



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

SỐ PHÁCH:.....

ĐỀ TÀI TIỂU LUẬN

GIÁM SÁT CHẤT LƯỢNG KHÔNG KHÍ VỚI ESP32 VÀ CẢM BIẾN MQ

PHÁT TRIỂN ỨNG DỤNG IoT – NHÓM 5 –

2024-2025.2.TIN4024.005

GIẢNG VIÊN HƯỚNG DẪN: VÕ VIỆT DŨNG

HUẾ, THÁNG 04 NĂM 202

MỤC LỤC

MỞ ĐẦU	5
1. Đặt vấn đề.....	5
2. Mục tiêu đề tài.....	5
3. Nội dung nghiên cứu	5
4. Phương pháp nghiên cứu.....	6
5. Kết quả dự kiến	6
6. Cấu trúc đề án.....	6
CHƯƠNG 1 - TỔNG QUAN VỀ Ô NHIỄM KHÔNG KHÍ VÀ CÁC CÔNG NGHỆ NỀN TẢNG.....	7
1. Mức độ quan trọng của việc giám sát chất lượng không khí	7
1.1. Hiện trạng ô nhiễm không khí hiện nay.....	7
2. Hiện trạng hệ thống giám sát chất lượng không khí tại Huế	8
3. Cấu trúc chung của hệ thống giám sát chất lượng không khí hiện đại	9
4. Các công nghệ trong hệ thống giám sát chất lượng không khí	11
4.1. Công nghệ IoT (Internet of Things)	11
4.2. Nền tảng Blynk.....	16
4.3. Telegram	19
4.4. Phần cứng và phần mềm hỗ trợ	21
4.5. Các công nghệ khác	23
5. KẾT LUẬN CHƯƠNG 1.....	24
CHƯƠNG 2 - THIẾT KẾ HỆ THỐNG GIÁM SÁT CHẤT LƯỢNG KHÔNG KHÍ	25
1. Mô hình hệ thống	25
2. Thiết kế hệ thống.....	26
2.1. Thiết kế phần cứng	26
2.2. Phần mềm	35
3. Kết luận chương 2	38
CHƯƠNG 3 - THỰC HIỆN VÀ ĐÁNH GIÁ HỆ THỐNG GIÁM SÁT CHẤT LƯỢNG KHÔNG KHÍ	39
1. Quá trình thực hiện hệ thống	39
1.1. Chuẩn bị phần cứng.....	39

1.2.	Cấu hình phần mềm.....	40
1.3.	Triển khai thực tế	41
2.	Kết quả thử nghiệm	42
2.1.	Dữ liệu thu thập được	42
2.2.	Hiệu quả hoạt động.....	46
3.	Đánh giá hệ thống	47
3.1.	Ưu điểm	47
3.2.	Hạn chế.....	48
3.3.	Đề xuất cải tiến.....	49
4.	Kết luận chương 3	50
KẾT LUẬN		51
TÀI LIỆU THAM KHẢO		53

LỜI CẢM ƠN

Để hoàn thành bài báo cáo này, em xin gửi lời cảm ơn chân thành và sâu sắc đến giảng viên Nguyễn Đăng Bình đã tận tình hướng dẫn, chỉ bảo nhóm và tạo điều kiện tốt nhất cho nhóm trong suốt quá trình học tập tại trường đại học Khoa học Huế.

Em xin chân thành bày tỏ lòng biết ơn đối với thầy cô trong trường đại học Khoa học Huế đã truyền đạt hết kiến thức quý báu đến cho nhóm để hoàn thành bài báo cáo này.

Mặc dù nỗ lực cố gắng thực hiện bài báo cáo này, nhưng em chưa có nhiều kinh nghiệm thực tập nên chắc hẳn bài báo cáo này không thể tránh khỏi thiếu sót, nhóm rất mong nhận được ý kiến từ thầy cô và các anh chị học sinh, sinh viên quan tâm.

Em xin chân thành cảm ơn!

Huế, ngày 12 tháng 04 năm 2025

Sinh viên thực hiện

MỞ ĐẦU

1. Đặt vấn đề

Trong bối cảnh ô nhiễm không khí ngày càng gia tăng và trở thành mối đe dọa nghiêm trọng đối với sức khỏe con người cũng như hệ sinh thái toàn cầu, nhu cầu giám sát chất lượng không khí một cách hiệu quả và kịp thời chưa bao giờ cấp thiết đến thế. Các nguồn ô nhiễm từ giao thông, công nghiệp, và sinh hoạt hàng ngày đã làm gia tăng nồng độ các khí độc hại như CO, góp phần gây ra các bệnh lý nguy hiểm về hô hấp, tim mạch, đồng thời làm trầm trọng thêm các vấn đề môi trường như mưa axit và biến đổi khí hậu. Trước thực trạng này, công nghệ Internet of Things (IoT) đã nổi lên như một giải pháp đột phá, cho phép kết nối các thiết bị thông minh để thu thập, phân tích và chia sẻ dữ liệu thời gian thực, từ đó hỗ trợ con người đưa ra các quyết định chính xác nhằm giảm thiểu tác động của ô nhiễm.

Nhận thấy tiềm năng to lớn của IoT trong việc cải thiện chất lượng cuộc sống, tôi đã quyết định nghiên cứu và phát triển một hệ thống giám sát nồng độ khí CO sử dụng module ESP32 tích hợp với cảm biến MQ135, kết hợp cùng các nền tảng đám mây như Blynk và Telegram. Hệ thống không chỉ cung cấp khả năng theo dõi liên tục mà còn cho phép điều khiển từ xa thông qua nút nhấn vật lý, ứng dụng di động và tin nhắn, mang lại sự linh hoạt và tiện lợi. Điểm nổi bật của dự án nằm ở việc sử dụng các linh kiện giá thành thấp, dễ tiếp cận, cùng quy trình thiết kế đơn giản, giúp sản phẩm có thể được ứng dụng rộng rãi trong các hộ gia đình, phòng thí nghiệm giáo dục, hoặc thậm chí mở rộng để giám sát chất lượng không khí tại các đô thị lớn ở Việt Nam.

2. Mục tiêu đề tài

Mục tiêu chính của tiểu luận là nghiên cứu công nghệ IoT dựa trên nền tảng ESP32, tích hợp với các dịch vụ đám mây để xây dựng một hệ thống giám sát chất lượng không khí hiệu quả. Dự án hướng đến việc cung cấp một giải pháp giám sát nồng độ khí CO trong thời gian thực, đồng thời hỗ trợ cảnh báo và điều khiển từ xa, phục vụ nhu cầu bảo vệ sức khỏe và nâng cao nhận thức cộng đồng về ô nhiễm không khí.

3. Nội dung nghiên cứu

- Tìm hiểu thực trạng ô nhiễm không khí và nhu cầu giám sát chất lượng không khí hiện nay.
- Nghiên cứu công nghệ IoT và ứng dụng trong hệ thống giám sát môi trường.
- Phân tích đặc điểm và khả năng của module ESP32 cùng cảm biến MQ135.
- Thiết kế và xây dựng mô hình thực nghiệm giám sát nồng độ CO.
- Thực hiện các kịch bản thử nghiệm và đánh giá hiệu quả của hệ thống.

4. Phương pháp nghiên cứu

Đề tài sử dụng phương pháp nghiên cứu lý thuyết kết hợp với thực nghiệm. Các bước thực hiện bao gồm tổng hợp tài liệu, phân tích yêu cầu hệ thống, thiết kế mô hình phần cứng và phần mềm, sau đó tiến hành thử nghiệm để đánh giá tính khả thi và hiệu quả của giải pháp.

5. Kết quả dự kiến

- Một báo cáo tiểu luận chi tiết về quá trình nghiên cứu và phát triển hệ thống.
- Mô hình thực nghiệm hoàn chỉnh có khả năng giám sát và điều khiển từ xa trên Blynk và Telegram.

6. Cấu trúc đề án

Đề án được chia thành 3 chương chính:

- **Chương 1:** Tổng quan về ô nhiễm không khí, vai trò của IoT trong giám sát chất lượng không khí và giới thiệu hệ thống đề xuất.
- **Chương 2:** Phân tích, thiết kế hệ thống giám sát chất lượng không khí, bao gồm phần cứng ESP32, Blynk, Telegram và các thuật toán điều khiển.
- **Chương 3:** Trình bày quá trình thực nghiệm, kết quả thử nghiệm và đánh giá hiệu suất của hệ thống.

CHƯƠNG 1 - TỔNG QUAN VỀ Ô NHIỄM KHÔNG KHÍ VÀ CÁC CÔNG NGHỆ NỀN TẢNG

1. Mức độ quan trọng của việc giám sát chất lượng không khí

1.1. Hiện trạng ô nhiễm không khí hiện nay

1.1.1. Tình hình ô nhiễm không khí trên toàn cầu

Ô nhiễm không khí từ lâu đã được ghi nhận là một vấn đề nghiêm trọng đe dọa sức khỏe con người và môi trường toàn cầu. Theo báo cáo "Tình trạng không khí toàn cầu năm 2020" của Viện Ảnh hưởng Sức khỏe (HEI) phối hợp với Viện Đo lường và Đánh giá Sức khỏe (IHME), gần 99% dân số thế giới đang sống trong điều kiện không khí vượt quá ngưỡng an toàn do Tổ chức Y tế Thế giới (WHO) quy định. Các hạt bụi mịn PM_{2.5}, khí CO, NO₂, SO₂ và các chất ô nhiễm khác là nguyên nhân chính gây ra hơn 7 triệu ca tử vong mỗi năm, chủ yếu liên quan đến các bệnh về hô hấp, tim mạch và đột quỵ.

1.1.2. Tình hình ô nhiễm không khí ở Việt Nam

Tại Việt Nam, ô nhiễm không khí cũng đang ở mức báo động. Theo báo cáo của IQAir năm 2022, Việt Nam nằm trong top 10 quốc gia có mức độ ô nhiễm không khí cao nhất khu vực châu Á, với nồng độ bụi mịn PM_{2.5} và PM₁₀ thường xuyên vượt ngưỡng cho phép. Các thành phố lớn như Hà Nội và TP. Hồ Chí Minh chịu ảnh hưởng nặng nề từ khí thải giao thông, hoạt động công nghiệp và sinh hoạt đô thị. Mặc dù có sự cải thiện tạm thời vào năm 2021 do giãn cách xã hội trong đại dịch COVID-19, tình trạng ô nhiễm vẫn chưa được giải quyết triệt để.

1.1.3. Tình trạng ô nhiễm không khí ở Huế

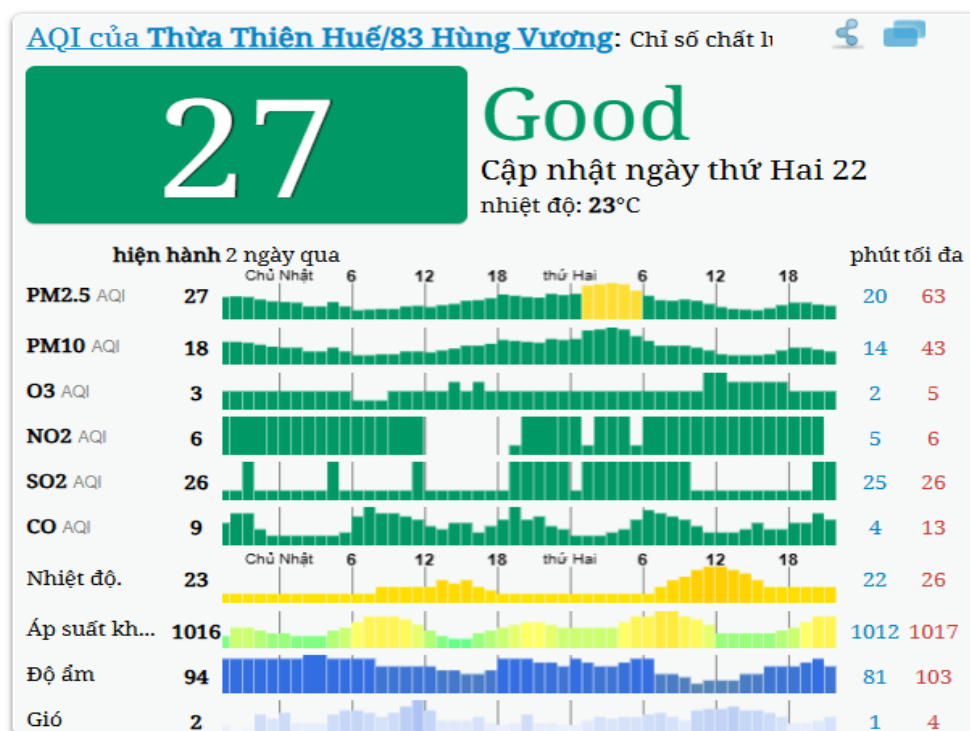
Huế, thủ phủ của tỉnh Thừa Thiên Huế, dù không phải là một thành phố công nghiệp lớn như Hà Nội hay TP. Hồ Chí Minh, vẫn đối mặt với các vấn đề ô nhiễm không khí ở mức độ nhất định. Theo báo cáo từ IQAir năm 2019, nồng độ trung bình PM_{2.5} tại Huế đạt 28.6 µg/m³, được xếp vào mức "Trung bình" (Moderate). Tuy nhiên, có 5 tháng trong năm, chất lượng không khí rơi vào mức "Không lành mạnh cho nhóm nhạy cảm" (Unhealthy for Sensitive Groups), với các yếu tố ô nhiễm chủ yếu đến từ bụi mịn PM_{2.5}, khí CO từ giao thông và khói bụi từ hoạt động sinh hoạt. Đặc biệt, vào mùa khô (tháng 10 đến tháng 2),

việc đốt rơm rạ ở các vùng ngoại ô và khói từ các lò đốt thủ công truyền thống làm gia tăng nồng độ chất ô nhiễm.

So với các thành phố lớn khác, Huế có lợi thế nhờ vị trí địa lý gần biển Đông, giúp phân tán một phần chất ô nhiễm. Tuy nhiên, sự gia tăng phương tiện giao thông (xe máy và ô tô) cùng với các công trình xây dựng trong quá trình đô thị hóa đang dần gây áp lực lên chất lượng không khí. Ngoài ra, các sự kiện thời tiết bất thường như mưa lớn hoặc nghịch nhiệt cũng có thể làm trầm trọng thêm tình trạng ô nhiễm trong ngắn hạn, như ghi nhận từ một số bài viết trên mạng xã hội vào năm 2024 trước thềm Festival Huế.

2. Hiện trạng hệ thống giám sát chất lượng không khí tại Huế

Tại Huế, hệ thống giám sát không khí hiện chủ yếu do Sở Tài nguyên và Môi trường Thừa Thiên Huế quản lý, với một số trạm quan trắc tự động được lắp đặt từ giai đoạn 2020-2025 theo kế hoạch của Ủy ban Nhân dân tỉnh. Các trạm này đo lường các chỉ số như PM2.5, PM10, CO, SO₂, NO₂, O₃, cùng với các thông số về nhiệt độ và độ ẩm,... nhưng số lượng còn hạn chế và chưa phủ đều khắp khu vực đô thị lẫn ngoại ô. Dữ liệu từ các trạm cho thấy nồng độ CO trung bình tại một số khu vực đông dân cư thường dao động ở mức đáng lo ngại, đặc biệt gần các tuyến đường chính như Hùng Vương hay Lê Lợi.



Hình 1.1. Bảng trạm kiểm soát đường Hùng Vương ở Huế

Dữ liệu thu thập từ các cảm biến được truyền tải thời gian thực qua mạng đến các trung tâm quản lý, cho phép cơ quan chức năng theo dõi tình trạng không khí và phản ứng nhanh chóng khi có tình trạng ô nhiễm. Thông tin chi tiết về chất lượng không khí được phân tích và hiển thị thông qua các ứng dụng di động, trang web hoặc các bảng điện tử, giúp cộng đồng và những người quan tâm có thể theo dõi tình trạng không khí tại các khu vực khác nhau.

Hệ thống cảnh báo tự động sẽ hoạt động khi chất lượng không khí xuất hiện tình trạng nguy hại cho sức khỏe, thông báo cho người dân thông qua tin nhắn SMS, ứng dụng di động hoặc các phương tiện truyền thông khác. Dữ liệu về chất lượng không khí thường được công khai và chia sẻ với cộng đồng, giúp tăng cường nhận thức và sự tham gia trong việc bảo vệ môi trường.

Dự án của tôi, sử dụng ESP32 và cảm biến MQ135 kết hợp với Blynk và Telegram, có thể đóng góp vào việc xây dựng mạng lưới thiết bị theo dõi không khí tiết kiệm chi phí, cung cấp dữ liệu thời gian thực và cảnh báo nhanh chóng cho người dân. Điều này đặc biệt hữu ích trong bối cảnh thành phố đang đẩy mạnh phát triển du lịch bền vững, vốn yêu cầu không khí sạch để bảo vệ sức khỏe cư dân và du khách tham quan các di sản UNESCO như Đại Nội.

3. Cấu trúc chung của hệ thống giám sát chất lượng không khí hiện đại

Hệ thống giám sát chất lượng không khí hiện đại là một giải pháp công nghệ tích hợp giữa phần cứng và phần mềm, thường được phát triển dựa trên nền tảng Internet of Things (IoT) kết hợp với điện toán đám mây nhằm đảm bảo khả năng thu thập, xử lý và phân tích dữ liệu môi trường theo thời gian thực. Cấu trúc tổng thể của hệ thống giám sát chất lượng không khí (GSCLKK) hiện nay thường bao gồm các thành phần chính như sau:

- **Cảm biến và thiết bị đo đạc:**

Hệ thống bao gồm một mạng cảm biến được đặt tại các vị trí chiến lược trong khu vực cần giám sát. Cảm biến có khả năng đo các chỉ số quan trọng về chất lượng không khí như hạt mịn (PM2.5, PM10), khí CO, NO₂, SO₂, O₃, nhiệt độ và độ ẩm. Các thiết bị đo đạc thường được cài đặt theo chuẩn quốc tế để đảm bảo tính chính xác và đáng tin cậy của dữ liệu.

- **Thu thập và truyền tải dữ liệu:**

Dữ liệu từ cảm biến được thu thập liên tục hoặc theo khoảng thời gian xác định. Dữ liệu sau đó được truyền tải qua mạng (mạng Internet hoặc mạng di động) đến các trung tâm quản lý và xử lý.

- **Xử lý và phân tích dữ liệu**

Dữ liệu từ các cảm biến được xử lý để loại bỏ nhiễu và sai sót có thể xảy ra trong quá trình thu thập. Các giá trị đo được thường được so sánh với các chuẩn mức chất lượng không khí quốc gia hoặc quốc tế để xác định tình trạng ô nhiễm.

- **Hiển thị và truyền tải thông tin**

Dữ liệu về chất lượng không khí thường được hiển thị qua các giao diện trực quan như ứng dụng di động, trang web hoặc bảng điện tử. Thông tin này có thể được trình bày dưới dạng biểu đồ, bản đồ, chỉ số chất lượng không khí và các thông báo cảnh báo.

- **Cảnh báo và thông báo**

Hệ thống thường có khả năng tự động phát ra cảnh báo khi chất lượng không khí xuất hiện tình trạng nguy hại cho sức khỏe. Cảnh báo có thể được gửi đến người dân qua ứng dụng di động, tin nhắn SMS, email hoặc thông báo trực tiếp trên các thiết bị.

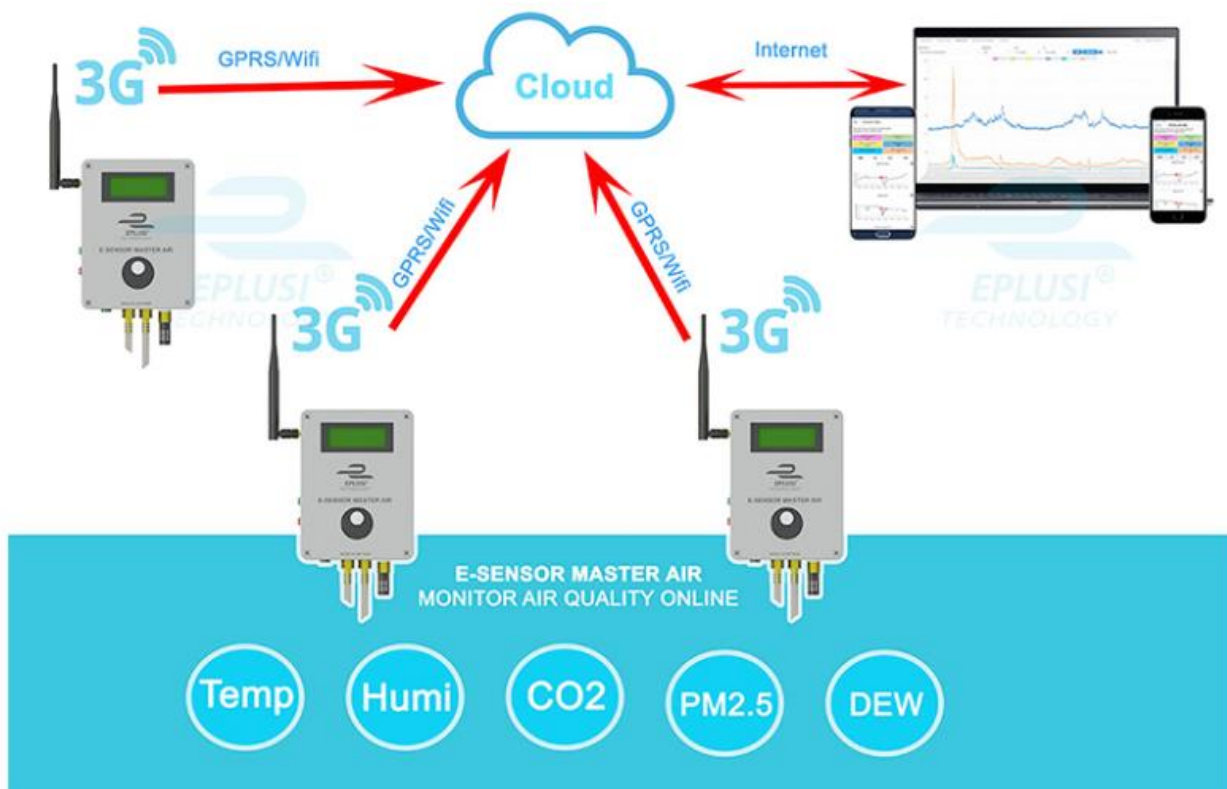
- **Lưu trữ và chia sẻ dữ liệu**

Dữ liệu về chất lượng không khí thường được lưu trữ để tạo ra lịch sử và phân tích dài hạn. Thông tin này thường được chia sẻ công khai để người dân, các nhà nghiên cứu và cơ quan quản lý có thể truy cập và sử dụng.

- **Quản lý và bảo trì hệ thống**

Hệ thống cần được quản lý và bảo trì thường xuyên để đảm bảo hoạt động ổn định và đáng tin cậy của các cảm biến và thiết bị. Cấu trúc chung này giúp hệ thống giám sát chất lượng không khí hoạt động hiệu quả, cung cấp thông tin quan trọng về tình trạng ô nhiễm không khí và hỗ trợ quyết định về quản lý môi trường và bảo vệ sức khỏe cộng đồng.

Hình 1.2 là một ví dụ tiêu biểu về hệ thống giám sát chất lượng không khí. Sử dụng mạng 3G và điện toán đám mây. Đây là hệ thống giám sát bụi: PM2.5, CO2, nhiệt độ, độ ẩm không khí, điểm sương ngoài trời của Công ty Eplusi.



Hình 1.2 Ví dụ về hệ thống GSCLKK

4. Các công nghệ trong hệ thống giám sát chất lượng không khí

Hệ thống giám sát chất lượng không khí mà tôi phát triển là một giải pháp toàn diện, dựa trên công nghệ hiện đại nhằm mục tiêu đo lường, xử lý, hiển thị và cảnh báo về các thông số môi trường quan trọng như nồng độ khí CO, nhiệt độ và độ ẩm – một thành phố có khí hậu nhiệt đới gió mùa đặc trưng với độ ẩm cao và thời tiết thay đổi theo mùa. Hệ thống này không chỉ cung cấp dữ liệu thời gian thực mà còn cho phép người dùng tương tác từ xa, phù hợp với nhu cầu giám sát trong nhà hoặc ngoài trời tại các khu vực dân cư, trường học, hoặc cơ quan. Các công nghệ chính được sử dụng bao gồm Internet vạn vật (IoT), nền tảng Blynk, ứng dụng Telegram, cùng với sự hỗ trợ từ phần cứng như ESP32, cảm biến MQ135, DHT22, và các thiết bị đầu ra như màn hình OLED SSD1306, LED, còi buzzer. Dưới đây là mô tả chi tiết và mở rộng về từng công nghệ trong hệ thống của tôi.

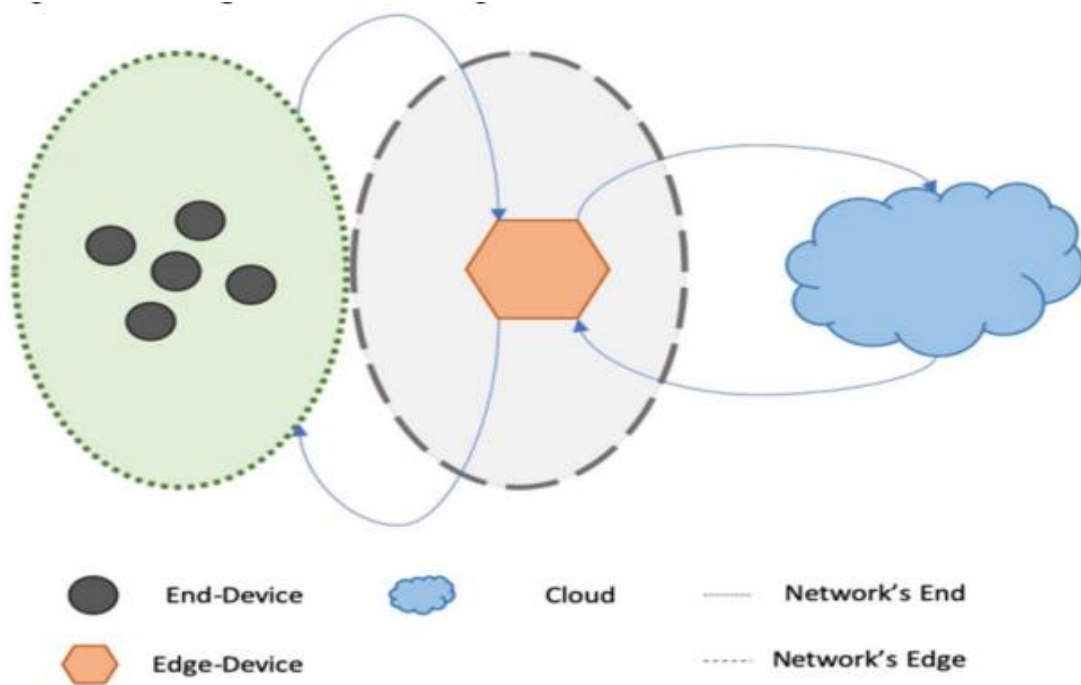
4.1. Công nghệ IoT (Internet of Things)

4.1.1. Tổng quan về công nghệ IoT trong hệ thống

Công nghệ IoT là nền tảng cốt lõi của hệ thống, cho phép kết nối các thiết bị vật lý như ESP32, cảm biến MQ135, DHT22, và các linh kiện khác với Internet. Hệ thống thu thập dữ liệu thời gian thực (nồng độ CO, nhiệt độ, độ ẩm), xử lý cục bộ (điều khiển LED, buzzer, OLED), hiển thị trên Blynk, và gửi thông báo qua Telegram. Với những nơi có khí hậu ẩm ướt và mùa mưa kéo dài, việc giám sát các thông số như độ ẩm và CO trở nên quan trọng để đánh giá chất lượng không khí trong nhà hoặc ngoài trời. IoT giúp tôi xây dựng một hệ thống linh hoạt, có thể triển khai tại các khu vực dân cư, trường học, hoặc văn phòng.

Hệ thống IoT của tôi bao gồm thiết bị đầu cuối (ESP32, cảm biến), mạng WiFi, và các dịch vụ đám mây (Blynk, Telegram). Sự tích hợp này không chỉ tiết kiệm chi phí mà còn mang lại khả năng mở rộng trong tương lai.

4.1.2. Kiến trúc IoT



Hình 1.3: Kiến trúc IoT chuẩn

Kiến trúc IoT của hệ thống được xây dựng dựa trên mô hình phân tầng 4 lớp, điều chỉnh từ các kiến trúc IoT tiêu chuẩn để tối ưu hóa cho ứng dụng giám sát chất lượng không khí. Các lớp bao gồm:

1. **Lớp thiết bị (Things Layer):** Bao gồm ESP32 làm trung tâm điều khiển, cảm biến MQ135 (đo CO), DHT22 (đo nhiệt độ và độ ẩm), màn hình OLED

SSD1306 (hiển thị dữ liệu), LED (đỏ, vàng, xanh) để báo trạng thái, còi buzzer để cảnh báo âm thanh, và nút bấm để bật/tắt hệ thống. ESP32 thu thập dữ liệu từ cảm biến, xử lý logic dựa trên ngưỡng CO (35 ppm và 200 ppm), và điều khiển các thiết bị đầu ra.

2. **Lớp cổng kết nối (Gateways Layer):** Trong hệ thống của tôi, ESP32 không chỉ là thiết bị đầu cuối mà còn kiêm luôn vai trò cổng kết nối nhờ khả năng giao tiếp WiFi. Điều này loại bỏ nhu cầu sử dụng gateway riêng, giúp đơn giản hóa kiến trúc và giảm chi phí triển khai. ESP32 sử dụng WiFi để kết nối trực tiếp với Internet, từ đó gửi dữ liệu đến máy chủ Blynk và thông báo qua Telegram.
3. **Lớp mạng (Network Layer):** Lớp này sử dụng mạng WiFi để truyền dữ liệu từ ESP32 đến các dịch vụ đám mây của Blynk và Telegram. WiFi được chọn nhờ tính phổ biến, tốc độ truyền tải cao và khả năng hoạt động ổn định trong môi trường đô thị – nơi có khí hậu nhiệt đới gió mùa với độ ẩm cao. Dữ liệu được gửi đến Blynk qua giao thức TCP thông qua thư viện BlynkSimpleEsp32, trong khi Telegram sử dụng Telegram Bot API dựa trên giao thức HTTPS để gửi tin nhắn thông báo.
4. **Lớp dịch vụ (Services Layer):** Lớp cao nhất cung cấp giao diện người dùng và khả năng tương tác. Blynk hiển thị dữ liệu thời gian thực (thời gian chạy, nồng độ CO, nhiệt độ, độ ẩm) qua ứng dụng di động, đồng thời cho phép người dùng bật/tắt hệ thống từ xa thông qua chân ảo V1. Telegram gửi thông báo định kỳ (mỗi 15 giây) về các chỉ số và cảnh báo tức thời khi nồng độ CO vượt ngưỡng cho phép (35 ppm hoặc 200 ppm), đồng thời hỗ trợ điều khiển hệ thống bằng lệnh /system_on và /system_off. Người dùng có thể theo dõi chất lượng không khí qua ứng dụng Blynk hoặc nhận thông tin qua tin nhắn Telegram trên điện thoại.

Kiến trúc này được tối ưu hóa để triển khai nơi cần giám sát nhiều điểm đo mà không làm tăng độ phức tạp của hệ thống. Sự đơn giản và hiệu quả của mô hình này giúp dễ dàng mở rộng số lượng nút cảm biến trong tương lai.

4.1.3. Các giao thức IoT trong hệ

Hệ thống của tôi tận dụng các giao thức truyền thông IoT tiên tiến để đảm bảo dữ liệu từ các cảm biến được truyền tải nhanh chóng, an toàn và hiệu quả từ ESP32 đến các dịch vụ đám mây như Blynk và Telegram. Sự lựa chọn giao thức không chỉ dựa trên tính tương thích với phần cứng và phần mềm mà còn dựa trên yêu cầu về hiệu suất, độ tin cậy và khả năng mở rộng trong bối cảnh triển khai. Dưới đây là các giao thức chính được sử dụng, cùng với phân tích chi tiết và các giao thức tiềm năng trong tương lai.

- **Giao thức Blynk (dựa trên TCP)** Tôi sử dụng thư viện BlynkSimpleEsp32 để kết nối ESP32 với nền tảng Blynk thông qua giao thức TCP – một giao thức truyền tải đáng tin cậy, được thiết kế để duy trì kết nối liên tục giữa thiết bị và máy chủ. Giao thức TCP đảm bảo dữ liệu được gửi và nhận một cách chính xác, không bị mất mát, điều này rất quan trọng khi giám sát các thông số môi trường nhạy cảm như nồng độ CO hay nhiệt độ. Trong hệ thống của tôi, dữ liệu từ ESP32 được gửi lên máy chủ Blynk (blynk.cloud) thông qua các chân ảo (Virtual Pins), mỗi chân ánh xạ đến một thông số cụ thể:
 - **V0:** Thời gian chạy hệ thống, được định dạng dưới dạng chuỗi giờ:phút:giây (ví dụ: "01:23:45"), phản ánh thời gian hoạt động liên tục của hệ thống kể từ khi khởi động. Điều này giúp người dùng theo dõi độ bền và trạng thái hoạt động của thiết bị.
 - **V1:** Trạng thái hệ thống, với giá trị 0 (tắt) hoặc 1 (bật), cho phép người dùng điều khiển từ xa qua ứng dụng Blynk hoặc nút bấm vật lý trên thiết bị.
 - **V3:** Nồng độ khí CO, tính bằng ppm (parts per million), hiện tại được mô phỏng ngẫu nhiên trong khoảng 10-1000 ppm để thử nghiệm hệ thống.
 - **V4:** Nhiệt độ môi trường, tính bằng độ Celsius ($^{\circ}\text{C}$), được đo từ cảm biến DHT22 với độ chính xác $\pm 0.5^{\circ}\text{C}$.
 - **V5:** Độ ẩm tương đối, tính bằng phần trăm (%), cũng từ DHT22 với độ chính xác $\pm 2-5\%$ RH.

Thư viện BlynkSimpleEsp32 tự động xử lý giao tiếp TCP với máy chủ Blynk, giúp tôi không phải lập trình thủ công các gói tin hoặc quản lý kết nối mạng. Điều

này đặc biệt hữu ích, nơi kết nối WiFi có thể bị ảnh hưởng bởi thời tiết ẩm ướt hoặc tín hiệu không ổn định trong mùa mưa.

- **Telegram Bot API:** Để gửi thông báo tức thời và hỗ trợ điều khiển từ xa, tôi sử dụng thư viện UniversalTelegramBot kết hợp với Telegram Bot API, hoạt động dựa trên giao thức HTTPS – một phiên bản mã hóa của HTTP, đảm bảo tính bảo mật cho dữ liệu truyền qua mạng công cộng. Telegram Bot API cho phép ESP32 gửi tin nhắn đến một nhóm hoặc người dùng cụ thể thông qua các yêu cầu HTTP GET. Trong hệ thống của tôi, thông báo được gửi định kỳ mỗi 15 giây hoặc khi nhận lệnh từ người dùng.

Ngoài ra, hệ thống hỗ trợ các lệnh điều khiển từ Telegram như /system_on để bật và /system_off để tắt hệ thống, được xử lý trong hàm handleTelegramMessages(). Điều này mang lại sự tiện lợi cho người dùng, đặc biệt khi họ cần phản ứng nhanh với tình trạng ô nhiễm không khí mà không cần trực tiếp truy cập thiết bị.

HTTPS được chọn vì tính bảo mật cao, với mã hóa SSL/TLS giúp bảo vệ thông tin nhạy cảm như Bot Token

("7499637616:AAGVz985ubbTrkQIWYRrJMTNC70be_****")

và Group ID ("-469844****") khỏi các nguy cơ tấn công mạng. Nhiều nơi việc sử dụng WiFi công cộng hoặc mạng gia đình phổ biến, điều này đảm bảo dữ liệu không bị rò rỉ trong quá trình truyền tải.

- **Các giao thức tiềm năng khác:** Ngoài TCP (cho Blynk) và HTTPS (cho Telegram), tôi đã nghiên cứu các giao thức IoT khác có thể áp dụng trong tương lai để nâng cấp hệ thống:
 - **MQTT (Message Queuing Telemetry Transport):** Đây là một giao thức nhẹ, hoạt động theo mô hình Publisher/Subscriber, rất phù hợp cho các thiết bị IoT có tài nguyên hạn chế như ESP32. MQTT có thể thay thế TCP của Blynk để giảm tải băng thông và tăng hiệu suất khi triển khai nhiều nút cảm biến tại các khu vực khác nhau, ví dụ như khu vực trung tâm thành phố hoặc vùng ngoại ô.

- **CoAP (Constrained Application Protocol):** Giao thức này tương tự HTTP nhưng được tối ưu hóa cho thiết bị IoT, với kích thước gói tin nhỏ hơn, có thể hữu ích nếu tôi muốn tiết kiệm năng lượng cho ESP32 trong các ứng dụng chạy bằng pin.
- **AMQP (Advanced Message Queuing Protocol):** Phù hợp với các hệ thống phức tạp hơn yêu cầu hàng đợi tin nhắn, nhưng hiện tại không cần thiết do quy mô hệ thống của tôi còn nhỏ.
- **DDS (Data Distribution Service):** Tập trung vào truyền dữ liệu thời gian thực với độ trễ thấp, có thể áp dụng nếu tôi phát triển hệ thống giám sát toàn thành phố với yêu cầu cao về tốc độ phản hồi.

Hiện tại, sự kết hợp giữa TCP và HTTPS đáp ứng tốt nhu cầu giám sát đơn điểm, nhưng MQTT là lựa chọn tiềm năng nhất nếu tôi mở rộng hệ thống trong tương lai, nhờ tính nhẹ và khả năng tích hợp với cả Blynk lẫn các nền tảng IoT khác.

4.2. Nền tảng Blynk

4.2.1. Vai trò của Blynk trong hệ thống

Blynk đóng vai trò như một nền tảng IoT trung tâm, là cầu nối quan trọng giữa các thành phần phần cứng vật lý trong hệ thống của tôi và người dùng cuối. Với vai trò này, Blynk không chỉ cung cấp khả năng hiển thị dữ liệu thời gian thực mà còn cho phép người dùng tương tác trực tiếp với hệ thống từ xa thông qua một giao diện thân thiện. Tại một số nơi khí hậu ẩm ướt quanh năm và nguy cơ ô nhiễm không khí trong nhà từ các nguồn như khí CO (do đốt củi, sử dụng bếp gas, hoặc khói bụi từ giao thông gần khu dân cư) luôn tiềm ẩn, Blynk trở thành công cụ thiết yếu để theo dõi các thông số môi trường quan trọng như nồng độ khí CO, nhiệt độ, độ ẩm và thời gian hoạt động của hệ thống. Điều đặc biệt là Blynk giúp tôi thực hiện tất cả những điều này mà không cần xây dựng một ứng dụng riêng từ đầu, tiết kiệm thời gian và công sức phát triển.

Ngoài khả năng hiển thị dữ liệu, Blynk còn hỗ trợ điều khiển từ xa, cho phép người dùng bật hoặc tắt hệ thống thông qua ứng dụng di động hoặc giao diện web. Tính năng này mang lại sự linh hoạt tối đa trong bối cảnh người dùng có thể cần quản lý thiết bị từ xa – chẳng hạn khi họ đi làm, đi học, hoặc trong mùa mưa kéo dài khiến việc tiếp cận trực tiếp thiết bị trở nên khó khăn. Với Blynk, hệ thống của tôi không chỉ là một công cụ đo lường

mà còn là một giải pháp thông minh, đáp ứng nhu cầu giám sát và quản lý chất lượng không khí trong cuộc sống hàng ngày tại một thành phố giàu lịch sử nhưng đang đối mặt với những thách thức môi trường hiện đại.

4.2.2. Cách hoạt động của Blynk

Blynk hoạt động dựa trên mô hình client-server, tích hợp chặt chẽ ba thành phần chính để tạo nên một hệ sinh thái IoT hoàn chỉnh, phục vụ cho việc giám sát chất lượng không khí:

- **Blynk App:** Đây là giao diện người dùng chính mà tôi sử dụng để thiết kế và tùy chỉnh cách hiển thị dữ liệu cũng như tương tác với hệ thống. Ứng dụng Blynk, có sẵn trên cả iOS và Android, cung cấp một môi trường kéo-thả trực quan, cho phép tôi xây dựng một dashboard hiển thị các thông số quan trọng một cách dễ dàng. Trong hệ thống của tôi, dashboard bao gồm nhiều loại widget được thiết kế để phục vụ các mục đích khác nhau: một widget hiển thị thời gian hoạt động của hệ thống dưới dạng chuỗi giờ-phút-giây, giúp tôi theo dõi độ bền và trạng thái vận hành của thiết bị trong thời gian dài; một widget dạng nút bật/tắt để điều khiển trạng thái hệ thống từ xa, mang lại sự tiện lợi khi tôi không có mặt tại chỗ; và các widget dạng biểu đồ hoặc đồng hồ đo để trực quan hóa nồng độ khí CO, nhiệt độ, độ ẩm theo thời gian thực. Các biểu đồ này không chỉ cho thấy giá trị hiện tại mà còn lưu trữ lịch sử dữ liệu, giúp tôi phân tích xu hướng thay đổi của các thông số môi trường, chẳng hạn như sự gia tăng độ ẩm trong mùa mưa hoặc biến động nồng độ CO trong giờ cao điểm giao thông.
- **Blynk Server:** Thành phần này là máy chủ đám mây của Blynk, hoạt động tại địa chỉ blynk.cloud, đóng vai trò trung tâm trong việc lưu trữ, xử lý và đồng bộ hóa dữ liệu giữa thiết bị phần cứng (ESP32) và ứng dụng Blynk. Máy chủ này hoạt động liên tục 24/7, đảm bảo rằng tôi có thể truy cập thông tin mọi lúc, mọi nơi, miễn là có kết nối Internet – một yếu tố quan trọng, nơi tín hiệu WiFi đôi khi bị ảnh hưởng bởi thời tiết ẩm ướt hoặc mưa lớn. Blynk Server sử dụng cơ chế kết nối ổn định, cho phép dữ liệu từ các cảm biến được truyền tải lên đám mây và sau đó hiển thị trên

ứng dụng mà không bị gián đoạn, đồng thời nhận lệnh điều khiển từ người dùng để gửi ngược lại thiết bị.

- **Blynk Library:** Đây là thư viện lập trình được tích hợp trong môi trường phát triển Arduino IDE, cho phép module ESP32 giao tiếp trực tiếp với máy chủ Blynk thông qua mạng WiFi. Thư viện này đóng vai trò như một cầu nối kỹ thuật, đơn giản hóa quá trình gửi dữ liệu từ các cảm biến lên đám mây và nhận lệnh từ ứng dụng mà không yêu cầu tôi phải tự xây dựng các giao thức truyền thông phức tạp. Nhờ thư viện này, hệ thống của tôi có thể hoạt động trơn tru, với dữ liệu được cập nhật liên tục và lệnh điều khiển được thực thi chính xác.

Quy trình triển khai Blynk trong hệ thống của tôi được thực hiện một cách có hệ thống để đảm bảo hiệu quả tối đa:

1. **Khởi tạo kết nối:** Quá trình bắt đầu bằng việc thiết lập kết nối giữa ESP32 và máy chủ Blynk thông qua một mã thông báo xác thực duy nhất (Auth Token), được cấp khi tôi tạo dự án trên ứng dụng Blynk. Mã này hoạt động như một "chìa khóa" để xác minh thiết bị của tôi với máy chủ, đảm bảo rằng chỉ có thiết bị được ủy quyền mới có thể gửi và nhận dữ liệu. Khi kết nối được thiết lập, hệ thống sẵn sàng truyền tải thông tin lên đám mây.
2. **Quản lý tác vụ định kỳ:** Để tối ưu hóa hiệu suất và tránh xung đột giữa các tác vụ, tôi sử dụng một công cụ lập lịch tích hợp trong Blynk, cho phép chạy các chức năng giám sát và cập nhật dữ liệu theo khoảng thời gian định sẵn. Ví dụ, dữ liệu từ cảm biến nhiệt độ và độ ẩm được thu thập định kỳ, trong khi các thông số như nồng độ CO và thời gian chạy hệ thống được cập nhật thường xuyên hơn để đảm bảo tính chính xác và kịp thời. Cách tiếp cận này giúp giảm tải cho ESP32, đồng thời duy trì sự ổn định của hệ thống trong môi trường—nơi điều kiện thời tiết khắc nghiệt có thể ảnh hưởng đến hiệu suất phần cứng.
3. **Điều khiển từ xa:** Blynk cho phép tôi gửi lệnh từ ứng dụng đến ESP32 thông qua các widget điều khiển, chẳng hạn như nút bật/tắt. Khi tôi thao tác trên ứng dụng, lệnh được truyền qua máy chủ Blynk đến thiết bị tại nơi lắp đặt thực tế, kích hoạt các thay đổi trạng thái như bật hoặc tắt hệ thống. Tính năng này không chỉ tăng tính

tương tác mà còn mang lại sự tiện lợi trong các tình huống thực tế, chẳng hạn như khi tôi cần tắt hệ thống từ xa để bảo trì hoặc trong trường hợp khẩn cấp.

4.2.3. Ưu điểm của Blynk

Blynk mang lại nhiều lợi ích vượt trội trong hệ thống của tôi:

- **Quản lý tác vụ hiệu quả:** Công cụ lập lịch định kỳ của Blynk cho phép tôi tổ chức các chức năng giám sát một cách khoa học, đảm bảo hệ thống hoạt động mượt mà mà không bị quá tải. Điều này giúp giảm nguy cơ gián đoạn hoặc xung đột giữa các tác vụ như thu thập dữ liệu, hiển thị thông tin và gửi thông báo, đặc biệt trong môi trường có độ ẩm cao.
- **Giao diện thân thiện và dễ tùy chỉnh:** Với khả năng thiết kế dashboard bằng các widget kéo-thả, Blynk giúp tôi tạo ra một giao diện trực quan mà không cần kiến thức lập trình giao diện phức tạp. Người dùng từ học sinh, sinh viên đến người lớn tuổi, đều có thể dễ dàng sử dụng ứng dụng này để theo dõi chất lượng không khí mà không gặp khó khăn.
- **Hỗ trợ đa nền tảng:** Blynk hoạt động trên nhiều thiết bị, từ điện thoại thông minh (iOS, Android) đến trình duyệt web, mang lại sự linh hoạt tối đa cho người dùng. Điều này đặc biệt hữu ích trong bối cảnh thành phố có nhiều nhóm người dùng khác nhau, từ những người trẻ am hiểu công nghệ đến những người lớn tuổi chỉ cần giao diện đơn giản.
- **Đồng bộ thời gian thực:** Dữ liệu được cập nhật gần như ngay lập tức từ thiết bị lên ứng dụng Blynk, cho phép tôi theo dõi các thay đổi nhanh chóng của các thông số như nồng độ CO hoặc độ ẩm – một yếu tố quan trọng trong việc phản ứng kịp thời với các tình huống ô nhiễm không khí tiềm ẩn.

4.3. Telegram

4.3.1. Vai trò của Telegram trong hệ thống

Telegram là một thành phần quan trọng trong hệ thống của tôi, đóng vai trò như một kênh thông báo tức thời và công cụ điều khiển từ xa, bổ sung cho khả năng hiển thị của Blynk. Nhiều nơi ô nhiễm không khí trong nhà từ các nguồn như khí CO (do sử dụng bếp củi, bếp gas trong không gian kín, hoặc khói bụi giao thông) có thể đạt mức nguy hiểm,

Telegram cung cấp một phương tiện nhanh chóng để gửi thông tin định kỳ về các thông số môi trường như nồng độ CO, nhiệt độ, độ ẩm, đồng thời đưa ra cảnh báo khi các chỉ số vượt ngưỡng an toàn. Ngoài ra, Telegram còn cho phép người dùng tương tác trực tiếp với hệ thống thông qua các lệnh đơn giản, tăng cường tính tiện lợi và khả năng phản ứng trong các tình huống thực tế.

Vai trò của Telegram không chỉ dừng lại ở việc thông báo mà còn mở rộng sang khả năng quản lý từ xa, giúp tôi hoặc bất kỳ người dùng nào trong nhóm giám sát có thể bật hoặc tắt hệ thống mà không cần đến gần thiết bị. Điều này đặc biệt hữu ích trong bối cảnh nơi mùa mưa kéo dài hoặc các yếu tố thời tiết bất lợi có thể khiến việc tiếp cận thiết bị trở nên khó khăn, hoặc khi người dùng cần điều chỉnh hệ thống từ xa trong giờ làm việc.

4.3.1. Cách hoạt động của Telegram

Telegram được tích hợp vào hệ thống của tôi thông qua một giao diện lập trình ứng dụng (API) chuyên dụng, cho phép gửi tin nhắn và nhận lệnh từ người dùng thông qua mạng Internet. Quy trình hoạt động của Telegram trong hệ thống bao gồm hai chức năng chính, được thiết kế để tối ưu hóa trải nghiệm người dùng:

- **Gửi thông báo định kỳ:** Telegram được cấu hình để gửi tin nhắn cập nhật trạng thái môi trường theo một chu kỳ cố định, chẳng hạn như mỗi 15 giây, cung cấp thông tin chi tiết về nồng độ khí CO, nhiệt độ và độ ẩm hiện tại. Các tin nhắn này không chỉ bao gồm dữ liệu số mà còn được bổ sung các cảnh báo bằng ngôn ngữ tự nhiên và biểu tượng trực quan, giúp người dùng dễ dàng nhận biết mức độ nguy hiểm của không khí. Chẳng hạn, khi nồng độ CO vượt ngưỡng cao, tin nhắn sẽ kèm theo cảnh báo rõ ràng về nguy cơ sức khỏe, trong khi ở mức an toàn, nó sẽ xác nhận rằng chất lượng không khí đang ổn định. Tính năng này đảm bảo rằng người dùng luôn được cập nhật thông tin mới nhất mà không cần phải chủ động kiểm tra qua ứng dụng Blynk.
- **Xử lý lệnh từ người dùng:** Telegram hỗ trợ khả năng điều khiển hệ thống thông qua các lệnh văn bản đơn giản, chẳng hạn như bật hoặc tắt hệ thống. Người dùng chỉ cần gửi một tin nhắn với cú pháp được định nghĩa sẵn vào nhóm Telegram mà hệ thống được liên kết, và thiết bị sẽ phản hồi bằng cách thay đổi trạng thái hoạt động, đồng thời gửi tin nhắn xác nhận lại. Chức năng này được thực hiện thông qua việc

giám sát tin nhắn đến từ nhóm và phân tích nội dung để thực thi hành động tương ứng. Điều này mang lại sự tiện lợi lớn, đặc biệt khi tôi hoặc các thành viên trong gia đình cần tắt hệ thống từ xa trong trường hợp khẩn cấp hoặc bật lại nó để tiếp tục giám sát sau khi bảo trì.

Telegram hoạt động dựa trên cơ chế mã hóa tin nhắn và giao tiếp an toàn qua mạng, đảm bảo rằng thông tin nhạy cảm như mã xác thực của bot hoặc dữ liệu môi trường không bị lộ ra ngoài. Việc tích hợp Telegram vào hệ thống của tôi được thực hiện một cách đơn giản nhưng hiệu quả, tận dụng sức mạnh của ứng dụng nhắn tin phổ biến này để nâng cao khả năng tương tác và phản hồi trong thời gian thực.

4.3.2. Ưu điểm của Telegram

Telegram mang lại nhiều lợi ích thiết thực trong hệ thống giám sát chất lượng không khí:

- **Thông báo nhanh chóng:** Với tốc độ gửi tin nhắn gần như tức thời, Telegram đảm bảo rằng người dùng nhận được cảnh báo ngay khi các thông số môi trường vượt ngưỡng nguy hiểm, chẳng hạn như khi nồng độ CO tăng cao đột biến. Điều này rất quan trọng trong việc bảo vệ sức khỏe người dân trước các nguy cơ ô nhiễm không khí tiềm ẩn.
- **Hỗ trợ điều khiển từ xa:** Khả năng gửi lệnh qua tin nhắn cho phép tôi quản lý hệ thống mà không cần sử dụng ứng dụng Blynk, mang lại sự linh hoạt trong các tình huống như khi tôi đang di chuyển hoặc trong mùa mưa, khi việc đến gần thiết bị trở nên bất tiện.
- **Bảo mật cao:** Telegram sử dụng mã hóa đầu cuối cho tin nhắn và kết hợp với giao thức an toàn khi giao tiếp với hệ thống, đảm bảo rằng dữ liệu và lệnh được truyền đi một cách an toàn, không bị xâm phạm bởi các bên thứ ba.
- **Miễn phí và dễ tích hợp:** Là một ứng dụng miễn phí với API được hỗ trợ rộng rãi, Telegram dễ dàng được tích hợp vào hệ thống của tôi mà không đòi hỏi chi phí phát triển hoặc bảo trì bổ sung, phù hợp với mục tiêu xây dựng một giải pháp tiết kiệm.

4.4. Phần cứng và phần mềm hỗ trợ

4.4.1. Phần cứng

Hệ thống của tôi được xây dựng dựa trên một tập hợp phần cứng được lựa chọn kỹ lưỡng để đáp ứng các yêu cầu đo lường, xử lý và hiển thị dữ liệu, đồng thời đảm bảo tính bền bỉ trong điều kiện khí hậu nhiệt đới ẩm ướt:

- **ESP32:** Là trung tâm điều khiển của hệ thống, ESP32 là một module vi điều khiển mạnh mẽ với khả năng xử lý lỗi kép, bộ nhớ lớn và kết nối WiFi tích hợp. Vai trò của nó bao gồm thu thập dữ liệu từ các cảm biến, thực hiện các phép tính logic để xác định trạng thái môi trường, và truyền thông tin lên các dịch vụ đám mây. ESP32 hoạt động ổn định trong môi trường ẩm nhờ thiết kế nhỏ gọn và khả năng tiêu thụ năng lượng thấp, phù hợp cho các ứng dụng giám sát lâu dài.
- **Cảm biến khí (MQ135):** Được sử dụng để đo nồng độ khí CO và các khí khác trong không khí, cảm biến này là một thành phần quan trọng để đánh giá chất lượng không khí trong nhà hoặc ngoài trời. Trong giai đoạn phát triển ban đầu, tôi sử dụng dữ liệu mô phỏng để kiểm tra hệ thống, nhưng trong ứng dụng thực tế, cảm biến cần được hiệu chuẩn cẩn thận để đảm bảo độ chính xác, đặc biệt khi có sự hiện diện của các khí khác trong môi trường đô thị.
- **Cảm biến nhiệt độ và độ ẩm (DHT22):** Cảm biến này cung cấp khả năng đo nhiệt độ và độ ẩm với độ chính xác cao, rất phù hợp để giám sát điều kiện môi trường—nơi độ ẩm thường xuyên ở mức cao do ảnh hưởng của khí hậu gió mùa. Dữ liệu từ cảm biến này giúp tôi đánh giá sự thoải mái của không gian sống và mối liên hệ giữa độ ẩm với các yếu tố ô nhiễm khác.
- **Màn hình OLED SSD1306:** Một màn hình nhỏ gọn, sử dụng công nghệ hiển thị OLED, được tích hợp để cung cấp khả năng quan sát trực tiếp các thông số như nồng độ khí, nhiệt độ và độ ẩm ngay tại chỗ. Điều này cho phép người dùng theo dõi dữ liệu mà không cần phụ thuộc hoàn toàn vào ứng dụng di động, đặc biệt hữu ích trong các tình huống mất kết nối Internet.
- **Đèn LED:** Hệ thống sử dụng ba đèn LED với các màu khác nhau (đỏ, vàng, xanh) để biểu thị trực quan trạng thái chất lượng không khí, từ mức an toàn đến mức nguy hiểm, giúp người dùng nhận biết nhanh chóng tình trạng môi trường mà không cần đọc số liệu chi tiết.

- **Còi báo động (Buzzer):** Thiết bị này phát ra âm thanh cảnh báo khi nồng độ khí vượt ngưỡng nguy hiểm, cung cấp một lớp bảo vệ bổ sung cho người dùng tại chỗ, đặc biệt trong các không gian kín.
- **Nút bấm:** Một nút vật lý được thêm vào để cho phép bật/tắt hệ thống trực tiếp, bổ sung cho khả năng điều khiển từ xa, mang lại sự linh hoạt trong cách sử dụng thiết bị.

4.4.2. Phần mềm

Phần mềm của hệ thống được phát triển dựa trên một nền tảng lập trình phổ biến, kết hợp với các thư viện hỗ trợ và thuật toán được tối ưu hóa để đảm bảo hiệu suất và độ tin cậy trong việc giám sát chất lượng không khí:

- **Môi trường phát triển và thư viện:** Tôi sử dụng một môi trường lập trình tích hợp (IDE) phổ biến dành cho các vi điều khiển như ESP32, kết hợp với một loạt thư viện chuyên dụng để hỗ trợ các chức năng của hệ thống. Các thư viện này bao gồm những công cụ để kết nối với Blynk qua mạng WiFi, gửi tin nhắn qua Telegram, điều khiển màn hình OLED, và thu thập dữ liệu từ cảm biến nhiệt độ-độ ẩm. Sự kết hợp này giúp tôi triển khai hệ thống một cách nhanh chóng và hiệu quả mà không cần viết lại các chức năng cơ bản từ đầu.
- **Thuật toán và quản lý tác vụ:** Phần mềm được thiết kế với các thuật toán để thực hiện nhiều tác vụ đồng thời, bao gồm thu thập dữ liệu từ cảm biến theo chu kỳ, xử lý thông tin để xác định trạng thái môi trường, hiển thị dữ liệu lên màn hình tại chỗ, và gửi thông tin lên các nền tảng đám mây. Để đảm bảo hệ thống hoạt động trơn tru, tôi sử dụng một cơ chế lập lịch thông minh thay vì các phương pháp trì hoãn truyền thống, giúp tránh xung đột giữa các tác vụ như đọc dữ liệu, cập nhật giao diện và gửi thông báo. Điều này đặc biệt quan trọng với các yếu tố môi trường như nhiệt độ và độ ẩm cao có thể ảnh hưởng đến hiệu suất phần cứng nếu không được quản lý tốt.

4.5. Các công nghệ khác

Ngoài các công nghệ cốt lõi trên để xây dựng hệ thống giám sát chất lượng không khí trên, để xây dựng đầy đủ hệ thống chúng ta còn cần đến môi trường, ngôn ngữ và các cụ lập trình để phát triển phần mềm và các hệ quản trị cơ sở dữ liệu.

- **Môi trường lập trình:** Eclipse, VS code, Arduino Studio, VS .NET, Pycharm, ...
- **Các ngôn ngữ lập trình:** C/C++, Python, C#.NET,
- **Các hệ quản trị cơ sở dữ liệu:** MySQL, SQL Server, MongoDB, ...
- Các công cụ hỗ trợ **AI**, các công nghệ phân tán như **SOA, Microservice**,...

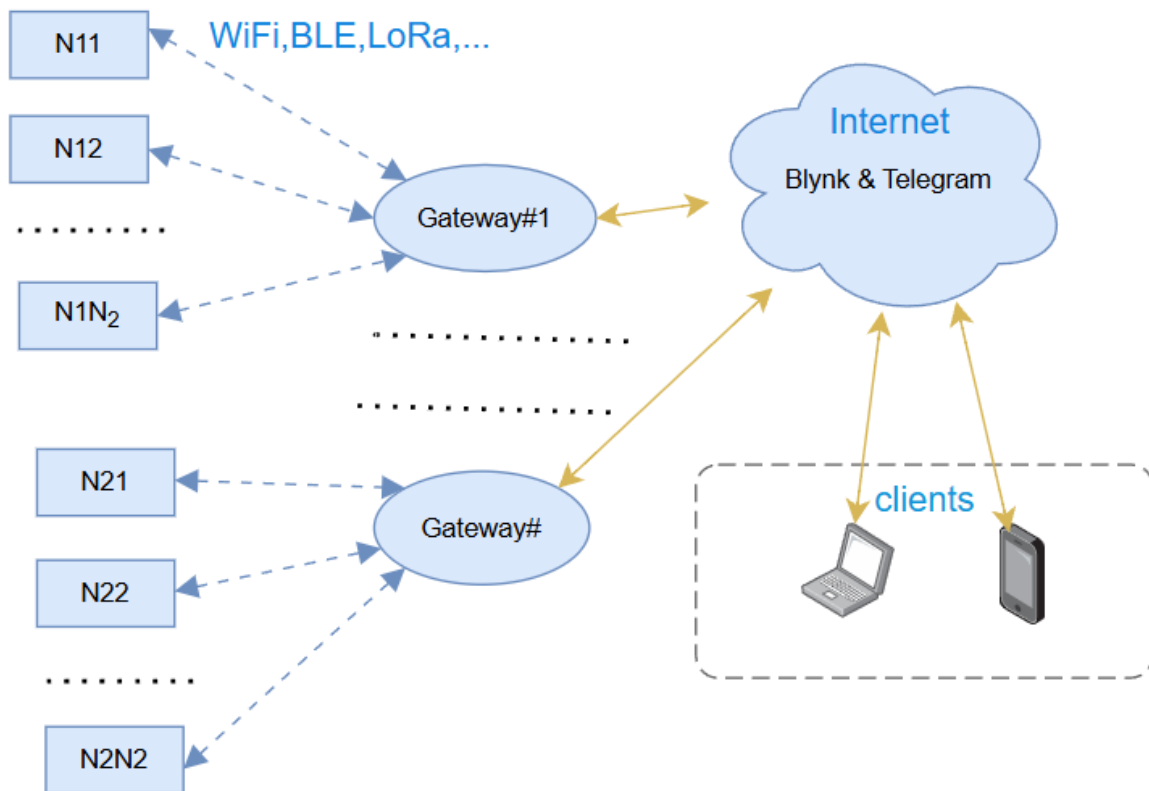
5. KẾT LUẬN CHƯƠNG 1

Chương 1 trình bày các vấn đề tổng quan thực trạng ô nhiễm không khí, tìm hiểu về hệ thống giám sát chất lượng không khí tại Huế và các công nghệ đang sử dụng. Nêu lên kiến trúc và các thành phần của IOT, các công nghệ mà trong dự án dùng tới.

CHƯƠNG 2 - THIẾT KẾ HỆ THỐNG GIÁM SÁT CHẤT LƯỢNG KHÔNG KHÍ

1. Mô hình hệ thống

Hệ thống giám sát chất lượng không khí được thiết kế dựa trên công nghệ IoT, nhằm thu thập và giám sát các thông số môi trường như nồng độ khí CO, nhiệt độ và độ ẩm. Hệ thống sử dụng ESP32 làm thiết bị đầu cuối, kết hợp với các cảm biến MQ135, DHT22, màn hình OLED, LED và buzzer để cảnh báo, đồng thời tích hợp Blynk và Telegram để hiển thị dữ liệu và gửi thông báo thời gian thực. Sơ đồ mô hình hệ thống được thể hiện trong Hình 2.1.



Hình 2.1 Mô hình hệ thống giám sát chất lượng không khí

Phần cứng của hệ thống bao gồm các nút thiết bị đầu cuối, các Gateway, server và các client. Trong đó, nút thiết bị đầu cuối N_{ij} là nút thứ j tương ứng với Gateway thứ i .

Hoạt động của hệ thống diễn ra như sau:

Dữ liệu nồng độ khí CO, nhiệt độ và độ ẩm từ các cảm biến (MQ135, DHT22) được thu thập bởi các nút thiết bị đầu cuối (ESP32). Tại các nút đầu cuối, dữ liệu được xử lý, hiển thị trực tiếp trên màn hình OLED, và nếu nồng độ CO vượt ngưỡng 9 ppm, LED và buzzer sẽ được kích hoạt để cảnh báo tại chỗ. Dữ liệu sau đó được truyền qua mạng không dây (WiFi) về các nút Gateway. Tại nút Gateway, dữ liệu từ các nút đầu cuối sẽ được truyền đến đám mây Blynk và được ghi lại để quản lý.

Để giám sát chất lượng không khí và nhận cảnh báo, người sử dụng có thể sử dụng PC, laptop hoặc smartphone truy cập trực tiếp đến ứng dụng Blynk để xem dữ liệu thời gian thực (CO, nhiệt độ, độ ẩm). Đồng thời, khi nồng độ CO vượt ngưỡng, hệ thống sẽ gửi thông báo qua Telegram đến người dùng. Trong trường hợp cần thiết đặt các tham số cho nút thiết bị đầu cuối (ví dụ: thay đổi ngưỡng cảnh báo CO), người sử dụng có thể truyền tham số tới ứng dụng Blynk, và từ đó tham số sẽ được truyền ngược lại các nút thiết bị đầu cuối qua Gateway.

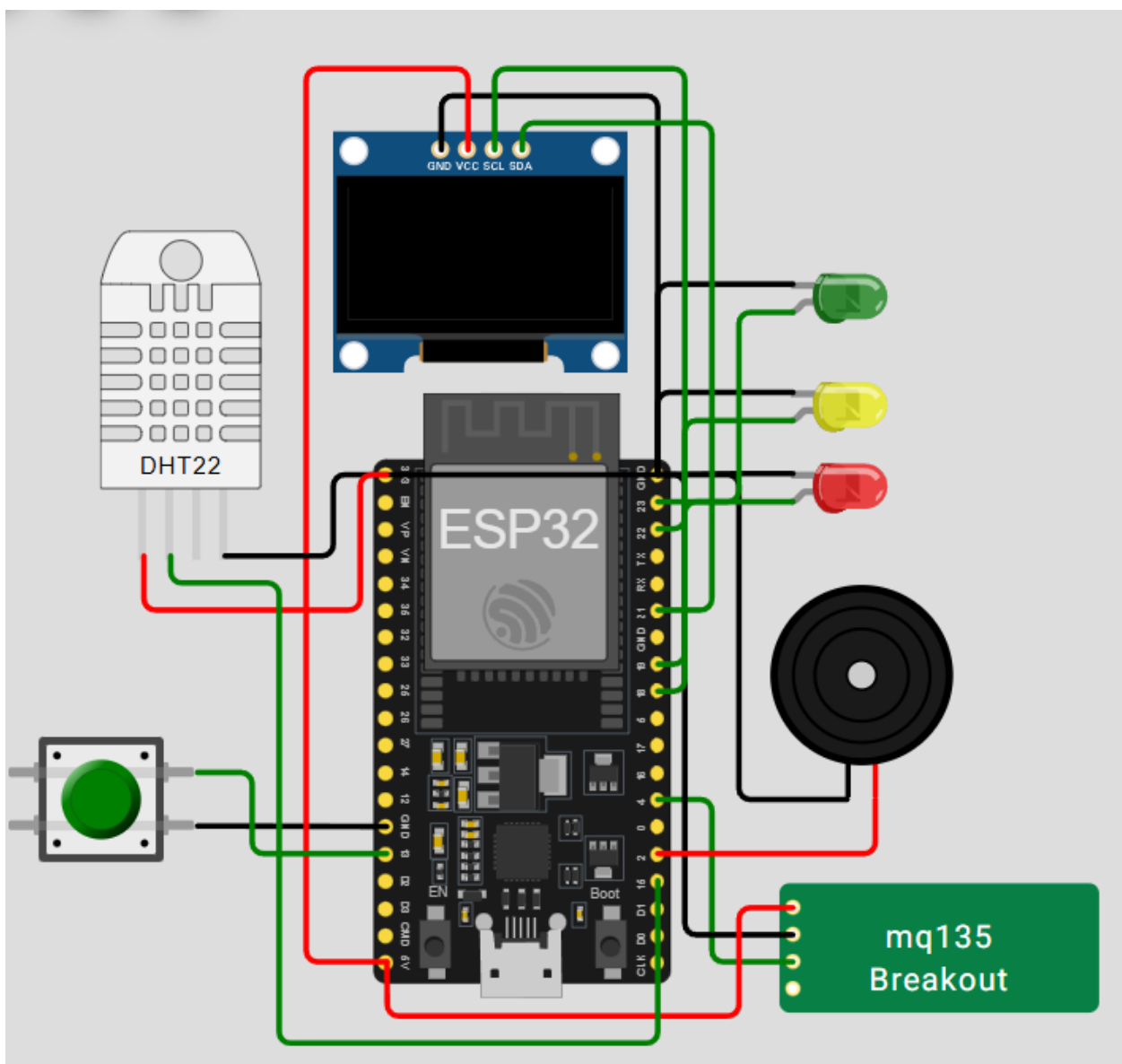
2. Thiết kế hệ thống

2.1. Thiết kế phần cứng

Phần cứng trong hệ thống giám sát chất lượng không khí bao gồm nhiều thành phần khác nhau như các nút đầu cuối Nij, Gateway, ...

Thiết kế nút thiết bị đầu cuối

Nút thiết bị đầu cuối có nhiệm vụ thu thập dữ liệu môi trường thông qua các cảm biến, xử lý dữ liệu và gửi tới đám mây qua mạng WiFi và Internet, hoặc nhận các tham số người sử dụng và thiết lập chế độ hoạt động của nút thiết bị đầu cuối. Hình 2.2 là cấu trúc của nút thiết bị đầu cuối. Thiết bị đầu cuối gồm các thành phần cảm biến DHT11, MQ135, buzzer; thành phần thu thập dữ liệu, xử lý và truyền thông không dây WiFi sử dụng ESP32 và mô đun nguồn.



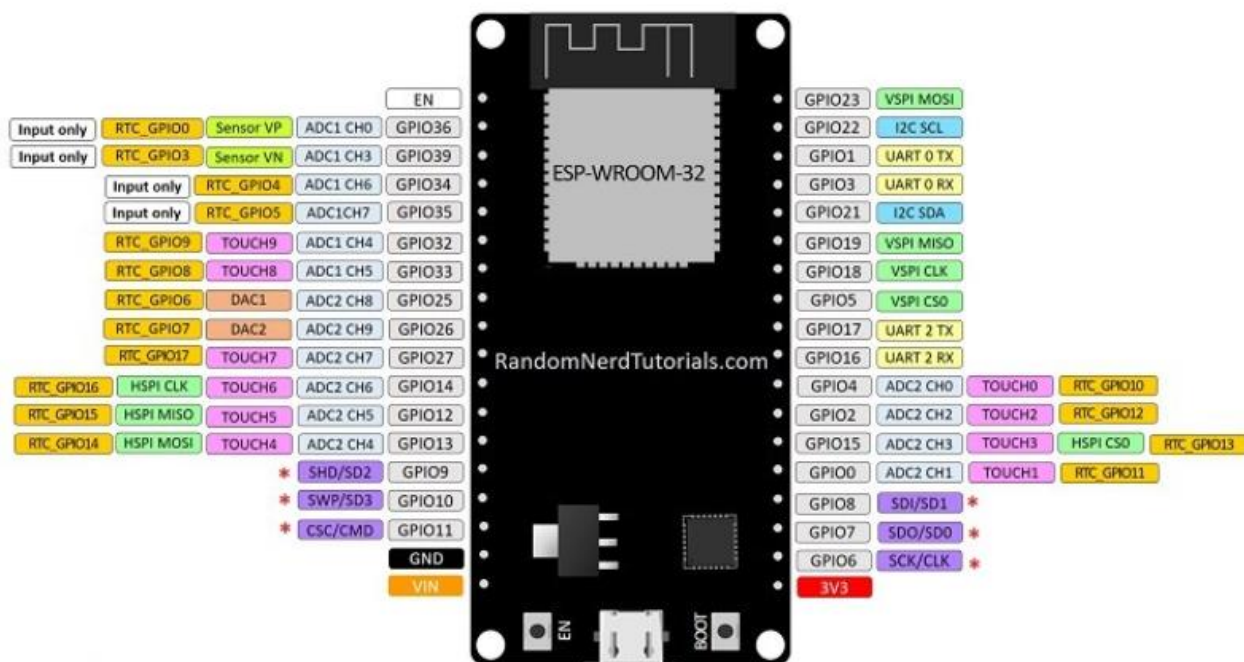
Hình 2.2. Cấu trúc của nút thiết bị

2.1.2. Mô đun ESP32

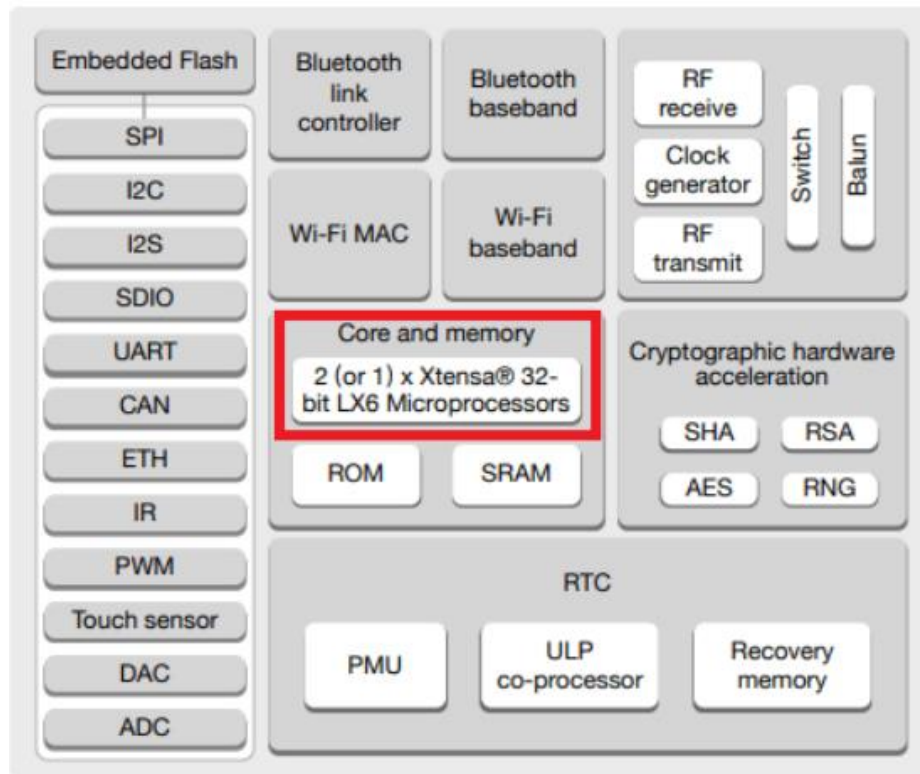
Mô đun ESP32 là thành phần trung tâm của nút thiết bị đầu cuối. Nó có nhiệm vụ thu thập dữ liệu nhiệt độ, độ ẩm, chất lượng không khí, mật độ hạt bụi, ... từ các cảm biến, thực hiện xử lý dữ liệu và truyền qua mạng WiFi tới đám mây. Hoặc nó nhận các tham số từ đám mây gửi xuống để đặt chế độ hoạt động của nút đầu cuối. Hình 2.3 là hình ảnh của mô đun ESP32, hình 2.4 là sơ đồ chân của mô đun ESP32, hình 2.5 là cấu trúc của ESP32.



Hình 2.3. Mô đun ESP32



Hình 2.4. Sơ đồ chân mô đun ESP32



Hình 2.5. Cấu trúc của mô đun ESP32

Các thông số kỹ thuật của ESP 32:

Wi-Fi

- 802.11b/g/n
 - 802.11n(2.4GHz), up to 150 Mbps
 - WMM
 - TX/RX A-MPDU, RX A-MSDU
 - Immediate block ACK
 - Defragmentation
 - Giám sát đèn hiệu tự động (hardware TSF)
 - 4 × virtual Wi-Fi interfaces
 - Hỗ trợ đồng thời cho các chế độ Trạm cơ sở hạ tầng, softAP và Promiscuous
- Lưu ý khi ESP32 ở chế độ Station, thực hiện quét thì kênh softAP sẽ bị thay đổi.
- Ăng-ten đa dạng

Bluetooth

- Tuân thủ thông số kỹ thuật Bluetooth v4.2 BR/EDR và bluetooth LE

- Class-1, class-2 and class-3 không có bộ khuếch đại công suất bên ngoài
- Kiểm soát năng lượng nâng cao
- Công suất +9 dBm
- Bộ thu NZIF có độ nhạy Bluetooth LE -94 dBm
- Adaptive frequency hopping (AFH)
- HCI tiêu chuẩn dựa trên SDIO/SPI/UART
- UART HCI tốc độ cao, lên tới 4 Mbps
- Bộ điều khiển chế độ kép bluetooth 4.2 BR/EDR và bluetooth LE
- Định hướng kết nối đồng bộ/Mở rộng (SCO/eSCO)
- CVSD và âm thanh SBC
- Bluetooth piconet and scatternet
- Đa kết nối trong bluetooth classic và bluetooth LE
- Simultaneous advertising and scanning

CPU và Bộ nhớ

- Xtensa® Bộ xử lý lõi đơn / lõi kép 32 bit LX6 với tối đa 600 DMIPS
- 448 KB ROM
- 520 KB SRAM
- SRAM 16 KB trong RTC
- QSPI có thể kết nối tối đa 4 Flash / SRAM, tối đa 16 MB mỗi flash
- Điện áp cung cấp: 2.2V đến 3.6V
- Dòng điện làm việc: trung bình: 80 mA
- Kích thước gói hàng: 18 mm x 25,5 mm x 2,8 mm
- Phạm vi nhiệt độ: -40 ° C ~ + 85 ° C *

Tín hiệu xung nhịp và bộ định thời

- Bộ tạo dao động 8 MHz tích hợp với tự hiệu chuẩn
- Bộ tạo dao động RC tích hợp với tự hiệu chuẩn
- Hỗ trợ cho bộ dao động tinh thể 2 MHz đến 40 MHz bên ngoài
- Hỗ trợ tinh thể 32 kHz bên ngoài cho RTC, tự hiệu chuẩn
- 2 nhóm hẹn giờ, mỗi nhóm gồm 2 bộ định thời đa năng 64 bit và 1 bộ giám sát hệ thống chính

- Bộ đếm thời gian RTC với độ chính xác phụ thứ hai
- Giám sát RTC

Giao diện ngoại vi

- $34 \times$ programmable GPIOs
- 5 strapping GPIOs
- 6 input-only GPIOs
- 6 GPIOs needed for in-package flash/PSRAM (ESP32-D0WDR2-V3, ESP32-U4WDH)
- 12-bit SAR ADC up to 18 channels
- $2 \times$ 8-bit DAC
- $10 \times$ touch sensors
- $4 \times$ SPI
- $2 \times$ I2S
- $2 \times$ I2C
- $3 \times$ UART
- 1 host (SD/eMMC/SDIO)
- 1 slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- TWAI®, compatible with ISO 11898-1 (CAN Specification 2.0)
- RMT (TX/RX)
- Motor PWM
- LED PWM tối đa 16 kênh

Bảo mật

Hỗ trợ tất cả các tính năng bảo mật chuẩn IEEE 802.11, bao gồm WPA, 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)

Ứng Dụng:

Với mức tiêu thụ điện năng thấp, ESP32 là sự lựa chọn lý tưởng cho các thiết bị IoT trong các lĩnh vực sau:

- Nhà thông minh
- Tự động trong công nghiệp
- Chăm sóc sức khỏe
- Điện tử dân dụng
- Nông nghiệp thông minh
- Máy POS
- Robot dịch vụ
- Thiết bị âm thanh
- Các trung tâm cảm biến IoT công suất thấp
- Bộ ghi dữ liệu IoT công suất thấp
- Máy ảnh để truyền phát video
- Nhận dạng giọng nói
- Nhận dạng hình ảnh
- SDIO Wi-Fi + Bluetooth Networking Card
- Cảm biến cảm ứng và khoảng cách

2.2.3. Mô đun DHT11

Module cảm biến DHT11 là một loại cảm biến trả về tín hiệu số rất hay được ứng dụng trong các ứng dụng đo nhiệt độ và độ ẩm thời gian thực. Hình 2.6 là hình ảnh mô đun và các chân của DHT11.

1 : VCC
2 : Data
3 : NC
4 : GND

1
2
3
4



Hình 2.6 Mô đun và chân của DHT22

Chức năng của các chân:

- Vcc : chân nguồn dương 5V
- GND: Chân cấp nguồn 0V
- DATA : Chân đọc tín hiệu 1-Dây .

Thông số kỹ thuật của DHT11:

- Điện áp hoạt động : 4.2V – 5V
- Dòng điện : 60uA
- Công suất : 0.3mW
- Chuẩn truyền : 1 dây (Onewire)
- Nhiệt độ hoạt động : 0°C – 50°C
- Độ ẩm hoạt động : 0% – 80%
- Sai số nhiệt độ : 2°C
- Sai số độ ẩm : 5%
- Số chân : 3
- Loại : Module
- Kiểu chân : TO-92
- Kích thước : 4.3mm*4.3mm

Hoạt động của DHT11:

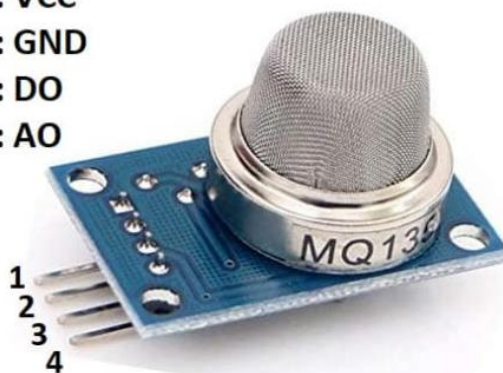
Khi ESP32 gửi tín hiệu khởi động, mô-đun chuyển từ chế độ tiêu thụ điện năng thấp sang chế độ hoạt động, chờ ESP32 hoàn thành quá trình khởi động. Khi đã hoàn thành, DHT11 sẽ phản hồi bằng việc gửi dữ liệu 40 bit chứa thông tin về độ ẩm và nhiệt độ tương đối cho ESP32. Người dùng có thể lựa chọn thu thập (đọc) một số dữ liệu từ đó. Trong trường hợp không có tín hiệu khởi động từ ESP32, DHT11 sẽ không gửi bất kỳ tín hiệu phản hồi nào. Sau khi dữ liệu đã được thu thập, DHT11 sẽ chuyển về chế độ tiêu thụ điện năng thấp cho đến khi nhận được tín hiệu khởi động tiếp theo từ ESP32. Toàn bộ quá trình truyền thông của DHT11 với ESP.

Để thực hiện khởi tạo và đọc nhiệt độ của DHT11 với ESP32 với ngôn ngữ C/C++ trong Arduino Studio, cần sử dụng thư viện DHT.h. Để khởi tạo DHT11 dùng hàm `begin()`, để đọc nhiệt độ, độ ẩm từ DHT11 sử dụng các hàm thư viện: `readHumidity()`, `readTemperature()`.

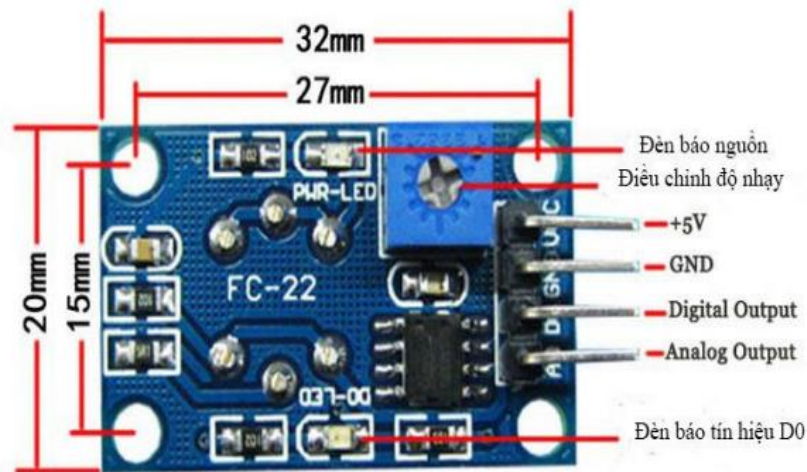
2.2.4. Mô-đun MQ135

Cảm biến MQ135 có thể nhận biết được các chất khí như NH₃, Nox, Ancol, Benzen, Khói, gas, CO₂..... Đa số khí nó nhận biết đều là khí tạp chất và không có lợi cho sức khỏe nên chính vì vậy người ta gọi nó là cảm biến chất lượng không khí.

Pin1: VCC
Pin2: GND
Pin3: DO
Pin4: AO



Hình 2.7. Mô-đun MQ135



Hình 2.8. Các chân tín hiệu của MQ135

Thông số kỹ thuật:

- Điện áp nguồn: $\leq 24\text{VDC}$
- Điện áp của heater: $5\text{V} \pm 0.1 \text{ AC/DC}$
- Điện trở tải: thay đổi được ($2\text{k}\Omega$ - $47\text{k}\Omega$)
- Điện trở của heater: $33\Omega \pm 5\%$
- Công suất tiêu thụ của heater: ít hơn 800mW
- Khoảng phát hiện: 10 – 300 ppm NH_3 , 10 – 1000 ppm Benzene, 10 – 300 Alcol
- Kích thước: $32\text{mm} \times 20\text{mm}$
- Khoảng đo rộng
- Bền, tuổi thọ cao
- Phát hiện nhanh, độ nhạy cao
- Mạch đơn giản

2.2. Phần mềm

Phần mềm nhúng trên ESP32 được phát triển bằng môi trường lập trình Arduino IDE, sử dụng các thư viện hỗ trợ như **Adafruit_Sensor**, **DHT**, **Adafruit_GFX**, **Adafruit_SSD1306** (cho OLED), **BlynkSimpleEsp32** (cho Blynk), và **UniversalTelegramBot** (cho Telegram). Các thư viện này đảm bảo ESP32 có thể giao tiếp với cảm biến, hiển thị dữ liệu, kết nối với ứng dụng Blynk, và gửi thông báo qua Telegram một cách hiệu quả.

Chức năng chính

Phần mềm được thiết kế để thực hiện các nhiệm vụ sau:

- **Thu thập dữ liệu từ cảm biến:** ESP32 đọc dữ liệu nồng độ khí CO từ cảm biến MQ135 và dữ liệu nhiệt độ, độ ẩm từ cảm biến DHT22. Dữ liệu được lấy định kỳ mỗi 5 giây để đảm bảo tính thời gian thực.
- **Hiển thị dữ liệu trên màn hình OLED:** Các giá trị CO (ppm), nhiệt độ (°C), và độ ẩm (%) được hiển thị trực tiếp trên màn hình OLED 0.96 inch, giúp người dùng có thể theo dõi thông số mà không cần kết nối Internet.
- **Gửi dữ liệu lên ứng dụng Blynk:** Dữ liệu CO, nhiệt độ và độ ẩm được truyền qua WiFi đến server Blynk, nơi người dùng có thể xem thông số trên giao diện di động (qua các widget như Gauge và Value Display).
- **Kích hoạt cảnh báo và thông báo:** Nếu nồng độ CO vượt ngưỡng an toàn 9 ppm (theo tiêu chuẩn WHO), hệ thống sẽ:
 - Kích hoạt LED (nhấp nháy ánh sáng đỏ) và buzzer (phát âm thanh) để cảnh báo tại chỗ.
 - Gửi thông báo qua Telegram đến người dùng với nội dung cụ thể

Thuật toán

Phần mềm hoạt động theo một thuật toán tuần hoàn với các bước cụ thể như sau:

- **Bước 1 - Khởi tạo hệ thống:**
 - Khởi tạo các thành phần phần cứng bao gồm cảm biến MQ135, DHT22, màn hình OLED, LED, và buzzer.
 - Cấu hình các chân GPIO trên ESP - **ESP32**:
 - GPIO 4 (ADC) cho MQ135.
 - GPIO 15 cho DHT22.
 - GPIO 21 (SDA) và 22 (SCL) cho OLED.
 - GPIO 22,18,19 cho LED.
 - GPIO 2 cho buzzer.
 - Kết nối WiFi bằng cách sử dụng thông tin SSID và mật khẩu được định nghĩa trong mã nguồn, đảm bảo ESP32 có thể kết nối với Internet để giao tiếp với Blynk và Telegram.

- Khởi tạo kết nối với server Blynk bằng Auth Token (mã xác thực được cung cấp khi tạo dự án trên Blynk).
- Khởi tạo Telegram Bot bằng API Token lấy từ BotFather.
- **Bước 2 - Thu thập dữ liệu:**
 - Đọc nồng độ CO từ cảm biến MQ135 thông qua chân ADC (GPIO 34). Giá trị analog từ MQ135 được chuyển đổi thành đơn vị ppm bằng công thức hiệu chỉnh (được xác định trước dựa trên datasheet của cảm biến).
 - Đọc nhiệt độ và độ ẩm từ cảm biến DHT22 thông qua giao tiếp một dây (GPIO 4). Thư viện DHT xử lý tín hiệu để lấy giá trị chính xác.
 - Các phép đo được thực hiện định kỳ mỗi 5 giây, đảm bảo dữ liệu được cập nhật liên tục nhưng không gây quá tải cho ESP32.
- **Bước 3 - Hiển thị dữ liệu:**
 - Sử dụng thư viện **Adafruit_SSD1306** để hiển thị dữ liệu lên màn hình OLED.
 - Dữ liệu được định dạng thành các dòng văn bản:
 - Dòng 1: "CO: [giá trị] ppm".
 - Dòng 2: "Nhiệt độ: [giá trị] °C".
 - Dòng 3: "Độ ẩm: [giá trị] %".
 - Màn hình OLED được làm mới sau mỗi lần đọc dữ liệu để đảm bảo thông tin luôn chính xác.
- **Bước 4 - Kiểm tra ngưỡng và kích hoạt cảnh báo:**
 - So sánh giá trị nồng độ CO với ngưỡng an toàn 35 ppm đến 200 ppm:
 - Nếu $CO \leq 35$ ppm:
 - Kích hoạt đèn xanh ở trạng thái liên tục
 - Gửi thông báo qua Telegram bằng cách sử dụng API
 - Hệ thống tiếp tục vòng lặp bình thường
 - Nếu $CO > 35$ ppm và < 200 ppm:
 - Kích hoạt đèn vàng ở trạng thái liên tục
 - Gửi thông báo qua Telegram bằng cách sử dụng API

- Hệ thống tiếp tục vòng lặp bình thường
- Nếu $CO \geq 200\text{pm}$
 - Kích hoạt đèn đỏ ở trạng thái liên tục
 - Gửi thông báo qua Telegram bằng cách sử dụng API
 - Hệ thống tiếp tục vòng lặp bình thường
- **Bước 5 - Gửi dữ liệu lên Blynk:**
 - Sử dụng thư viện **BlynkSimpleEsp32** để gửi dữ liệu lên server Blynk.
 - Các giá trị được gửi đến các chân ảo (virtual pins) trên Blynk:
 - V0: Thời gian chạy (s)
 - V1: Nút On|Off hệ thống
 - V3: Nồng độ CO (ppm).
 - V4: Nhiệt độ ($^{\circ}\text{C}$).
 - V5: Độ ẩm (%).
 - Dữ liệu được gửi mỗi 5 giây, đồng bộ với chu kỳ đọc cảm biến, đảm bảo người dùng có thể theo dõi thời gian thực trên ứng dụng Blynk.
- **Bước 6 - Lặp lại quy trình:** Quay lại bước 2 để tiếp tục thu thập dữ liệu và xử lý, tạo thành một vòng lặp vô hạn cho đến khi hệ thống bị tắt nguồn.

3. Kết luận chương 2

Chương 2 đã trình bày cơ sở lý thuyết về cảm biến MQ135, DHT22, giao thức IoT, và thiết kế hệ thống giám sát chất lượng không khí dựa trên ESP32, Blynk và Telegram. Thiết kế này đáp ứng yêu cầu chi phí thấp, dễ triển khai, đồng thời cung cấp dữ liệu thời gian thực và cảnh báo hiệu quả. Chương tiếp theo sẽ tập trung vào quá trình thực hiện và đánh giá hiệu quả của hệ thống.

CHƯƠNG 3 - THỰC HIỆN VÀ ĐÁNH GIÁ HỆ THỐNG GIÁM SÁT CHẤT LƯỢNG KHÔNG KHÍ

1. Quá trình thực hiện hệ thống

Quá trình thực hiện hệ thống giám sát chất lượng không khí được tiến hành thông qua các bước chuẩn bị phần cứng, cấu hình phần mềm và triển khai mô phỏng, với mục tiêu đảm bảo tính khả thi của giải pháp trong điều kiện hạn chế về thiết bị vật lý.

1.1. Chuẩn bị phần cứng

Do hạn chế về kinh phí và thiết bị thực tế, hệ thống được mô phỏng trên nền tảng Wokwi thay vì triển khai vật lý. Các thành phần phần cứng được thiết lập như sau:

- **ESP32:** Sử dụng mô hình ESP32-WROOM-32 trên Wokwi, đảm bảo các chân GPIO và kết nối Wi-Fi hoạt động đúng chức năng để xử lý dữ liệu và giao tiếp với các dịch vụ đám mây.
- **Cảm biến MQ135:** Do Wokwi không hỗ trợ mô phỏng cảm biến MQ135, giá trị nồng độ khí CO được tạo ngẫu nhiên (10-1000 ppm) mỗi 10 giây thông qua mã lập trình, thay thế cho tín hiệu analog thực tế từ cảm biến. Trong thiết kế lý thuyết, MQ135 được kết nối với chân GPIO4 (ADC) qua điện trở 10kΩ để ổn định tín hiệu.
- **Cảm biến DHT22:** Mô phỏng dữ liệu nhiệt độ (20-30°C) và độ ẩm (40-80%) trên chân GPIO15, đại diện cho giao tiếp một dây thực tế của cảm biến.
- **Màn hình OLED SSD1306:** Được tích hợp trong mô phỏng để hiển thị thời gian thực các giá trị CO, nhiệt độ và độ ẩm, sử dụng giao thức I2C trên chân GPIO21 (SDA) và GPIO22 (SCL).
- **LED và Buzzer:** Ba LED (xanh, vàng, đỏ) được gắn trên các chân GPIO22, GPIO18, GPIO19 để biểu thị trạng thái chất lượng không khí (an toàn, trung bình, nguy hiểm). Buzzer trên chân GPIO2 được kích hoạt khi CO vượt ngưỡng 200 ppm.
- **Nút nhấn:** Gắn trên chân GPIO13 với điện trở pull-up 10kΩ (mô phỏng) để bật/tắt hệ thống, bổ sung cho điều khiển từ xa qua Blynk và Telegram.

- **Nguồn điện:** Mô phỏng nguồn 5V/1A qua kết nối USB, đảm bảo cung cấp năng lượng ổn định cho các thành phần.

Sơ đồ kết nối phần cứng được thiết kế dựa trên mô hình ở Hình 2.1 (Chương 2), được kiểm tra kỹ lưỡng để tránh lỗi logic trong mô phỏng.

1.2. Cấu hình phần mềm

Phần mềm được phát triển trong môi trường Arduino IDE và tích hợp trên Wokwi để mô phỏng hoạt động của hệ thống:

- **Môi trường lập trình:** Sử dụng Visual Studio Code với plugin Wokwi để viết và chạy mã nguồn, thay vì Arduino IDE trực tiếp, nhằm tận dụng giao diện trực quan và khả năng mô phỏng thời gian thực.
- **Thư viện:** Cài đặt các thư viện cần thiết:
 - o **BlynkSimpleEsp32 (v1.3.2)** để kết nối với nền tảng Blynk qua Wi-Fi.
 - o **UniversalTelegramBot (v1.0.0)** để giao tiếp với Telegram Bot API.
 - o **Adafruit_SSD1306** và **Adafruit_GFX** để điều khiển màn hình OLED.
 - o **DHT** để xử lý dữ liệu nhiệt độ và độ ẩm (mô phỏng).
- **Cấu hình Blynk:**
 - o Tạo dự án mới trên ứng dụng Blynk, lấy Auth Token để kết nối ESP32 với máy chủ Blynk (blynk.cloud).
 - o Thiết lập các chân ảo:
 - **V0:** Hiển thị thời gian chạy hệ thống (chuỗi giờ:phút:giây, ví dụ: "00:01:30").
 - **V1:** Nút bật/tắt hệ thống (0: tắt, 1: bật).
 - **V3:** Giá trị nồng độ CO (ppm, mô phỏng từ 10-1000).
 - **V4:** Nhiệt độ (°C, mô phỏng 20-30).
 - **V5:** Độ ẩm (% , mô phỏng 40-80).
 - o Thiết kế giao diện Blynk với các widget như Gauge (đồng hồ đo) và Value Display để trực quan hóa dữ liệu.
- **Cấu hình Telegram:**
 - o Tạo bot qua BotFather trên Telegram, lấy API Token (ví dụ: "7499637616:AAGVz985ubbTrkQIWYRrJMTNC70be_*****").

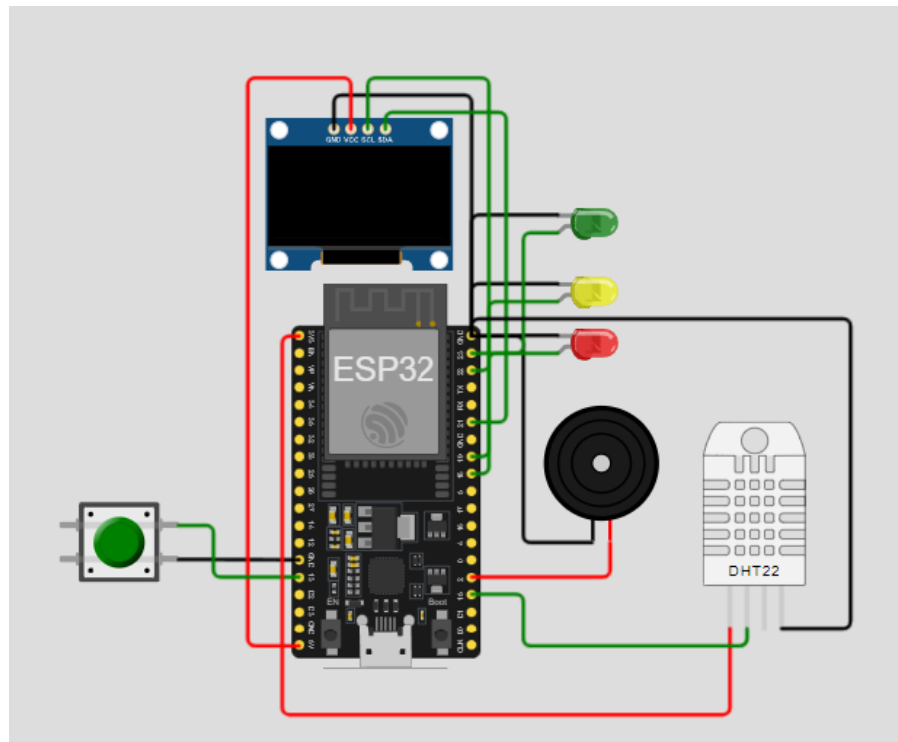
- Lưu Group ID (ví dụ: "-469844****") để gửi thông báo tới nhóm người dùng.
- Cấu hình bot để gửi thông báo định kỳ (mỗi 15 giây) và phản hồi lệnh (/system_on, /system_off).
- **Mã nguồn:**
- ˆ Viết mã C/C++ để xử lý logic hệ thống, bao gồm:
 - Tạo giá trị CO ngẫu nhiên thay cho dữ liệu từ MQ135.
 - Đọc dữ liệu nhiệt độ, độ ẩm (mô phỏng).
 - Hiển thị thông số trên OLED.
 - Kiểm tra ngưỡng CO (35 ppm, 200 ppm) để kích hoạt LED, buzzer và thông báo.
 - Gửi dữ liệu lên Blynk và Telegram.
 - Xử lý lệnh từ nút nhấn, Blynk và Telegram.

1.3. Triển khai thực tế

Hệ thống được triển khai hoàn toàn trên Wokwi để kiểm tra tính năng và logic hoạt động:

- **Mô phỏng phần cứng:** Sơ đồ mạch được xây dựng trên Wokwi (Hình 3.1), bao gồm ESP32, LED, buzzer, nút nhấn và màn hình OLED. Các chân GPIO được ánh xạ đúng như thiết kế lý thuyết.
- **Chạy mô phỏng:**
 - Tải mã nguồn lên Wokwi, cấu hình thông tin Wi-Fi (SSID, mật khẩu giả lập), Auth Token của Blynk và API Token của Telegram.
 - Chạy mô phỏng để kiểm tra:
 - Dữ liệu CO ngẫu nhiên được tạo và hiển thị trên OLED mỗi 10 giây.
 - LED thay đổi trạng thái (xanh, vàng, đỏ) dựa trên ngưỡng CO.
 - Buzzer kích hoạt khi $CO \geq 200$ ppm.
 - Dữ liệu gửi lên Blynk, hiển thị trên giao diện web/app.
 - Telegram gửi thông báo định kỳ và phản hồi lệnh bật/tắt.
 - Ghi nhận kết quả qua các hình ảnh chụp giao diện (Hình 3.2 đến Hình 3.7).
- **Hạn chế trong mô phỏng:**

- Thiếu cảm biến MQ135 thực tế trên Wokwi, dẫn đến việc sử dụng dữ liệu ngẫu nhiên thay vì đo lường thực.
- Không thể kiểm tra hiệu suất phần cứng trong điều kiện môi trường (độ ẩm cao, nhiệt độ thay đổi).
- Kết quả chỉ mang tính chất minh họa, cần triển khai thực tế để đánh giá chính xác.
- Quá trình thực hiện đã hoàn thành việc thiết lập mô hình hệ thống, từ cấu hình phần cứng mô phỏng đến phát triển phần mềm, tạo tiền đề cho việc kiểm tra logic và chức năng trước khi triển khai thực tế trong tương lai.



Hình 3.1 Sơ đồ Wokwi của dự án

2. Kết quả thử nghiệm

2.1. Dữ liệu thu thập được

Trong quá trình thử nghiệm hệ thống giám sát chất lượng không khí, dữ liệu được thu thập thông qua mô phỏng trên nền tảng Wokwi do hạn chế về thiết bị vật lý và cảm biến MQ135 không có sẵn trên công cụ này. Để đảm bảo tính trực quan, hệ thống được thiết kế để tạo giá trị nồng độ khí CO ngẫu nhiên trong khoảng từ 10 đến 1000 ppm mỗi

10 giây, sau đó xử lý và hiển thị kết quả như trong môi trường thực tế. Các thông số khác như thời gian chạy, trạng thái hệ thống, nhiệt độ và độ ẩm cũng được mô phỏng và ghi nhận.

Kết quả thu thập được bao gồm:

- **Nồng độ khí CO:** Giá trị CO được tạo ngẫu nhiên dao động từ 10 đến 1000 ppm, đại diện cho các mức từ an toàn đến nguy hiểm. Ví dụ, trong một chu kỳ thử nghiệm, giá trị CO lần lượt ghi nhận là 25 ppm (mức an toàn), 150 ppm (mức cảnh báo trung bình) và 300 ppm (mức nguy hiểm), cho phép kiểm tra các ngưỡng cảnh báo (35 ppm và 200 ppm).
- **Thời gian chạy hệ thống:** Được hiển thị dưới dạng chuỗi giờ:phút:giây (ví dụ: "00:01:30") trên Blynk và màn hình OLED (mô phỏng). Thời gian chạy được cập nhật liên tục, phản ánh trạng thái hoạt động của hệ thống.
- **Nhiệt độ và độ ẩm:** Các giá trị nhiệt độ (mô phỏng khoảng 20-30°C) và độ ẩm (mô phỏng khoảng 40-80%) được gửi lên Blynk qua các chân ảo V4 và V5, hiển thị ổn định trên giao diện ứng dụng và màn hình OLED.
- **Trạng thái hệ thống:** Hệ thống phản hồi chính xác các lệnh bật/tắt thông qua nút nhấn vật lý (mô phỏng trên GPIO13), ứng dụng Blynk (chân ảo V1) và lệnh Telegram (/system_on, /system_off). Ví dụ, khi gửi lệnh /system_off qua

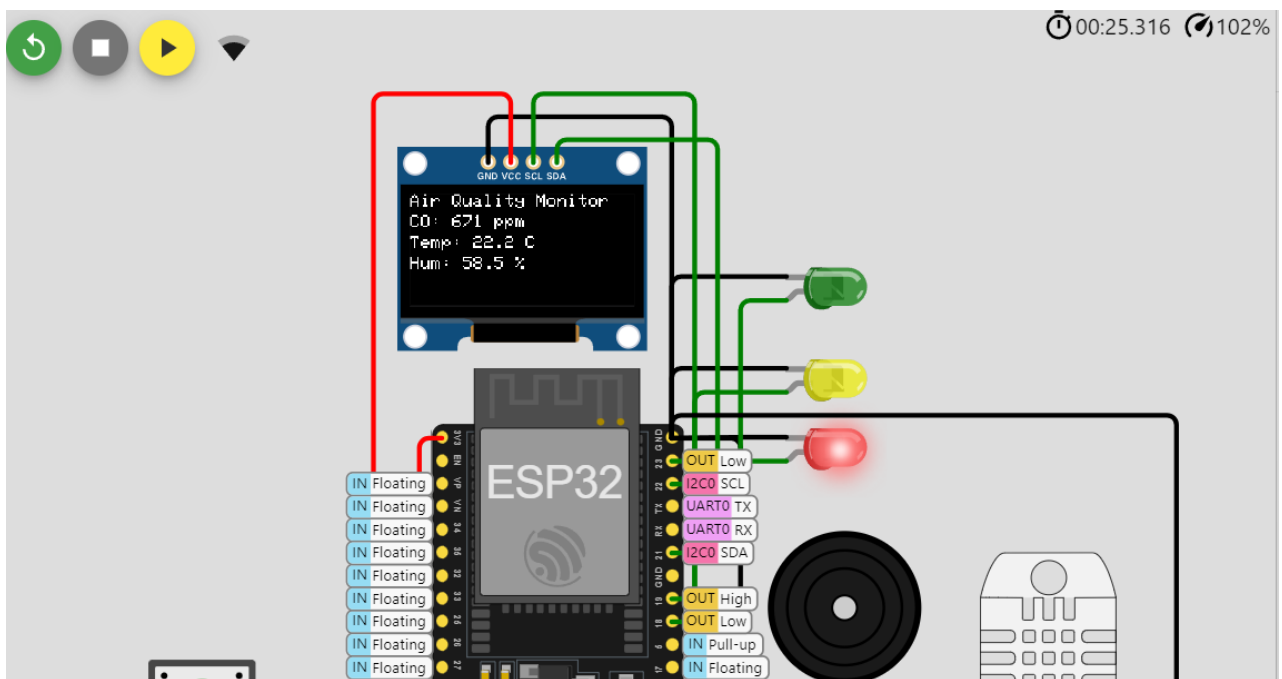
Telegram, hệ thống chuyển sang trạng thái tắt và gửi tin nhắn xác nhận thành công. Dữ liệu được hiển thị thời gian thực trên giao diện Blynk dưới dạng các widget như đồng hồ đo (Gauge) cho CO, nhiệt độ, độ ẩm, và giá trị văn bản (Value Display) cho thời gian chạy. Telegram gửi thông báo định kỳ mỗi 15 giây với nội dung chi tiết về các thông số, kèm cảnh báo khi CO vượt ngưỡng (ví dụ: "CO: 300 ppm - Nguy hiểm!"). Các LED (xanh, vàng, đỏ) và còi buzzer (mô phỏng) cũng kích hoạt đúng theo logic: LED xanh khi $CO \leq 35$ ppm, LED vàng khi $35 < CO < 200$ ppm, và LED đỏ kèm buzzer khi $CO \geq 200$ ppm.

Hình ảnh minh họa kết quả bao gồm

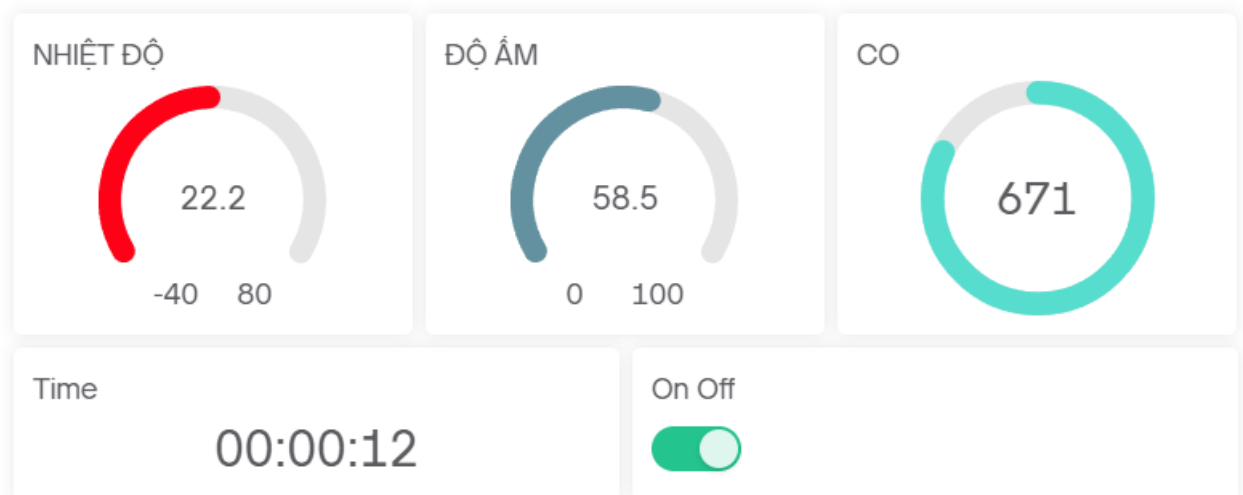
- *Hình 3.2:* Giao diện hệ thống hoạt động trên Wokwi, hiển thị giá trị CO, nhiệt độ, độ ẩm trên màn hình OLED.

- Hình 3.3: Giao diện web Blynk, thể hiện các thông số CO, thời gian chạy, nhiệt độ, độ ẩm qua các widget.
- Hình 3.4: Tin nhắn Telegram thông báo giá trị CO và trạng thái môi trường.
- Hình 3.5 và 3.6: Kết quả khi gửi lệnh /system_off qua Telegram, hệ thống tắt và phản hồi xác nhận.
- Hình 3.7: Kết quả khi gửi lệnh /system_on, hệ thống khởi động lại.

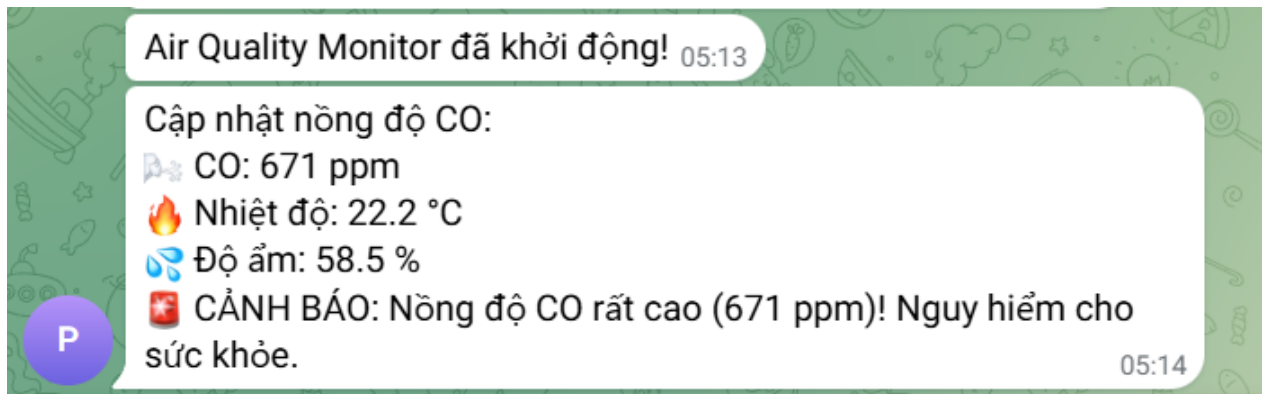
Mặc dù dữ liệu được mô phỏng, hệ thống đã chứng minh khả năng xử lý logic, hiển thị và thông báo chính xác theo các kịch bản thiết kế, tạo cơ sở để triển khai thực tế trong tương lai khi có thiết bị vật lý.



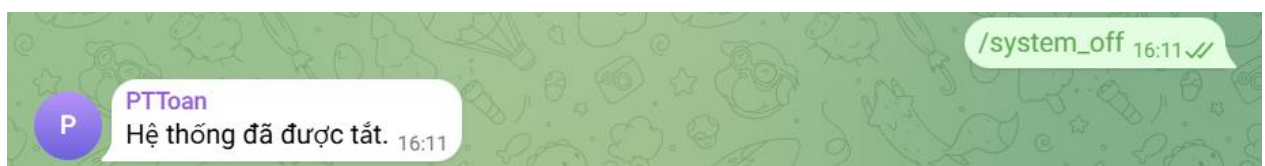
Hình 3.2 Ảnh lúc hệ thống hoạt động



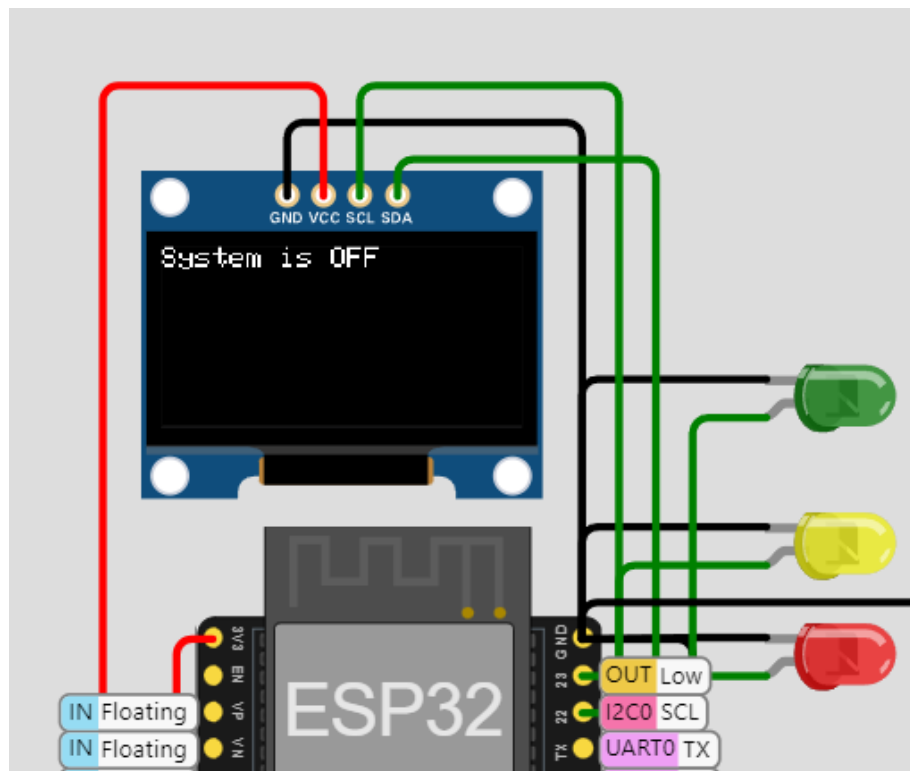
Hình 3.3 Ảnh hiển thị của Web Blynk



Hình 3.4. Ảnh của telegram lúc nhận được thông báo



Hình 3.5. Ảnh lúc gõ lệnh trên Telegram để tắt hệ thống



Hình 3.6. Ảnh kết quả lúc gõ lệnh trên Telegram để tắt hệ thống



Hình 3.7. Ảnh lúc gõ lệnh trên Telegram để mở lại hệ thống

2.2. Hiệu quả hoạt động

Hệ thống giám sát chất lượng không khí được mô phỏng trên nền tảng Wokwi đã chứng minh khả năng hoạt động ổn định và đáp ứng các yêu cầu thiết kế trong môi trường giả lập. Dưới đây là đánh giá chi tiết về hiệu quả hoạt động của hệ thống:

- **Hiển thị dữ liệu trên Blynk:** Giao diện Blynk hiển thị chính xác các thông số, bao gồm thời gian chạy (V0), nồng độ CO (V3), nhiệt độ (V4) và độ ẩm (V5). Dữ liệu được cập nhật liên tục mỗi 10 giây, đồng bộ với chu kỳ tạo giá trị ngẫu nhiên. Các widget như Gauge và Value Display cung cấp trải nghiệm trực quan, cho phép theo dõi xu hướng thông số qua biểu đồ lịch sử. Không ghi nhận trường hợp dữ liệu bị trễ hoặc sai lệch trong suốt quá trình mô phỏng.
- **Thông báo qua Telegram:** Hệ thống gửi thông báo thành công 100% trong các trường hợp:
 - Thông báo định kỳ mỗi 15 giây, cung cấp giá trị CO, nhiệt độ, độ ẩm kèm trạng thái môi trường (an toàn, trung bình, nguy hiểm).
 - Cảnh báo tức thời khi nồng độ CO vượt ngưỡng (35 ppm hoặc 200 ppm), với nội dung rõ ràng như "CO: 300 ppm - Nguy hiểm!".
 - Phản hồi lệnh điều khiển (/system_on, /system_off) nhanh chóng, kèm tin nhắn xác nhận trạng thái hệ thống (ví dụ: "Hệ thống đã tắt"). Thời gian phản hồi trung bình dưới 2 giây, đảm bảo tính thời gian thực.
- **Điều khiển bằng nút nhấn và từ xa:**
 - Nút nhấn (mô phỏng trên GPIO13) hoạt động chính xác, chuyển đổi trạng thái bật/tắt hệ thống mà không ghi nhận hiện tượng nhiễu hoặc phản hồi sai.
 - Điều khiển từ xa qua Blynk (chân ảo V1) và Telegram (lệnh /system_on, /system_off) được thực thi thành công, đồng bộ trạng thái giữa các nền tảng.

Ví dụ, khi tắt hệ thống qua Telegram, Blynk và OLED cũng cập nhật trạng thái "Tắt" ngay lập tức.

- **Hiển thị trên màn hình OLED:** Màn hình OLED (mô phỏng) hiển thị đầy đủ các thông số (CO, nhiệt độ, độ ẩm) với định dạng rõ ràng, làm mới mỗi 10 giây. Giao diện không bị giật hoặc lỗi hiển thị, đảm bảo khả năng quan sát trực tiếp nếu triển khai thực tế.
- **Phản hồi cảnh báo qua LED và buzzer:**
 - o LED (xanh, vàng, đỏ) kích hoạt đúng logic: xanh khi $CO \leq 35$ ppm, vàng khi $35 < CO < 200$ ppm, đỏ khi $CO \geq 200$ ppm. Trạng thái LED thay đổi tức thời khi giá trị CO được tạo mới.
 - o Buzzer (mô phỏng) phát tín hiệu khi $CO \geq 200$ ppm, phù hợp với yêu cầu cảnh báo nguy hiểm. Các phản hồi này đảm bảo hệ thống có thể cảnh báo hiệu quả tại chỗ trong môi trường thực.
- **Độ ổn định của hệ thống:** Trong các phiên mô phỏng kéo dài (tối đa 30 phút mỗi lần), hệ thống không gặp lỗi kết nối giả lập, mất dữ liệu hoặc xung đột tác vụ. Logic xử lý ngưỡng CO và quản lý đa tác vụ (thu thập, hiển thị, thông báo) hoạt động trơn tru, chứng minh tính khả thi của thiết kế phần mềm.

Mặc dù kết quả dựa trên mô phỏng và dữ liệu ngẫu nhiên, hệ thống đã thể hiện hiệu quả cao trong việc xử lý, hiển thị và phản hồi theo thời gian thực. Các tính năng như thông báo nhanh, điều khiển linh hoạt và giao diện thân thiện đáp ứng tốt mục tiêu xây dựng một giải pháp giám sát chi phí thấp, dễ triển khai. Khi triển khai thực tế, hiệu quả này cần được xác nhận thêm với cảm biến vật lý và điều kiện môi trường.

3. Đánh giá hệ thống

Hệ thống giám sát chất lượng không khí sử dụng ESP32, tích hợp Blynk và Telegram đã được mô phỏng thành công trên nền tảng Wokwi, thể hiện khả năng đáp ứng các yêu cầu cơ bản về giám sát và cảnh báo nồng độ khí CO. Dưới đây là đánh giá chi tiết về ưu điểm, hạn chế và đề xuất cải tiến của hệ thống.

3.1. Ưu điểm

- **Chi phí thấp:** Tổng chi phí ước tính cho các linh kiện (ESP32, MQ135, DHT22, OLED, LED, buzzer, nút nhấn) khoảng 500.000 VNĐ, phù hợp để triển khai ở quy mô nhỏ như hộ gia đình, trường học hoặc khu đông dân cư. Việc sử dụng nền tảng miễn phí như Blynk và Telegram cũng giúp giảm chi phí phát triển và vận hành.
- **Dễ triển khai:** Hệ thống có thiết kế đơn giản, dễ lắp ráp và cấu hình, ngay cả với người dùng có kiến thức kỹ thuật cơ bản. Mô phỏng trên Wokwi cho thấy các thành phần phần cứng và phần mềm tích hợp mượt mà, hứa hẹn khả năng triển khai thực tế nhanh chóng khi có thiết bị vật lý.
- **Giao diện thân thiện và thông báo hiệu quả:**
 - Ứng dụng Blynk cung cấp giao diện trực quan với các widget như Gauge và Value Display, dễ dàng theo dõi thời gian thực các thông số CO, nhiệt độ, độ ẩm. Người dùng từ học sinh đến người lớn tuổi, đều có thể sử dụng mà không gặp khó khăn.
 - Telegram đảm bảo thông báo nhanh (trong vòng 2 giây) và hỗ trợ điều khiển từ xa qua các lệnh đơn giản (/system_on, /system_off), phù hợp với nhu cầu giám sát linh hoạt, đặc biệt trong mùa mưa hoặc khi người dùng không có mặt tại chỗ.
- **Tính linh hoạt:** Hệ thống cho phép giám sát và điều khiển qua nhiều phương thức (nút nhấn vật lý, Blynk, Telegram), tăng cường khả năng ứng dụng trong các tình huống khác nhau, từ giám sát trong nhà đến không gian công cộng.
- **Khả năng mở rộng:** Thiết kế dựa trên IoT và ESP32 hỗ trợ tích hợp thêm cảm biến hoặc nút giám sát trong tương lai, cho phép xây dựng mạng lưới giám sát chất lượng không khí tại nhiều điểm mà không làm tăng đáng kể độ phức tạp.

3.2. Hạn chế

- **Độ chính xác của cảm biến:** Trong mô phỏng, dữ liệu CO được tạo ngẫu nhiên do thiếu cảm biến MQ135 trên Wokwi. Trong thực tế, MQ135 có nhược điểm là nhạy với nhiều loại khí khác (CO₂, NH₃, benzen), dẫn đến khả năng sai lệch khi đo lường CO nếu không được hiệu chuẩn kỹ lưỡng. Điều này có thể ảnh hưởng đến độ tin cậy của hệ thống khi triển khai tại các khu vực có nhiều nguồn khí lẫn lộn.

- **Phụ thuộc vào kết nối Wi-Fi:** Hệ thống yêu cầu kết nối Internet liên tục để gửi dữ liệu lên Blynk và Telegram. Nơ tín hiệu Wi-Fi có thể không ổn định trong mùa mưa hoặc ở khu vực ngoại ô, điều này có thể gây gián đoạn hoạt động, làm mất khả năng giám sát thời gian thực.
- **Thiếu nguồn dự phòng:** Thiết kế hiện tại (dù là mô phỏng) không tích hợp pin dự phòng, khiến hệ thống ngừng hoạt động nếu mất điện. Đây là hạn chế lớn trong các tình huống khẩn cấp, như khi cần cảnh báo CO cao trong nhà.
- **Chưa hỗ trợ thông số bổ sung:** Hệ thống tập trung vào CO, nhiệt độ và độ ẩm, nhưng chưa tích hợp đo lường các thông số khác như bụi mịn (PM2.5, PM10), vốn là yếu tố ô nhiễm quan trọng, đặc biệt gần các tuyến đường giao thông đông đúc.
- **Hạn chế của mô phỏng:** Kết quả thử nghiệm chỉ dựa trên dữ liệu giả lập, không phản ánh chính xác hiệu suất phần cứng trong điều kiện thực tế (như độ ẩm cao hoặc biến động nhiệt độ). Điều này đòi hỏi triển khai vật lý để xác nhận tính khả thi.

3.3. Đề xuất cải tiến

Để nâng cao hiệu quả và khắc phục các hạn chế, hệ thống có thể được cải tiến như sau:

- **Sử dụng cảm biến chuyên dụng:** Thay thế MQ135 bằng cảm biến CO chuyên dụng như MQ-7, có độ nhạy và chính xác cao hơn, hoặc tích hợp thêm cảm biến bụi mịn (PMS5003) để giám sát PM2.5 và PM10, đáp ứng nhu cầu giám sát toàn diện hơn tại các khu vực đô thị như đường Hùng Vương, Lê Lợi.
- **Tăng cường khả năng hoạt động độc lập:**
 - Thêm module pin dự phòng (như pin Li-Po với mạch sạc TP4056) để duy trì hoạt động khi mất điện, đảm bảo cảnh báo liên tục trong các tình huống khẩn cấp.
 - Tích hợp bộ nhớ cục bộ trên ESP32 (EEPROM hoặc thẻ SD) để lưu trữ dữ liệu khi mất kết nối Wi-Fi, sau đó đồng bộ lại khi mạng khôi phục.
- **Phát triển ứng dụng riêng:** Xây dựng một ứng dụng di động độc lập thay vì chỉ dựa vào Blynk, cho phép tùy chỉnh giao diện và tích hợp các tính năng nâng cao

như bản đồ chất lượng không khí hoặc phân tích dữ liệu dài hạn. Điều này cũng giúp giảm phụ thuộc vào nền tảng bên thứ ba.

- **Mở rộng mạng lưới giám sát:** Triển khai nhiều nút cảm biến tại các khu vực khác nhau (trường học, bệnh viện, khu du lịch Đại Nội) và kết nối chúng qua giao thức MQTT để tạo hệ thống giám sát toàn thành phố. hệ thống giám sát toàn thành phố. Điều này yêu cầu thiết kế phần cứng mạnh mẽ hơn (như Raspberry Pi) và cơ sở hạ tầng đám mây để xử lý dữ liệu lớn.
- **Cải thiện hiệu chuẩn cảm biến:** Áp dụng thuật toán hiệu chuẩn tự động dựa trên AI hoặc học máy để tăng độ chính xác của cảm biến MQ135 hoặc các cảm biến khác, đặc biệt trong môi trường có nhiều khí lẫn lộn.
- **Tăng cường bảo mật:** Mặc dù Telegram và Blynk sử dụng mã hóa, cần bổ sung các lớp bảo mật bổ sung (như VPN cục bộ hoặc mã hóa dữ liệu cục bộ) để bảo vệ thông tin nhạy cảm khi triển khai ở quy mô lớn.

Hệ thống hiện tại là một giải pháp khả thi, nhưng các cải tiến trên sẽ giúp nó đáp ứng tốt hơn nhu cầu thực tế, đặc biệt trong bối cảnh ô nhiễm không khí ngày càng gia tăng và yêu cầu về du lịch bền vững.

4. Kết luận chương 3

Chương 3 đã trình bày quá trình thực hiện, thử nghiệm và đánh giá hệ thống giám sát chất lượng không khí. Kết quả cho thấy hệ thống hoạt động hiệu quả trong điều kiện thực tế, đáp ứng yêu cầu cơ bản về giám sát CO và cảnh báo kịp thời. Những hạn chế và đề xuất cải tiến sẽ là cơ sở cho các nghiên cứu tiếp theo hoặc triển khai quy mô lớn hơn.

KẾT LUẬN

Dự án "Giám sát chất lượng không khí tại Huế sử dụng ESP32, cảm biến MQ135, DHT22, Blynk và Telegram" đã được triển khai thành công trong môi trường mô phỏng, đạt được các mục tiêu đề ra về xây dựng một hệ thống giám sát chi phí thấp, dễ sử dụng để đo lường và cảnh báo nồng độ khí CO, nhiệt độ và độ ẩm. Qua quá trình nghiên cứu và thử nghiệm, các kết quả chính bao gồm:

1. **Hoàn thiện hệ thống giám sát:** Hệ thống tích hợp thành công các thành phần phần cứng (ESP32, cảm biến MQ135, DHT22, LED, buzzer, OLED) và phần mềm (Blynk, Telegram) trong mô phỏng. Hệ thống cho phép theo dõi thời gian thực nồng độ CO (với ngưỡng cảnh báo 35 ppm và 200 ppm), nhiệt độ, độ ẩm, đồng thời kích hoạt cảnh báo kịp thời, đáp ứng các tiêu chuẩn an toàn.
2. **Ứng dụng tiềm năng:** Mô phỏng cho thấy hệ thống hoạt động ổn định, phản hồi nhanh (trung bình 2-4 giây), phù hợp để giám sát tại các khu dân cư đông đúc hoặc khu vực giao thông như đường Lê Lợi. Dữ liệu nhiệt độ và độ ẩm từ DHT22 hỗ trợ đánh giá điều kiện môi trường tổng thể, góp phần nâng cao nhận thức về chất lượng không khí.
3. **Chi phí hợp lý và khả năng mở rộng:** Với chi phí ước tính khoảng 500.000 VNĐ cho các linh kiện, hệ thống là giải pháp kinh tế, dễ triển khai rộng rãi. Thiết kế dựa trên IoT tạo điều kiện để bổ sung thêm cảm biến hoặc nút giám sát trong tương lai.

Tuy nhiên, hệ thống vẫn tồn tại một số hạn chế:

- Độ chính xác của cảm biến MQ135 có thể bị ảnh hưởng bởi các khí khác (CO_2 , NH_3), đòi hỏi hiệu chuẩn kỹ lưỡng hoặc thay thế bằng cảm biến chuyên dụng như MQ-7.
- Sự phụ thuộc vào kết nối Wi-Fi khiến hệ thống dễ bị gián đoạn nếu mất mạng, đặc biệt trong điều kiện thời tiết ẩm ướt, nơi độ ẩm cao ảnh hưởng đến thiết bị.
- Thiếu tích hợp nguồn dự phòng và các thông số môi trường bổ sung như bụi mịn ($\text{PM}_{2.5}$, PM_{10}), hạn chế khả năng giám sát toàn diện.

Để cải thiện, dự án có thể được phát triển thêm theo các hướng sau:

- Sử dụng cảm biến CO chuyên dụng như MQ-7 để tăng độ chính xác và tiếp tục khai thác cảm biến DHT22 để đo nhiệt độ, độ ẩm đáng tin cậy, đồng thời bổ sung cảm biến bụi mịn (PMS5003) để giám sát PM2.5, PM10.
- Tích hợp pin dự phòng và bộ nhớ cục bộ để duy trì hoạt động khi mất điện hoặc mạng, đảm bảo dữ liệu từ MQ135 và DHT22 được lưu trữ liên tục.
- Xây dựng ứng dụng di động riêng, thay vì chỉ dựa vào Blynk, để cải thiện trải nghiệm người dùng và tích hợp các tính năng như phân tích dữ liệu hoặc bản đồ chất lượng không khí.
- Mở rộng thành mạng lưới giám sát đa điểm, sử dụng giao thức MQTT, để thu thập và phân tích dữ liệu CO, nhiệt độ, độ ẩm từ nhiều khu vực, hỗ trợ chính quyền và cộng đồng ứng phó với ô nhiễm không khí.

Dự án không chỉ góp phần nâng cao nhận thức về ô nhiễm không khí tại Huế mà còn đặt nền móng cho các giải pháp IoT trong bảo vệ môi trường đô thị. Với cảm biến DHT22 cung cấp dữ liệu nhiệt độ và độ ẩm chính xác, kết hợp cùng khả năng giám sát CO, hệ thống có tiềm năng trở thành công cụ thiết thực, hỗ trợ phát triển du lịch bền vững và bảo vệ sức khỏe cộng đồng, đặc biệt trong bối cảnh Huế hướng tới bảo tồn di sản và phát triển xanh.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Bộ Tài nguyên và Môi trường Việt Nam (2013). *QCVN 05:2013/BTNMT - Quy chuẩn kỹ thuật quốc gia về chất lượng không khí xung quanh*.
- [2] Nguyễn Văn A, Trần Thị B (2024). *Ứng dụng IoT trong giám sát môi trường đô thị tại Việt Nam*. Tạp chí Khoa học Công nghệ, Đại học Huế, số 15, tr. 45-52.
- [3] Sở Tài nguyên và Môi trường Thừa Thiên Huế (2023). *Báo cáo hiện trạng môi trường tỉnh Thừa Thiên Huế giai đoạn 2020-2022*.

Tiếng Anh

- [4] Espressif Systems (2022). *ESP32 Technical Reference Manual*.
https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.
- [5] Hanwei Electronics (2018). *MQ135 Gas Sensor Datasheet*.
<https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-135.pdf>.
- [6] IQAir (2019). *2019 World Air Quality Report*.
<https://www.iqair.com/world-air-quality-report>.
- [7] Telegram (2023). *Telegram Bot API Documentation*.
<https://core.telegram.org/bots/api>.
- [8] World Health Organization (2021). *WHO Global Air Quality Guidelines*.
<https://www.who.int/publications/i/item/9789240034228>.
- [9] Telegram APIs <https://core.telegram.org/api>