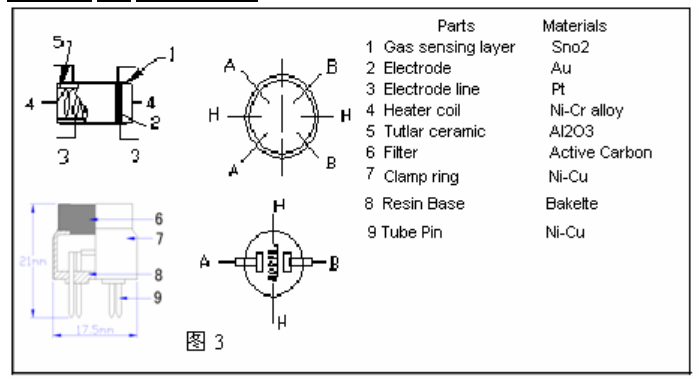


Cảm biến MQ-7 hoạt động dựa trên nguyên lý cảm biến điện hóa và phản ứng hóa học với khí CO. Dưới đây là nguyên tắc hoạt động cơ bản của MQ-7:

* Nguyên tắc cảm biến điện hóa: MQ-7 sử dụng một cặp điện cực (electrode) để tạo ra một môi trường điện hóa. Điện cực chủ yếu được làm bằng vật liệu như thiếc (Sn) và ôxit kẽm (ZnO). Khi khí CO tác động lên điện cực, nó phản ứng với các chất xúc tác trên bề mặt điện cực và tạo ra một sự thay đổi trong trạng thái điện tử của các chất xúc tác. Sự thay đổi này sẽ làm thay đổi điện trở của cảm biến.
* Phản ứng hóa học với khí CO: Khi khí CO tác động lên điện cực, nó tương tác với các chất xúc tác trên bề mặt điện cực. Quá trình này gây ra một phản ứng hóa học trong cảm biến MQ-7. Kết quả là, các electron được chuyển động trong cảm biến và tạo ra một tín hiệu điện tử.
* Đo và xử lý tín hiệu: Tín hiệu điện tử từ MQ-7 được đo và xử lý bởi mạch điện hoặc vi điều khiển. Thông qua quá trình này, tín hiệu điện tử từ cảm biến có thể được chuyển đổi thành mức độ khí CO trong môi trường.

**Dưới đây là một số thông số kỹ thuật cơ bản của cảm biến MQ-7:**

* Điện áp hoạt động: 5V DC.
* Dòng điện tiêu thụ: Khoảng 150mA.
* Điện trở đầu ra: Tín hiệu đầu ra của MQ-7 thay đổi tùy thuộc vào mức độ khí CO trong môi trường. Nó được biểu thị thông qua điện trở, có thể từ 2KΩ đến 20KΩ.
* Độ nhạy: MQ-7 có độ nhạy đối với khí CO trong khoảng từ 20ppm đến 2000ppm. Điều này có nghĩa là cảm biến có khả năng phát hiện và đo được mức độ khí CO trong phạm vi này.
* Thời gian khôi phục: MQ-7 có thời gian khôi phục sau khi tín hiệu đầu ra ổn định từ 60 đến 90 giây. Điều này có nghĩa là sau khi cảm biến bị tác động bởi khí CO, nó sẽ mất một khoảng thời gian để trở lại trạng thái ban đầu.
* Nhiệt độ hoạt động: MQ-7 hoạt động tốt trong khoảng nhiệt độ từ -10°C đến 50°C.
* **Cấu trúc của cảm biến MQ 7:**

****

Bao gồm AL2O3 và lớp cảm biến SnO2 đo dòng điện và cuộn sấy được cố định trong một lớp vỏ bằng nhựa và thép không gỉ. Cuộn sấy cung cấp điều kiện cần thiết để các thiết bị cảm biến làm việc. Các thành phần của cảm biến MQ-7 như sau:

1, Lớp cảm biến khí: SnO2

2, Điện cực: Au

3, Dây điện cực: Pt

4, Cuộn sấy: Ni-Cr alloy

5, Ống bằng gốm: Al2O3

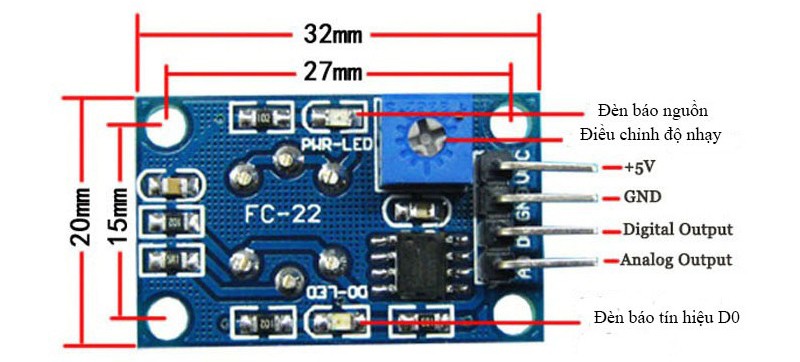
6, Chống nổ: thép không gỉ

7, Vòng kẹp: Đồng mạ Niken

8, Nhựa cơ sở: nhựa tổng hợp bakelite

9, Ống pin: Đồng mạ Niken

Mạch đo lường của thiết bị cảm biến MQ-7 được chia làm 2 phần: cuộn sấy có chức năng điều khiển theo chu kỳ cấp điện áp khác nhau trong mỗi 60s và 90s công suất tiêu thụ 350mW), phần thứ hai là phần cảm biến khí CO, nó có thể cho kết quả chính xác về sự thay đổi điện trở bề mặt của cảm biến.



*Sơ đồ cấu tạo MQ7*

## 3.3. Cảm Biến Chất Lượng Không Khí MQ-135



MQ-135 là một cảm biến chất lượng không khí dựa trên nguyên lý hoạt động của cảm biến khí tiêu hao. Nó được sử dụng để đo nồng độ các chất gây ô nhiễm trong không khí như khí CO2, khí NH3, khí NOx, và một số hợp chất hữu cơ bay hơi khác.

Cảm biến MQ-135 cung cấp một tín hiệu điện tử tương ứng với mức độ nồng độ của các chất gây ô nhiễm nhận được trong không khí. Tín hiệu này rơi vào một dải giá trị, và mức độ ô nhiễm càng lớn thì giá trị tín hiệu càng cao.

# CHƯƠNG 4: PHẦN CODE

## 4.1 Lập trình esp32

#include <stdio.h>

#include <stdlib.h>

#include <esp\_system.h>

#include "esp\_spi\_flash.h"

#include <stdio.h>

#include <freertos/FreeRTOS.h>

#include <freertos/task.h>

#include <math.h>

#include <driver/adc.h>

#include <esp\_http\_server.h>

#include <string.h>

#include "esp\_wifi.h"

#include "esp\_event.h"

#include "freertos/event\_groups.h"

#include "esp\_log.h"

#include "nvs\_flash.h"

#include "esp\_netif.h"

#include "driver/gpio.h"

#include <esp\_http\_client.h>

#include "lwip/err.h"

#include "lwip/sockets.h"

#include "lwip/sys.h"

#include "lwip/netdb.h"

#include "lwip/dns.h"

#include "esp\_adc\_cal.h"

#include "driver/adc.h"

#define WEB\_SERVER "api.thingspeak.com"

#define WEB\_PORT "80"

#define MQ135\_PIN GPIO\_NUM\_32

#define MQ135\_BASE\_VOLTAGE 200 // Điện áp cơ sở của MQ135 trong môi trường sạch (mV)

#define MQ135\_BASE\_RZERO 10000 // RZERO của MQ135 trong môi trường sạch (ohm)

#define MQ135\_RZERO\_COEFFICIENT 0.5 // Hệ số nhiệt độ RZERO (tính %/°C)

#define MQ135\_VOLTAGE\_DIVIDER 5 // Giá trị chia điện áp (tùy chỉnh tùy vào cấu hình mạch)

#define DEFAULT\_VREF 1100 // Điện áp tham chiếu mặc định (1100 mV)

#define NO\_OF\_SAMPLES 64 // Số lượng mẫu ADC để lấy trung bình (64 mẫu)

#define WIFI\_SSID "TP-Link\_454C"

#define WIFI\_PASS "63609875"

static const adc\_channel\_t channel = ADC\_CHANNEL\_5; // Kênh ADC được sử dụng (GPIO33)

static const adc\_atten\_t atten = ADC\_ATTEN\_DB\_0; // Độ rơi áp lớn nhất (0 dB)

static const adc\_unit\_t unit = ADC\_UNIT\_1; // Đơn vị ADC (ADC1)

static esp\_adc\_cal\_characteristics\_t \*adc\_chars;

static const char \*TAG = "MQ7";

char REQUEST[2000];

char SUBREQUEST[100];

char RESPONSE[2000];

char recv\_buf[2000];

static void wifi\_event\_handler(void\* arg, esp\_event\_base\_t event\_base,

int32\_t event\_id, void\* event\_data)

{

if (event\_id == WIFI\_EVENT\_STA\_START)

{

esp\_wifi\_connect();

}

else if (event\_id == WIFI\_EVENT\_STA\_DISCONNECTED)

{

esp\_wifi\_connect();

}

}

void wifi\_init\_sta(void)

{

esp\_netif\_init();

esp\_event\_loop\_create\_default();

esp\_netif\_create\_default\_wifi\_sta();

wifi\_init\_config\_t cfg = WIFI\_INIT\_CONFIG\_DEFAULT();

esp\_wifi\_init(&cfg);

esp\_event\_handler\_instance\_t instance\_any\_id;

esp\_event\_handler\_instance\_t instance\_got\_ip;

esp\_event\_handler\_instance\_register(WIFI\_EVENT,

ESP\_EVENT\_ANY\_ID,

&wifi\_event\_handler,

NULL,

&instance\_any\_id);

esp\_event\_handler\_instance\_register(IP\_EVENT,

IP\_EVENT\_STA\_GOT\_IP,

&wifi\_event\_handler,

NULL,

&instance\_got\_ip);

wifi\_config\_t wifi\_config = {

.sta = {

.ssid = WIFI\_SSID,

.password = WIFI\_PASS,

.pmf\_cfg = {

.capable = true,

.required = false

},

},

};

esp\_wifi\_set\_mode(WIFI\_MODE\_STA);

esp\_wifi\_set\_config(WIFI\_IF\_STA, &wifi\_config);

esp\_wifi\_start();

}

float estimate\_CO2\_ppm(float raw\_value)

{

// Chuyển đổi giá trị analog thành điện áp (mV)

float voltage = (raw\_value / 4095.0) \* 3300 \* MQ135\_VOLTAGE\_DIVIDER;

// Chuyển đổi điện áp thành giá trị RZERO (ohm)

float rzero = (MQ135\_BASE\_RZERO / (voltage / MQ135\_BASE\_VOLTAGE - 1));

// Ước tính ppm CO2 dựa trên công thức tương quan

float ppm = 5000 \* pow((rzero / MQ135\_BASE\_RZERO), MQ135\_RZERO\_COEFFICIENT);

return ppm;

}

void initialize\_adc()

{

// Khởi tạo ADC

adc1\_config\_width(ADC\_WIDTH\_BIT\_12);

adc1\_config\_channel\_atten(ADC1\_CHANNEL\_0, ADC\_ATTEN\_DB\_0);

}

void connect\_wifi()

{

// Khởi tạo NVS Flash để lưu trữ cấu hình Wi-Fi

esp\_err\_t ret = nvs\_flash\_init();

if (ret == ESP\_ERR\_NVS\_NO\_FREE\_PAGES || ret == ESP\_ERR\_NVS\_NEW\_VERSION\_FOUND)

{

ESP\_ERROR\_CHECK(nvs\_flash\_erase());

ret = nvs\_flash\_init();

}

ESP\_ERROR\_CHECK(ret);

// Kết nối Wi-Fi

wifi\_init\_sta();

}

static void send\_to\_thingspeak(float ppm\_CO2\_0, float ppm\_CO\_0)

{

const struct addrinfo hints = {

.ai\_family = AF\_INET,

.ai\_socktype = SOCK\_STREAM,

};

struct addrinfo \*res;

struct in\_addr \*addr;

int s, r;

int err = getaddrinfo(WEB\_SERVER, WEB\_PORT, &hints, &res);

if(err != 0 || res == NULL) {

ESP\_LOGE(TAG, "DNS lookup failed err=%d res=%p", err, res);

return;

}

addr = &((struct sockaddr\_in \*)res->ai\_addr)->sin\_addr;

ESP\_LOGI(TAG, "DNS lookup succeeded. IP=%s", inet\_ntoa(\*addr));

s = socket(res->ai\_family, res->ai\_socktype, 0);

if(s < 0) {

ESP\_LOGE(TAG, "... Failed to allocate socket.");

freeaddrinfo(res);

return;

}

ESP\_LOGI(TAG, "... allocated socket");

if(connect(s, res->ai\_addr, res->ai\_addrlen) != 0) {

ESP\_LOGE(TAG, "... socket connect failed errno=%d", errno);

close(s);

freeaddrinfo(res);

return;

}

ESP\_LOGI(TAG, "... connected");

freeaddrinfo(res);

sprintf(SUBREQUEST, "api\_key=4XLRCO8NGQS2HPMX&field1=%f&field2=%f", ppm\_CO2\_0, ppm\_CO\_0);

sprintf(REQUEST, "POST /update.json HTTP/1.1\nHost: api.thingspeak.com\nConnection: close\n Content-Type: application/x-www-form-urlencoded\nContent-Length:%d\n\n%s\n", strlen(SUBREQUEST), SUBREQUEST);

if (write(s, REQUEST, strlen(REQUEST)) < 0) {

ESP\_LOGE(TAG, "... socket send failed");

close(s);

return;

}

ESP\_LOGI(TAG, "Data send to Thingspeak");

close(s);

}

void app\_main()

{

ESP\_ERROR\_CHECK( nvs\_flash\_init() );

ESP\_ERROR\_CHECK(esp\_netif\_init());

ESP\_ERROR\_CHECK(esp\_event\_loop\_create\_default());

// Khởi tạo Wi-Fi và kết nối

connect\_wifi();

initialize\_adc();

adc1\_config\_width(ADC\_WIDTH\_BIT\_12); // Độ phân giải ADC 12 bit (0-4095)

adc1\_config\_channel\_atten(channel, atten);

// Hiệu chỉnh ADC

adc\_chars = calloc(1, sizeof(esp\_adc\_cal\_characteristics\_t));

esp\_adc\_cal\_value\_t val\_type = esp\_adc\_cal\_characterize(unit, atten, ADC\_WIDTH\_BIT\_12, DEFAULT\_VREF, adc\_chars);

uint32\_t adc\_reading;

float voltage;

float ppm\_CO;

while (1)

{

// Đọc giá trị từ cảm biến MQ135

float mq135\_value = adc1\_get\_raw(ADC1\_CHANNEL\_4); // ADC1\_CHANNEL\_4 tương ứng với GPIO\_NUM\_32

// Chuyển đổi giá trị đọc được thành ppm CO2

float ppm\_CO2 = estimate\_CO2\_ppm(mq135\_value);

// Đọc giá trị ADC từ cảm biến MQ7

adc\_reading = 0;

for (int i = 0; i < NO\_OF\_SAMPLES; i++)

{

adc\_reading += adc1\_get\_raw((adc1\_channel\_t)channel);

}

adc\_reading /= NO\_OF\_SAMPLES;

// Chuyển giá trị ADC sang đơn vị điện áp (V)

voltage = esp\_adc\_cal\_raw\_to\_voltage(adc\_reading, adc\_chars);

// Tính giá trị ppm CO dựa trên đoạn cong tương quan từ datasheet của MQ7

ppm\_CO = (voltage - 2.2) / 1.0;

send\_to\_thingspeak(ppm\_CO2, ppm\_CO);

vTaskDelay(pdMS\_TO\_TICKS(5000)); // gửi data mỗi 5 giây

}

}

## 4.2. Kết quả kiểm thử

Bảng 1: Kết quả đo CO2

|  |  |
| --- | --- |
| Thời gian đo | Giá trị (ppm) |
| 23:05:30 | 711.92 |
| 23:05:42 | 713.38 |
| 23:05:58 | 714.55 |
| 23:06:14 | 714.41 |
| 23:06:30 | 714.41 |
| 23:06:46 | 713.23 |
| 23:07:02 | 713.53 |
| 23:07:18 | 713.09 |
| 23:07:34 | 713.70 |
| 23:07:50 | 713.67 |

Bảng 2: Kết quả đo CO

|  |  |
| --- | --- |
| Thời gian đo | Kết quả đo (ppm) |
| 23:04:22 | 605.40 |
| 23:04:38 | 605.40 |
| 23:04:54 | 605.30 |
| 23:05:10 | 605.40 |
| 23:05:26 | 605.40 |
| 23:05:42 | 605.40 |
| 23:05:58 | 605.40 |
| 23:06:14 | 605.40 |
| 23:06:30 | 605.40 |
| 23:06:46 | 605.40 |

**Hướng phát triển**

Hoàn thiện sản phẩm hoạt động chính xác hơn.

Kết nối thêm các cảm biến đo khí khác như NO2, SO2, … để nâng cao kết quả đánh giá.

# TÀI LIỆU THAM KHẢO

[1] [ESP-IDF Programming Guide - ESP32 - — ESP-IDF Programming Guide latest documentation (espressif.com)](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/)

[2] [Quyết định 1459/QĐ-TCMT 2019 kỹ thuật tính toán và công bố chỉ số chất lượng không khí Việt Nam (thuvienphapluat.vn)](https://thuvienphapluat.vn/van-ban/Tai-nguyen-Moi-truong/Quyet-dinh-1459-QD-TCMT-2019-ky-thuat-tinh-toan-va-cong-bo-chi-so-chat-luong-khong-khi-Viet-Nam-428215.aspx)

[3] [Chỉ số chất lượng không khí là gì và mối liên hệ tới sức khỏe | Vinmec](https://www.vinmec.com/vi/tin-tuc/thong-tin-suc-khoe/chi-so-chat-luong-khong-khi-la-gi-va-moi-lien-he-toi-suc-khoe/)

[4] [Xây dựng, phát triển đô thị thông minh trong bối cảnh đô thị hóa & chuyển đổi số ở Việt Nam (moc.gov.vn)](https://moc.gov.vn/tl/tin-tuc/74979/xay-dung-phat-trien-do-thi-thong-minh-trong-boi-canh-do-thi-hoa-chuyen-doi-so-o-viet-nam.aspx)

[5] [Ô nhiễm không khí ở Việt Nam (who.int)](https://www.who.int/vietnam/vi/health-topics/air-pollution)

[6] [MQ7 datasheet](https://www.pololu.com/file/0J313/MQ7.pdf)

[7] [Hướng dẫn kĩ thuật tính toán và công bố chỉ số chất lượng ở Việt Nam](https://thuvienphapluat.vn/van-ban/Tai-nguyen-Moi-truong/Quyet-dinh-1459-QD-TCMT-2019-ky-thuat-tinh-toan-va-cong-bo-chi-so-chat-luong-khong-khi-Viet-Nam-428215.aspx)