

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  
PHÂN HIỆU TẠI TP.HCM  
KHOA ĐIỆN-ĐIỆN TỬ



**BÁO CÁO TỔNG KẾT  
IOT VÀ ỨNG DỤNG**

**GIÁM SÁT NHIỆT ĐỘ, ĐỘ ÂM VÀ ĐIỀU KHIỂN BẬT TẮT  
QUẠT QUA RELAY THÔNG QUA BLYNK IOT SỬ DỤNG  
ESP8266 VÀ CẢM BIẾN NHIỆT ĐỘ DHT11**

Giảng viên hướng dẫn: **VÕ THIỆN LĨNH**

Sinh viên thực hiện: **VÕ VĂN TUẤN**

MSSV: **6251020094**

Lớp: **CQ.62.KTĐTBT**

Khóa: **62**

**TPHCM, tháng 1 năm 2025**

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  
PHÂN HIỆU TẠI TP.HCM  
KHOA ĐIỆN-ĐIỆN TỬ



**BÁO CÁO TỔNG KẾT  
IOT VÀ ỨNG DỤNG  
GIÁM SÁT NHIỆT ĐỘ, ĐỘ ÂM VÀ ĐIỀU KHIỂN BẬT TẮT  
QUẠT QUA RELAY THÔNG QUA BLYNK IOT SỬ DỤNG  
ESP8266 VÀ CẢM BIẾN NHIỆT ĐỘ DHT11**

Giảng viên hướng dẫn: **VÕ THỊỆN LĨNH**

Sinh viên thực hiện: **VÕ VĂN TUẤN**

MSSV: **6251020094**

Lớp: **CQ.62.KTĐTVT**

Khóa: **62**

**TPHCM, tháng 1 năm 2025**

## LỜI CẢM ƠN

Em xin gửi lời tri ân chân thành đến thầy Võ Thiện Linh, người đã tận tâm hướng dẫn và truyền đạt cho chúng em những kiến thức vô cùng quý báu về IoT. Không chỉ dừng lại ở lý thuyết, thầy đã tạo ra một môi trường học tập sáng tạo, khuyến khích chúng em tự do khám phá, thực hành và áp dụng kiến thức vào thực tế. Chính nhờ vào sự nhiệt tình và tâm huyết của thầy mà chúng em có được nền tảng vững chắc để tự tin hoàn thành bài báo cáo này.

Sự đồng hành và hỗ trợ của thầy trong suốt quá trình học tập là nguồn cảm hứng to lớn, không chỉ giúp chúng em hiểu sâu hơn về lĩnh vực IoT mà còn tiếp thêm niềm đam mê và động lực để chúng em định hướng rõ ràng cho con đường phía trước.

Em kính chúc thầy luôn mạnh khỏe, hạnh phúc và tiếp tục đạt nhiều thành tựu trong sự nghiệp giảng dạy và nghiên cứu.

## LỜI MỞ ĐẦU

Trong thời đại công nghệ 4.0, Internet of Things (IoT) đang mở ra những bước tiến mới trong lĩnh vực tự động hóa và giám sát từ xa. Với khả năng kết nối và kiểm soát thiết bị dễ dàng, IoT hiện diện trong nhiều lĩnh vực quan trọng như quản lý môi trường, nông nghiệp thông minh, và nhà thông minh. Nhờ vào IoT, việc kiểm soát và theo dõi thiết bị không còn giới hạn trong không gian vật lý, góp phần nâng cao chất lượng cuộc sống và tối ưu hóa hiệu suất làm việc.

Xuất phát từ nhu cầu thực tế và ứng dụng tiềm năng của IoT, em tham khảo và triển khai đề tài “ Mạch giám sát nhiệt độ, độ ẩm và điều khiển quay qua relay thông qua Blynk Iot dùng ESP8266 và cảm biến DHT11 ” . Đề tài hướng đến xây dựng một hệ thống giám sát và điều khiển thông minh, giúp người dùng dễ dàng điều chỉnh các thiết bị điện từ xa qua relay và theo dõi nhiệt độ, độ ẩm trực quan qua ứng dụng Blynk IoT. Với việc sử dụng vi điều khiển ESP8266 và cảm biến DHT11, hệ thống có khả năng đo đặc các chỉ số môi trường chính xác và hiển thị tức thời trên ứng dụng di động, mang đến trải nghiệm tiện lợi và linh hoạt.

Báo cáo này sẽ trình bày chi tiết từ lý thuyết nền tảng đến quá trình thiết kế phần cứng và phần mềm, cùng các kết quả thử nghiệm và đánh giá hiệu quả của hệ thống. Em hy vọng rằng giải pháp này sẽ đóng góp thiết thực vào nhu cầu giám sát và điều khiển từ xa trong đời sống, đồng thời mở ra hướng đi mới trong việc ứng dụng IoT vào thực tế.

## MỤC LỤC

<b>PHẦN 1: TỔNG QUAN LÝ THUYẾT .....</b>	<b>1</b>
1.1. LÝ DO CHỌN ĐỀ TÀI .....	1
1.2. MỤC ĐÍCH THIẾT KẾ .....	1
1.3. PHẠM VI VÀ PHƯƠNG PHÁP THIẾT KẾ .....	2
1.3.1. Phạm vi thiết kế .....	2
1.3.2. Phương pháp thiết kế .....	2
1.4. NHIỆM VỤ THIẾT KẾ .....	3
1.4.1. Phân tích và chọn lựa linh kiện .....	3
1.4.2. Thiết lập và tích hợp phần cứng .....	3
1.4.3. Phát triển phần mềm điều khiển .....	3
1.4.4. Thủ nghiệm hệ thống và cải thiện hiệu năng .....	3
1.4.5. Đánh giá hệ thống .....	3
1.5. GIỚI THIỆU VỀ IOT VÀ ỨNG DỤNG ĐIỀU KHIỂN TỪ XA .....	3
1.6. GIỚI THIỆU VỀ VI ĐIỀU KHIỂN ESP8266 .....	3
1.6.1. Cấu tạo và chức năng .....	4
1.6.2. Nguyên lý hoạt động và Điện áp của ESP8266 .....	6
1.7. MODULE RELAY .....	6
1.7.1. Cấu tạo .....	6
1.7.1. Nguyên lý hoạt động .....	7
1.7.3. Điện áp cung cấp .....	8
1.8. CẢM BIẾN NHIỆT ĐỘ- ĐỘ ẨM DHT11 .....	8
1.8.1. Cấu tạo .....	8
1.8.2. Nguyên lý hoạt động .....	9
1.9. QUẠT THÔNG GIÓ .....	10
<b>PHẦN 2: THIẾT KẾ PHẦN CỨNG .....</b>	<b>10</b>
2.1. CÁC LINH KIỆN ĐƯỢC SỬ DỤNG .....	10
2.2. SƠ ĐỒ ĐẦU NỐI PHẦN CỨNG .....	10
2.3. NGUYÊN LÝ HOẠT ĐỘNG .....	12
<b>PHẦN 3: THIẾT KẾ PHẦN MỀM .....</b>	<b>14</b>
3.1.CÁC THƯ VIỆN SỬ DỤNG .....	14
3.2. MÔ TẢ CẤU TRÚC CHƯƠNG TRÌNH .....	14
3.3. CODE CHI TIẾT TRÊN ARDUINO IDE .....	15
ỨNG DỤNG .....	23

## DANH MỤC HÌNH ẢNH

HÌNH 1.7.1. Module Relay 1 CHANNEL .....	7
HÌNH 1.8.1. Cảm biến nhiệt độ-độ ẩm DHT11 .....	9
HÌNH 1.9.3 Quạt thông gió .....	10
HÌNH 2.2 Sơ đồ nguyên lý a .....	11
HÌNH 2.3 Sơ đồ nguyên lý b .....	12
Hình ảnh mô phỏng 3D của mạch .....	21
Hình ảnh kết nối để gia công mạch in .....	22
Hình ảnh thực tế sau khi gia công mạch .....	22

## **PHẦN 1: TỔNG QUAN LÝ THUYẾT**

### **1.1. LÝ DO CHỌN ĐỀ TÀI**

Trước xu hướng phát triển mạnh mẽ của công nghệ, nhu cầu về các hệ thống giám sát và điều khiển từ xa trở nên ngày càng phổ biến, đóng vai trò quan trọng trong việc tối ưu hóa năng lượng và cải thiện môi trường sống. Việc tích hợp các thiết bị điều khiển như relay cùng với cảm biến môi trường trong hệ thống IoT không chỉ tạo ra sự tiện nghi mà còn tăng cường khả năng tự động hóa hiệu quả. Sử dụng vi điều khiển ESP8266 với khả năng kết nối Wi-Fi, chúng ta có thể đơn giản hóa quá trình truyền dữ liệu không dây một cách hiệu quả và tiết kiệm.

Với mục đích khám phá và ứng dụng IoT vào thực tế, em đã tham khảo và chọn đề tài “ Mạch giám sát nhiệt độ, độ ẩm và điều khiển quạt qua relay thông qua Blynk IoT dùng ESP8266 và cảm biến DHT11 ”. Hệ thống này cho phép người dùng điều khiển quạt từ xa thông qua relay, đồng thời giám sát nhiệt độ và độ ẩm trực tiếp qua ứng dụng Blynk IoT. Với nền tảng này, các thông số môi trường sẽ được hiển thị thời gian thực trên điện thoại, tạo ra sự linh hoạt và tiện ích trong việc điều chỉnh không gian sống.

Báo cáo sẽ trình bày các nội dung chi tiết, từ lý thuyết nền tảng đến thiết kế phần cứng và phần mềm, cùng với các kết quả thử nghiệm và đánh giá hiệu quả của hệ thống. Chúng em kỳ vọng rằng đề tài này sẽ không chỉ mang lại một giải pháp hữu ích cho việc giám sát và điều khiển từ xa, mà còn mở rộng hơn nữa khả năng ứng dụng của IoT trong đời sống hiện đại.

### **1.2. MỤC ĐÍCH THIẾT KẾ**

Mục tiêu của đề tài này là phát triển một hệ thống IoT sử dụng vi điều khiển ESP8266, với khả năng:

- Điều khiển từ xa quạt thông qua relay, cho phép người dùng dễ dàng bật/tắt thiết bị từ xa qua kết nối Wi-Fi tích hợp trên ESP8266.

Giám sát nhiệt độ và độ ẩm môi trường thực tế theo thời gian thực nhờ cảm biến DHT11, giúp cập nhật các thông số một cách nhanh chóng và chính xác.

- Hiển thị dữ liệu nhiệt độ và độ ẩm trên nền tảng Blynk IoT, mang lại giao diện

trực quan trên thiết bị di động để người dùng có thể dễ dàng theo dõi và điều chỉnh các thiết bị khi cần.

- -> Sự linh hoạt của ESP8266, hệ thống này có thể ứng dụng trong nhiều trường hợp thực tế như quản lý môi trường trong nhà, giám sát nhiệt độ tại các phòng máy chủ, hoặc trong các khu vực cần duy trì độ ẩm ổn định để bảo quản thiết bị và hàng hóa.

### **1.3. PHẠM VI VÀ PHƯƠNG PHÁP THIẾT KẾ**

#### **1.3.1. Phạm vi thiết kế**

Đề tài tập trung vào việc ứng dụng ESP8266 để điều khiển và giám sát nhiệt độ, độ ẩm từ xa qua Blynk IoT. Hệ thống sử dụng cảm biến DHT11 để đo nhiệt độ và độ ẩm, kết hợp với relay để điều khiển quạt. Thiết kế này nhấn mạnh vào tính đơn giản và hiệu quả, giới hạn trong phạm vi điều khiển cơ bản và giám sát dữ liệu qua ứng dụng di động.

#### **1.3.2. Phương pháp thiết kế**

Nghiên cứu lý thuyết: Tìm hiểu nguyên lý hoạt động của ESP8266, cảm biến DHT11, relay và giao thức giao tiếp không dây của Blynk IoT để đảm bảo hiểu sâu về cách thức kết nối và điều khiển.

- Thiết kế phần cứng: Lựa chọn các thành phần như ESP8266, DHT11, Relay 1 channel và Quạt thông gió .
- Thiết kế phần mềm: Sử dụng Arduino IDE để lập trình ESP8266 với các chức năng điều khiển relay và gửi dữ liệu nhiệt độ, độ ẩm qua Blynk IoT, tạo giao diện giám sát trên ứng dụng di động.

Thử nghiệm và phân tích: Tiến hành thử nghiệm toàn bộ hệ thống để đảm bảo các thành phần phối hợp ổn định, hiệu quả và đáp ứng yêu cầu về giám sát, điều khiển thiết bị từ xa.

## **1.4. NHIỆM VỤ THIẾT KẾ**

### **1.4.1. Phân tích và chọn lựa linh kiện**

Lựa chọn các thành phần bắt buộc như ESP8266, relay, cảm biến DHT11 và những thành phần khác như testboard, dây nối, thiết bị ngoại vi, ... nhằm đảm bảo tính tương thích cao và khả năng hoạt động ổn định của hệ thống.

### **1.4.2. Thiết lập và tích hợp phần cứng**

Tiến hành xây dựng và hoàn thiện mạch kết nối các thành phần trên với ESP8266, đảm bảo truyền tín hiệu chính xác và cung cấp nguồn điện ổn định cho từng linh kiện.

### **1.4.3. Phát triển phần mềm điều khiển**

Thiết kế chương trình điều khiển hoạt động relay và xử lý dữ liệu nhiệt độ - độ ẩm từ cảm biến DHT11, hiển thị thông tin qua Blynk Iot bằng ngôn ngữ C/C++ thông qua nền tảng Arduino IDE.

### **1.4.4. Thử nghiệm hệ thống và cải thiện hiệu năng**

Kiểm tra khả năng điều khiển relay, giám sát nhiệt độ và độ ẩm, cũng như hiển thị thông tin thực tế trên Blynk Iot đồng thời tối ưu hóa mã nguồn để nâng cao độ chính xác và hiệu suất của hệ thống.

### **1.4.5. Đánh giá hệ thống**

Đánh giá kết quả thực nghiệm của hệ thống, so sánh với các mục tiêu thiết kế đã đề ra để xác định mức độ hiệu quả và độ tin cậy của giải pháp giám sát và điều khiển từ xa này.

## **1.5. GIỚI THIỆU VỀ IOT VÀ ÚNG DỤNG ĐIỀU KHIỂN TỪ XA**

Internet of Things (IoT) là một hệ thống kết nối thông minh, giúp các thiết bị có thể giao tiếp với nhau qua internet, cho phép chúng thu thập, xử lý và chia sẻ dữ liệu.

IoT đang phát triển mạnh mẽ và ứng dụng trong nhiều lĩnh vực như nhà thông minh, quản lý năng lượng, y tế, và giám sát môi trường,... Trong nghiên cứu này, IoT đóng vai trò cốt lõi để điều khiển và giám sát thiết bị từ xa thông qua vi điều khiển ESP8266 – một module Wi-Fi thông dụng và dễ tích hợp, phù hợp cho nhiều dự án IoT nhờ khả năng kết nối không dây ổn định.

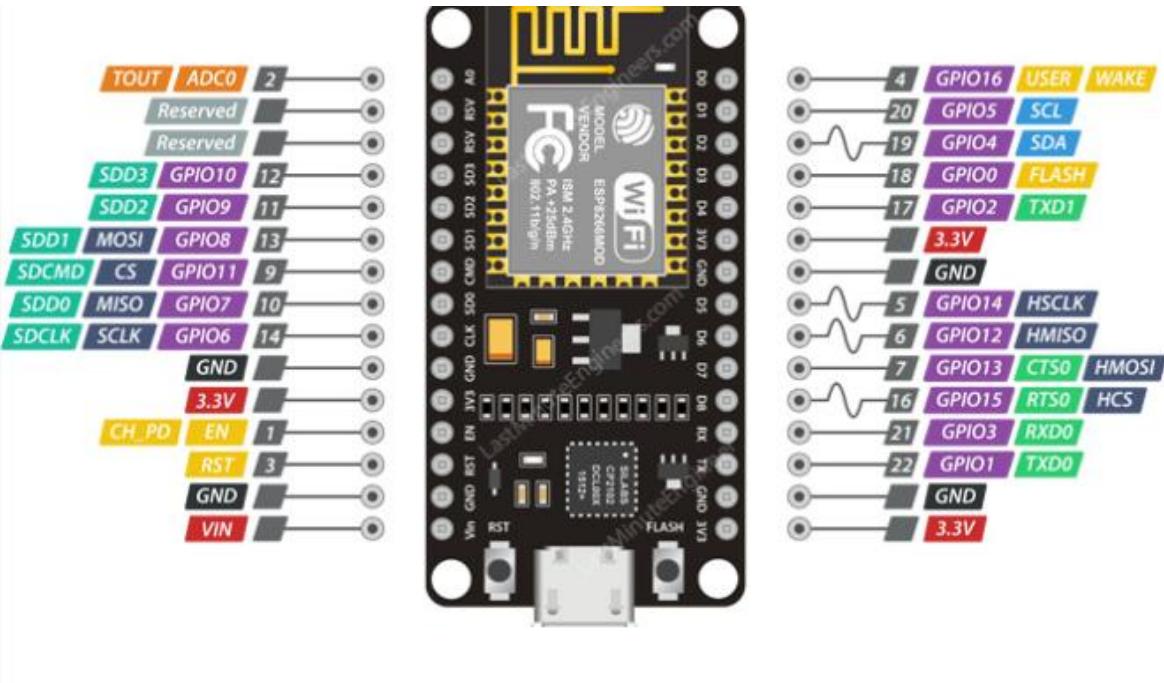
## **1.6. GIỚI THIỆU VỀ VI ĐIỀU KHIỂN ESP8266**

### **1.6.1. Cấu tạo và chức năng**

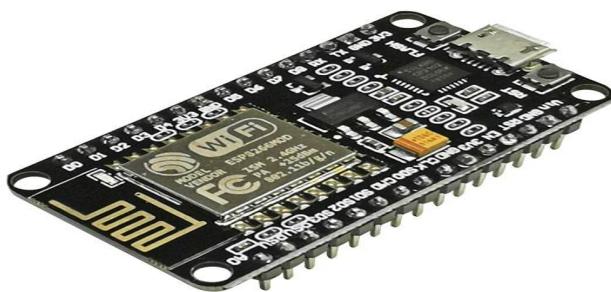
ESP8266 là một vi điều khiển tích hợp Wi-Fi với hiệu suất cao, thiết kế nhỏ gọn và giàu tính năng, gồm các thành phần chính:

- Bộ vi xử lý Tensilica LX106 32-bit: Được trang bị CPU mạnh mẽ với tần số xung nhịp tối đa 160 MHz (thông thường hoạt động ở 80 MHz), cho khả năng xử lý nhanh chóng, hỗ trợ các tác vụ phức tạp và phản hồi tức thì.
- RAM: Tích hợp 50 KB bộ nhớ RAM, đủ cho việc xử lý dữ liệu tạm thời và hỗ trợ các tác vụ trong thời gian thực.
- Bộ nhớ Flash: Dung lượng Flash từ 512 KB đến 4 MB (tuỳ phiên bản), cho phép lưu trữ chương trình và dữ liệu điều khiển ổn định lâu dài.
- GPIO và các giao thức kết nối đa dạng: ESP8266 hỗ trợ nhiều giao thức như GPIO, UART, I2C, SPI, PWM và ADC, cho phép kết nối linh hoạt với các thiết bị ngoại vi và mở rộng khả năng ứng dụng IoT.

Với những tính năng nổi bật, ESP8266 không chỉ đảm bảo khả năng xử lý và điều khiển vượt trội mà còn mang lại sự tiện lợi và hiệu quả cho các dự án điều khiển từ xa và tự động hóa trong thời đại IoT.



HÌNH 1.6.1.1. Sơ đồ chân kết nối ESP8266



HÌNH 1.6.1.2. Vi điều khiển ESP8266

### **1.6.2. Nguyên lý hoạt động và Điện áp của ESP8266**

*ESP8266 là một module Wi-Fi có vi xử lý 32-bit tích hợp, giúp nó thực hiện các nhiệm vụ IoT dễ dàng và hiệu quả.*

*Nguyên lý hoạt động của ESP8266 bao gồm:*

- Kết nối Wi-Fi: Hỗ trợ chuẩn 802.11 b/g/n với các chế độ kết nối như Access Point (AP), Station (STA) hoặc cả hai, cho phép module kết nối trực tiếp với mạng hoặc tạo mạng riêng.
- Giao tiếp đa dạng: ESP8266 có các cổng giao tiếp như UART, SPI, và I2C, giúp kết nối và điều khiển các cảm biến hoặc thiết bị ngoại vi khác.
- Giao thức TCP/IP và HTTP: Module có sẵn ngăn xếp TCP/IP và hỗ trợ API HTTP, giúp truyền tải dữ liệu qua mạng và giao tiếp với các server IoT.
- Quản lý năng lượng: ESP8266 có chế độ Light Sleep và Deep Sleep để tiết kiệm pin khi không hoạt động.
- Hệ điều hành thời gian thực (RTOS): Hỗ trợ đa nhiệm, cho phép thực hiện nhiều tác vụ mạng và điều khiển cùng lúc, tối ưu hóa cho các ứng dụng IoT.

*Điện áp hoạt động:*

ESP8266 hoạt động ở điện áp 3.3V. Để đảm bảo module hoạt động ổn định và tránh hỏng hóc:

- Điện áp đầu vào: Cung cấp nguồn ổn định 3.3V. Không nên dùng điện áp 5V trực tiếp vì có thể làm hỏng module.
- Dòng điện yêu cầu: ESP8266 cần dòng điện khoảng 70-200mA, với các đỉnh lên đến 300mA khi kết nối Wi-Fi.
- Khi kết nối với các vi điều khiển hoạt động ở 5V, cần dùng mạch chuyển đổi mức điện áp để đảm bảo an toàn cho ESP8266.

## **1.7. MODULE RELAY 1 CHANNEL**

### **1.7.1. Cấu tạo**

Module relay là một bảng mạch có các linh kiện giúp dễ dàng kết nối và điều khiển relay với vi điều khiển như ESP8266. Các thành phần chính bao gồm:

- Relay: Thường có một hoặc nhiều relay (1, 2, 4 hoặc 8 kênh), mỗi relay có cuộn dây và tiếp điểm.
- Transistor: Được dùng như một bộ khuếch đại để kích hoạt relay, vì vi điều khiển không cung cấp đủ dòng điện.
- Diode bảo vệ: Ngăn dòng điện ngược khi relay ngắt, bảo vệ vi điều khiển khỏi các xung đột biến.
- Optocoupler (nếu có): Cách ly giữa tín hiệu điều khiển từ vi điều khiển và mạch công suất của relay, giúp tăng độ an toàn cho hệ thống.
- Điện trở và LED: LED báo trạng thái giúp hiển thị rõ ràng khi relay đang bật hoặc tắt.



HÌNH 1.7.1. Module Relay 1 channel

### 1.7.1. Nguyên lý hoạt động

Module relay hoạt động như sau:

- Tín hiệu điều khiển: Khi ESP8266 gửi tín hiệu điều khiển (HIGH hoặc LOW) đến transistor qua chân IN của module, transistor sẽ dẫn điện và cung cấp dòng qua cuộn dây relay.
- Kích hoạt relay: Khi có dòng qua cuộn dây, relay sẽ chuyển từ trạng thái mở sang đóng, kết nối tải với nguồn.
- Chuyển mạch: Khi tín hiệu điều khiển ngừng (thường là khi chân IN trở về trạng thái LOW), relay sẽ trở lại trạng thái ban đầu, ngắt kết nối tải.

### 1.7.3. Điện áp cung cấp

- **Điện áp cung cấp cho module:**

- Thông thường, module relay được thiết kế để hoạt động với nguồn DC 5V hoặc 12V (tùy loại module).
- Điện áp này cấp cho cả cuộn dây relay và mạch điều khiển.

- **Điện áp điều khiển:**

- Điện áp tín hiệu điều khiển (Input) thường là 3.3V từ ESP8266

- **Điện áp và dòng tải tối đa:**

- Relay thường hỗ trợ dòng tải từ 5A đến 10A, điện áp tải tối đa là 250V AC hoặc 30V DC (tùy thuộc vào loại relay được sử dụng).

-> Module relay giúp đơn giản hóa việc điều khiển các thiết bị tải lớn, đảm bảo ESP8266 có thể điều khiển được thiết bị một cách an toàn và ổn định.

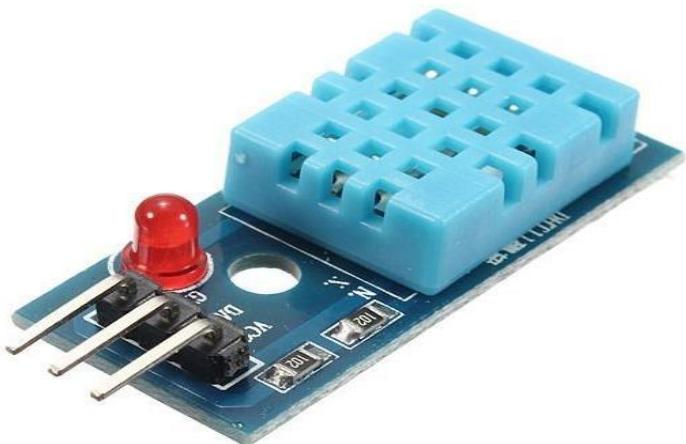
## 1.8. CẢM BIẾN NHIỆT ĐỘ- ĐỘ ẨM DHT11

### 1.8.1. Cấu tạo

Cảm biến DHT11 có các phần:

- Sensor nhiệt độ: Là một điện trở nhiệt, thay đổi giá trị khi nhiệt độ thay đổi.
- Sensor độ ẩm: Dùng chất liệu polymer với đặc tính thay đổi điện dung theo độ ẩm, cảm nhận sự thay đổi của hơi nước trong không khí.
- Bộ chuyển đổi A/D (Analog to Digital): Chuyển đổi tín hiệu analog của sensor thành tín hiệu số dễ đọc và truyền tới vi điều khiển.
- Vị điều khiển tích hợp: Điều khiển việc đọc và gửi tín hiệu số qua giao thức một

Số chân	Tên chân	Mô tả	dây. Sor đồ châ n <b>DH T11</b>
1	Vcc	Nguồn 3.3V	
2	Data	Đầu ra cả nhiệt độ và độ ẩm thông qua dữ liệu nối tiếp	
3	Ground	Nối đất	



HÌNH 1.8.1. Cảm biến nhiệt độ-độ ẩm DHT11

### 1.8.2. Nguyên lý hoạt động

Cảm biến nhiệt độ: Đo nhiệt độ thông qua thay đổi điện trở của một phần tử nhiệt bên trong, khi nhiệt độ môi trường thay đổi.

Cảm biến độ ẩm: Đo độ ẩm bằng cách theo dõi sự thay đổi điện dung của một vật liệu hấp thụ hơi nước bên trong cảm biến.

Xử lý và chuyển đổi dữ liệu: Bộ vi điều khiển tích hợp trong DHT11 sẽ thu thập các giá trị nhiệt độ và độ ẩm, chuyển đổi chúng thành tín hiệu số.

Gửi dữ liệu ra ngoài: Tín hiệu số chứa giá trị nhiệt độ và độ ẩm được truyền qua chân Data đến vi điều khiển bên ngoài để xử lý và hiển thị.

### 1.8.3. Điện áp cung cấp

- Điện áp hoạt động: 3.3V đến 5.5V.
- Dòng tiêu thụ: 0.5 mA khi hoạt động và dưới 0.2 mA ở chế độ nghỉ.

### **1.9. QUẠT THÔNG GIÓ**

- Quạt mini dùng để mô phỏng hệ thống làm mát. Nối chân dương của quạt với chân relay trên module và chân âm nối đất. Relay sẽ điều khiển bật/tắt quạt.



**HÌNH 1.9.3. Quạt**

- Quạt thông gió có 2 chân âm (-) và dương (+):
  - o Chân âm (-) nối GND.
  - o Chân dương nối với Relay\_(NC).

## **PHẦN 2: THIẾT KẾ PHẦN CỨNG**

### **2.1. CÁC LINH KIỆN ĐƯỢC SỬ DỤNG**

- Vi điều khiển ESP8266
- Cảm biến nhiệt độ-độ ẩm DHT11
- Module Relay 5V 1 channel kích hoạt mức cao/thấp
- Quạt 5V

### **2.2. SƠ ĐỒ ĐẦU NỐI PHẦN CỨNG**

Relay\_( COM) -----→ 3V3

Relay\_(NC) -----→ Quat\_(+)

Quat\_(-) -----→ GND

Relay(DC+) -----→3V3

Relay(DC-) -----→GND

Relay\_IN -----→D2\_(GPIO4)

DHT11\_+(+) -----→3V3

DHT11\_(-) -----→GND

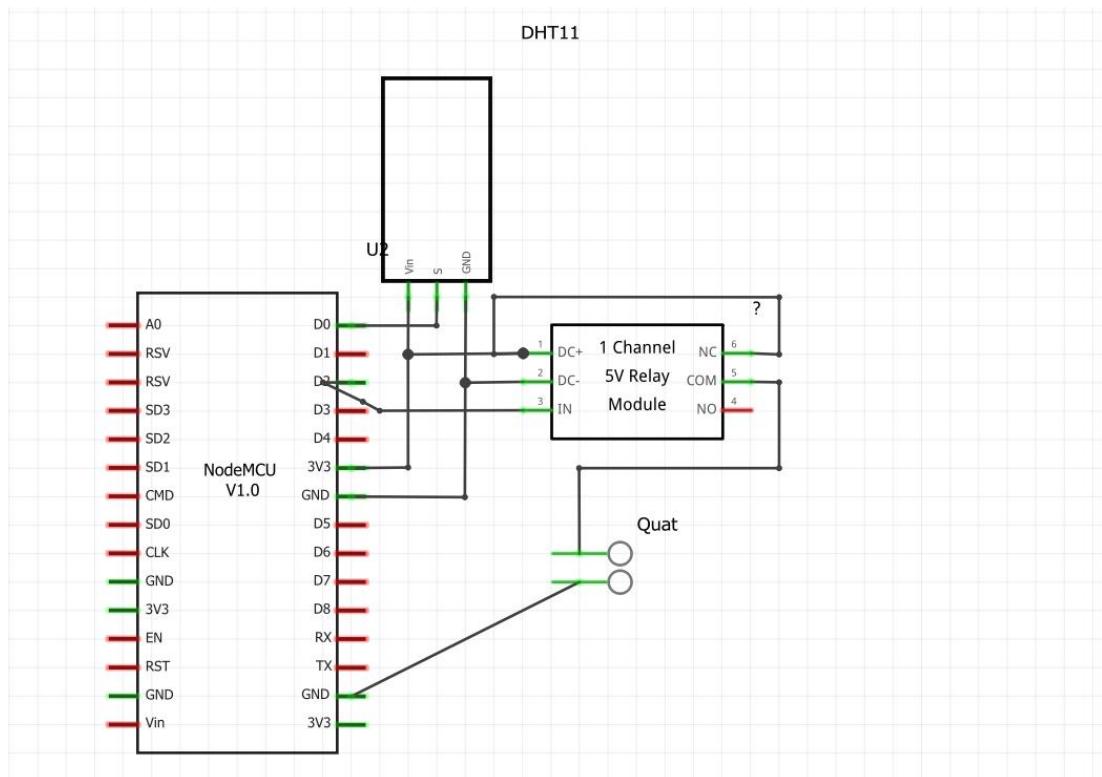
DHT11\_(out) ---→D0\_(GPIO 16)

( Khai báo chân trên Blynk ):

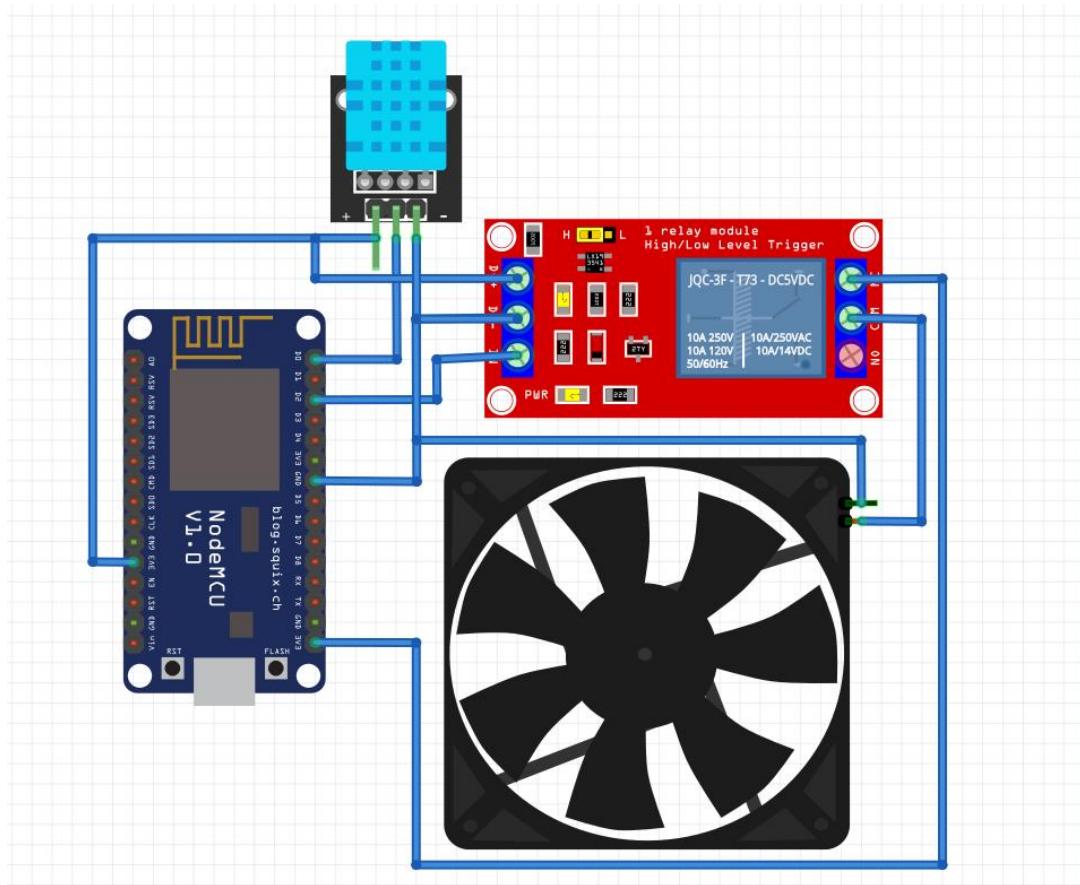
TEMP V1

HUMI V2

RELAY1 V3



HÌNH 2.2 Sơ đồ nguyên lý a



Người dùng có thể điều khiển hai relay thông qua ứng dụng Blynk hoặc nhấn nút vật lý, gửi lệnh từ điện thoại để bật/tắt thiết bị nối với các relay.

ESP8266 nhận tín hiệu điều khiển từ Blynk và điều khiển relay thông qua các chân GPIO, bật/tắt thiết bị theo yêu cầu.

- *Giám sát từ xa:*

Thông tin nhiệt độ và độ ẩm từ cảm biến DHT11 sẽ được gửi lên Blynk, cho phép người dùng giám sát từ xa qua ứng dụng.

## **PHẦN 3: THIẾT KẾ PHẦN MỀM**

### **3.1. CÁC THƯ VIỆN SỬ DỤNG**

- ESP8266WiFi: Kết nối ESP8266 với Wi-Fi, cho phép truyền và nhận dữ liệu qua Internet.
- BlynkSimpleEsp8266: Kết nối ESP8266 với ứng dụng Blynk, giúp điều khiển thiết bị từ xa và gửi dữ liệu lên Blynk để theo dõi qua điện thoại.
- DHTesp: Nó cung cấp các hàm để lấy dữ liệu nhiệt độ và độ ẩm từ cảm biến, giúp ESP8266 dễ dàng thu thập và xử lý thông tin môi trường từ DHT11.

### **3.2. MÔ TẢ CẤU TRÚC CHƯƠNG TRÌNH**

Cấu trúc chương trình gồm các thành phần chính sau:

#### **1. Kết nối và cấu hình**

- **Khai báo thông tin Wi-Fi và Blynk:**
  - Khai báo tên Wi-Fi (ssid) và mật khẩu (pass) để kết nối thiết bị với mạng.
  - Khai báo thông tin từ nền tảng Blynk như Template ID, Template Name, và Auth Token, lấy từ web Blynk IoT.
- **Cấu hình chân GPIO:**
  - Chân DHT11 (dhtPin) được định nghĩa để kết nối cảm biến nhiệt độ/độ ẩm.
  - Các chân relay được gán với các chân GPIO cụ thể trên ESP8266.

#### **2. Kết nối Blynk và điều khiển relay**

- **Đồng bộ trạng thái relay:**
  - BLYNK\_CONNECTED(): Khi thiết bị kết nối thành công với Blynk, trạng thái của relay 1 được đồng bộ từ ứng dụng. Điều này đảm bảo trạng thái relay đúng ngay từ khi khởi động.
- **Điều khiển relay qua Blynk:**
  - BLYNK\_WRITE(RELAY1) :
    - Nhận lệnh điều khiển relay từ ứng dụng Blynk.

- Dựa trên tín hiệu từ người dùng, quạt được bật hoặc tắt thông qua relay.

## 5. Hàm loop()

- **Duy trì kết nối với Blynk:**
  - Blynk.run(): Đảm bảo rằng thiết bị luôn giữ kết nối với máy chủ Blynk, cho phép nhận lệnh và gửi dữ liệu thời gian thực.
- **Đọc cảm biến và gửi dữ liệu:**
  - Mỗi giây, cảm biến DHT11 được đọc để lấy nhiệt độ và độ ẩm.
  - Nếu dữ liệu cảm biến hợp lệ, nhiệt độ và độ ẩm sẽ được gửi đến các chân ảo tương ứng trong Blynk (TEMP và HUMI).
- **Đồng bộ trạng thái relay:**
  - Sau khi đồng bộ, cờ updateRelay sẽ được tắt để chờ lần nhấn tiếp theo.

### 3.3. CODE CHI TIẾT TRÊN ARDUINO IDE

```
#define BLYNK_TEMPLATE_ID "TMPL6uVNDAH4b"
#define BLYNK_TEMPLATE_NAME "BTL"
#define BLYNK_AUTH_TOKEN "U82JVWh4Zm3v_ytmEdHUQzDGqenU9kLR"
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "DHTesp.h"

// Thông tin đăng nhập WiFi
const char* ssid = "Iphone";
const char* pass = "12345678";

DHTesp dht;
#define DHT 16
#define quat 4

unsigned long timeBlink = millis();
boolean updateRelay = 0;
WidgetLED LEDCONNECT(V0);
```

```

#define QUAT V3
#define NHIETDO V1
#define DOAM V2

float temperature = dht.getTemperature();

// Mỗi lần kết nối đến cloud...
BLYNK_CONNECTED() {
    // Yêu cầu trạng thái mới nhất từ máy chủ
    Blynk.syncVirtual(QUAT);
}

// Khi nút bấm ảo trong ứng dụng được nhấn - chuyển đổi trạng thái
BLYNK_WRITE(QUAT) {
    int p = param.asInt();
    digitalWrite(quat, p);
    Blynk.virtualWrite(QUAT, p); // Gửi trạng thái relay tới Blynk
}

void setup() {
    // Debug console
    Serial.begin(115200);

    pinMode(quat, OUTPUT);

    dht.setup(DHT, DHTesp::DHT11);
    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}

void loop() {
    Blynk.run();

    if (millis() - timeBlink > 1000) {
        if (LEDCONNECT.getValue()) {
            LEDCONNECT.off();
        } else {
            LEDCONNECT.on();
        }
    }

    float humidity = dht.getHumidity();
    float temperature = dht.getTemperature();

    if (dht.getStatusString() == "OK") {
        Blynk.virtualWrite(NHIETDO, temperature);
        Blynk.virtualWrite(DOAM, humidity);
    }
}

```

```

}

// Kiểm tra nhiệt độ, bật relay nếu vượt quá 33 độ
if (temperature > 33) {
    digitalWrite(quat, HIGH); // Bật relay
    Blynk.virtualWrite(QUAT, 1); // Cập nhật trạng thái relay lên Blynk
    Blynk.logEvent("CANH_BAO_NHIET_DO", "NHIET DO QUA CAO: " +
String(temperature) + "°C \nBAT QUAT THONG GIO");
}

timeBlink = millis();
}

```

### *Giải thích đoạn code:*

#### **1. Kết nối và cấu hình**

##### **Khai báo thông tin Wi-Fi và Blynk**

```
#define BLYNK_TEMPLATE_ID "TMPL6uVNDAH4b"
#define BLYNK_TEMPLATE_NAME "BTL"
#define BLYNK_AUTH_TOKEN
"U82JVWh4Zm3v_ytmEdHUQzDGqenU9kLR"
```

- BLYNK\_TEMPLATE\_ID: Mã định danh template từ ứng dụng Blynk.
- BLYNK\_TEMPLATE\_NAME: Tên của template Blynk.
- BLYNK\_AUTH\_TOKEN: Mã token xác thực để liên kết thiết bị với nền tảng Blynk.

##### **Khai báo thư viện**

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "DHTesp.h"
```

- ESP8266WiFi.h: Thư viện để kết nối ESP8266 với Wi-Fi.
- BlynkSimpleEsp8266.h: Thư viện giao tiếp với nền tảng Blynk.
- DHTesp.h: Thư viện giao tiếp với cảm biến DHT11.

## Cấu hình cảm biến và chân GPIO

```
DHTesp dht;  
#define DHT 16  
#define quat 4
```

- DHT: Chân GPIO 16 được kết nối với chân dữ liệu của cảm biến DHT11.
- quat: Chân GPIO 4 được kết nối với chân dữ liệu của Module Relay 5V .

## Cấu hình Wifi cho NodeMCU

```
const char* ssid = "Iphone";  
const char* pass = "12345678";
```

- ssid và pass: Tên và mật khẩu của mạng Wi-Fi mà ESP8266 sẽ kết nối.

## Widget ảo

```
unsigned long timeBlink = millis();  
boolean updateRelay = 0;  
WidgetLED LEDCONNECT(V0);
```

- timeBlink và timeDelay: Biến dùng để điều khiển thời gian giữa các tác vụ.
- updateRelay: Cờ xác định trạng thái relay có cần đồng bộ với Blynk hay không.
- LEDCONNECT (V0): LED ảo để hiển thị trạng thái kết nối của esp8266 với ứng dụng Blynk thông qua Blynk IOT.

## Khai báo biến để kết nối Blynk

```
#define QUAT V3  
#define NHIETDO V1  
#define DOAM V2
```

- QUAT (V3) Chân ảo trong ứng dụng Blynk để điều khiển relay.
- NHIETDO (V3) Chân ảo trong Blynk để hiển thị nhiệt độ
- DOAM (V4): Chân ảo trong Blynk để hiển thị độ ẩm.

## 2. Xử lý kết nối Blynk và QUAT

### Đồng bộ trạng thái relay khi kết nối Blynk

```
BLYNK_CONNECTED() {  
    // Yêu cầu trạng thái mới nhất từ máy chủ  
    Blynk.syncVirtual(QUAT);  
}
```

- Khi thiết bị kết nối lại với Blynk, trạng thái của QUAT được đồng bộ từ ứng dụng để đảm bảo trạng thái của QUAT và trạng thái trên Blynk IOT luôn đồng bộ với nhau.

### Nhận lệnh từ ứng dụng để điều khiển relay

```
BLYNK_WRITE(QUAT) {  
    int p = param.asInt();  
    digitalWrite(quat, p);  
    Blynk.virtualWrite(QUAT, p); // Gửi trạng thái relay tới Blynk  
}
```

- Nhận tín hiệu từ Blynk để bật hoặc tắt relay (RELAY1) Giá trị nhận được từ ứng dụng (0 hoặc 1) sẽ được dùng để điều khiển trạng thái chân GPIO của QUAT.

## 3. Hàm setup()

```
void setup() {  
    // Debug console  
    Serial.begin(115200);  
  
    pinMode(quat, OUTPUT);  
  
    dht.setup(DHT, DHTesp::DHT11);  
    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);  
}
```

- Cấu hình Serial để debug.
- Cấu hình chân GPIO cho quat.
- Khởi tạo cảm biến DHT11.
- Kết nối Wi-Fi và liên kết thiết bị với Blynk.

## 5. Hàm loop()

### Duy trì kết nối với Blynk

Blynk.run();

- Duy trì kết nối giữa esp8266 với Blynk để nhận và gửi dữ liệu liên tục.

### Cập nhật trạng thái LED

```
if (millis() - timeBlink > 1000) {
    if (LEDCONNECT.getValue()) {
        LEDCONNECT.off();
    } else {
        LEDCONNECT.on();
    }
}
```

- Cập nhập trạng thái cho LEDCONNECT nhấp nháy để biết được mạch với blynk có đang được kết nối với nhau hay không.

### Đọc nhiệt độ và độ ẩm từ DHT11

```
float humidity = dht.getHumidity();
float temperature = dht.getTemperature();
```

```
Serial.print(dht.getStatusString());
Serial.print("\t");
Serial.print(humidity, 1);
Serial.print("\t\t");
Serial.println(temperature, 1);
```

```
if (dht.getStatusString() == "OK") {
    Blynk.virtualWrite(TEMP, temperature);
```

```

Blynk.virtualWrite(HUMI, humidity);

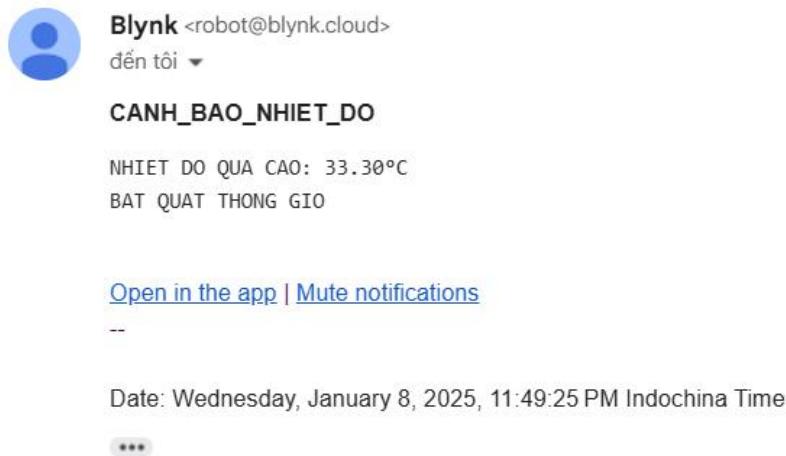
// Kiểm tra nhiệt độ, bật relay nếu vượt quá 30 độ
if (temperature > 35) {
    digitalWrite(relay1, HIGH); // Bật relay
    Blynk.virtualWrite(RELAY1, 1); // Cập nhật trạng thái relay lên Blynk
}
}

timeBlink = millis();
}

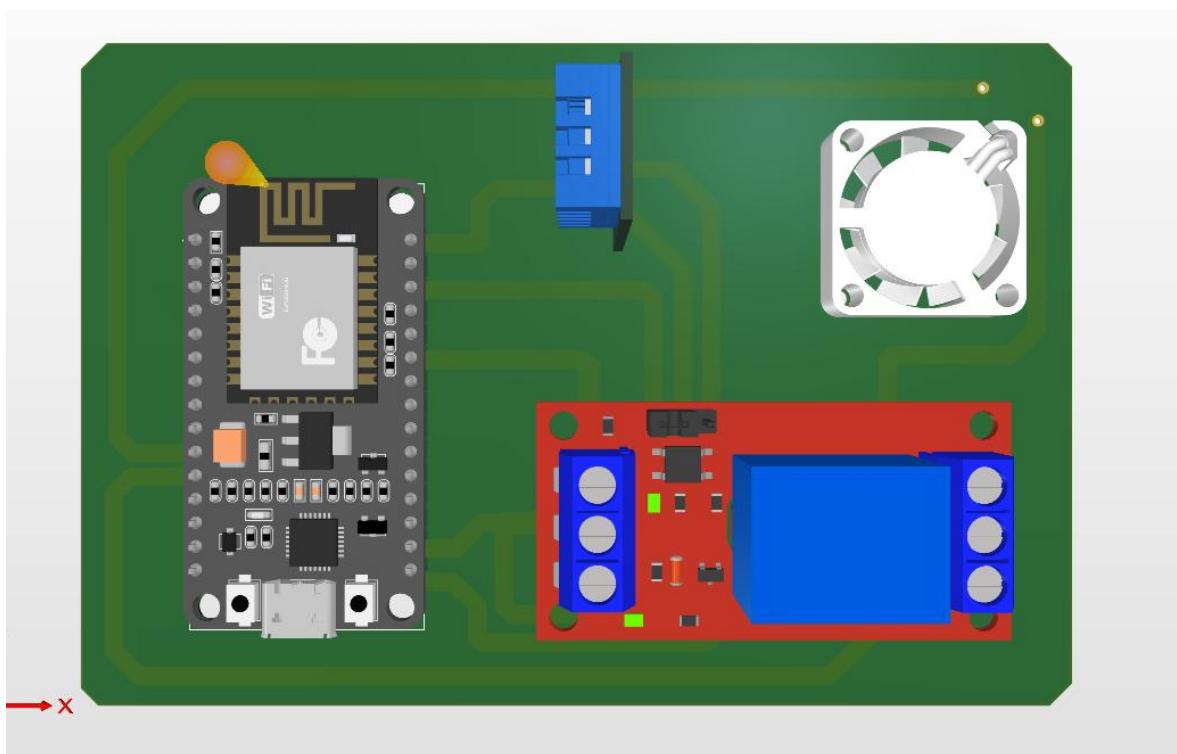
}

```

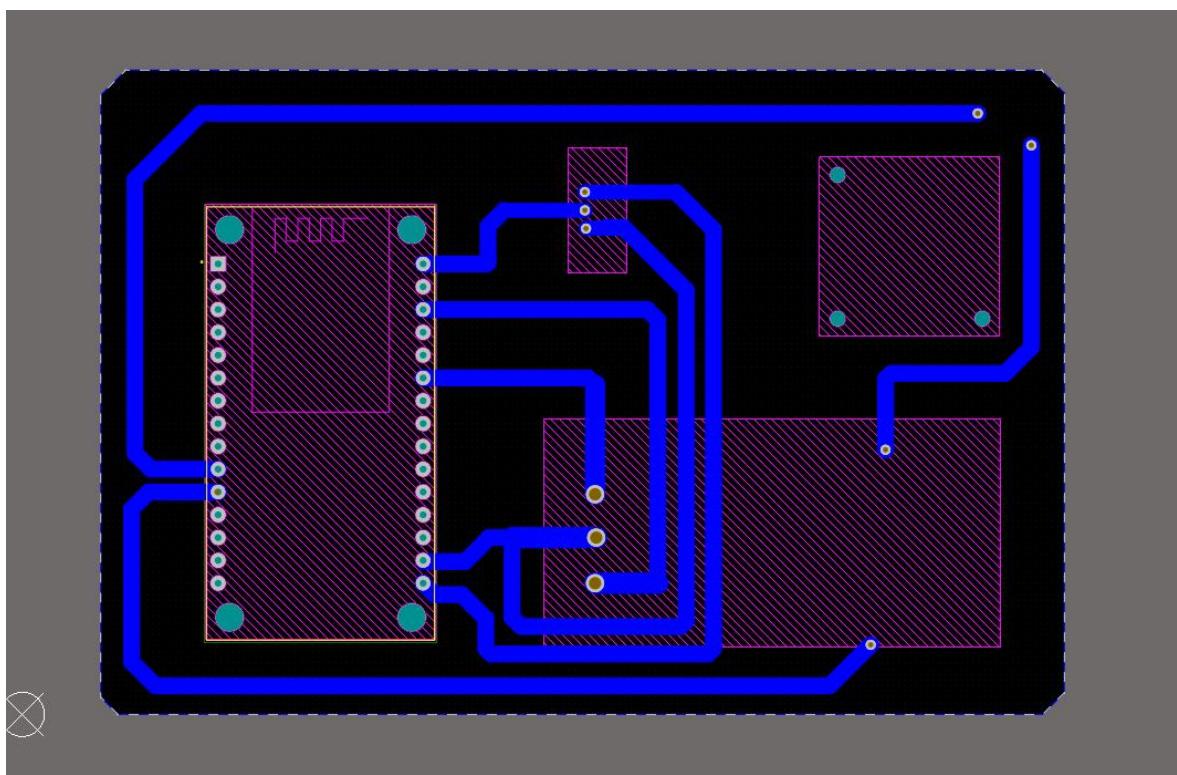
- Đọc dữ liệu từ cảm biến DHT11 và gửi dữ liệu lên Blynk.
- Kiểm tra nhiệt độ (nếu nhiệt độ lớn hơn 33) :
  - Bật quạt thông gió
  - Gửi email cảnh báo cho người dùng.



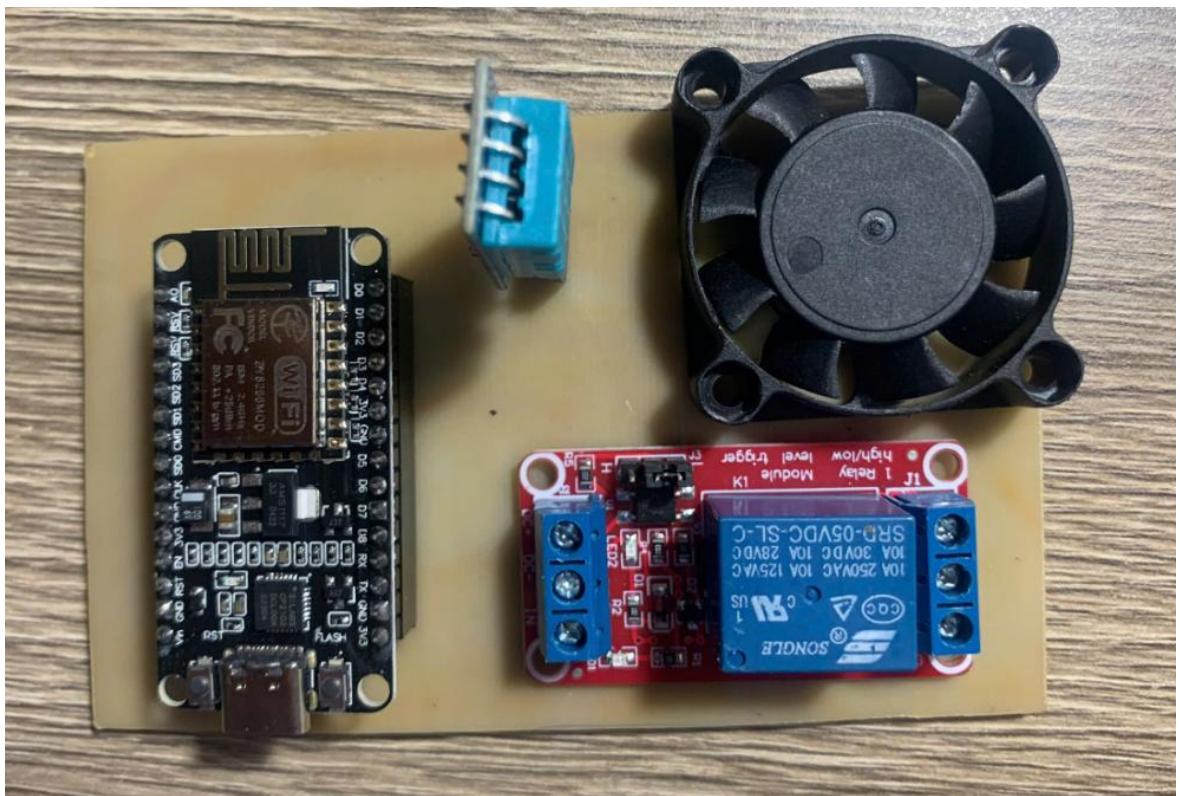
*Hình ảnh mạch PCB được thiết kế trên Altium:*



*Hình ảnh mô phỏng 3D của mạch*



Hình ảnh kết nối để gia công mạch in



Hình ảnh thực tế sau khi gia công mạch

### ● ỨNG DỤNG:

mạch “ Giám sát nhiệt độ, độ ẩm và điều khiển quạt thông gió qua relay thông qua esp8266 và blynk IOT” ứng dụng vào thchuwj té rất nhiều. Mạch ứng dụng cho các nhà sướng, kho lưu trữ để giúp giám sát nhiệt độ độ ẩm từ xa nếu như nhiệt độ và độ ẩm cao thì chúng ta có thể bật quạt thông gió từ xa để giảm nhiệt độ và độ ẩm tránh hư hại tài sản . Ngoài ra mạch còn tích hợp chế độ tự động bật quạt thông gió nếu như nhiệt độ cao quá 33 độ thì sẽ tự động bật quạt thông gió để tránh hư hại tài sản khi chúng ta không thể điều khiển từ xa và gửi email cảnh báo cho chúng ta biết.