TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HCM

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------------------------



**BÁO CÁO BÀI TẬP LỚN**

**HỆ THỐNG THEO DÕI, QUẢN LÝ NHIỆT ĐỘ PHÒNG SERVER**

Các thành viên nhóm:

|  |  |
| --- | --- |
| Họ và tên | Mã số sinh viên |
| Chau Tiểu Long | 21094341 |
| Nguyễn Nhật Tùng | 21096911 |
| Tống Thành Lộc | 21105351 |

Lớp: DHKHMT17B

GV hướng dẫn: Nguyễn Ngọc Lễ

*Thành phố Hồ Chí Minh – Năm 2025*

**Mục lục**

[1 Lời mở đầu (Giới thiệu) 3](#_Toc197760888)

[1.1 Giới thiệu chung về đề tài 3](#_Toc197760889)

[1.2 Lý do chọn đề tài 3](#_Toc197760890)

[1.3 Mục tiêu của báo cáo 4](#_Toc197760891)

[1.4 Phương pháp thực hiện 4](#_Toc197760892)

[2 Cơ sở lý thuyết (nếu có) 5](#_Toc197760893)

[2.1. Khái niệm IoT (Internet of Things) 5](#_Toc197760894)

[2.2 Các công nghệ sử dụng trong hệ thống 6](#_Toc197760895)

[2.2.1 Cảm biến và vi điều khiển. 6](#_Toc197760896)

[2.2.2 Gửi dữ liệu lên nền tảng giám sát 14](#_Toc197760897)

[2.2.3 Mô hình học máy – Random Forest (Giới thiệu) 15](#_Toc197760898)

[3. Nội dung chính (thực hiện đề tài) 18](#_Toc197760899)

[3.1 Thiết kế phần cứng 18](#_Toc197760900)

[3.2 Lập trình 21](#_Toc197760901)

[4. Kết quả và đánh giá 25](#_Toc197760902)

[5. Kết luận và hướng phát triển 26](#_Toc197760903)

# 1 Lời mở đầu (Giới thiệu)

## 1.1 Giới thiệu chung về đề tài

Hệ thống IoT hiển thị dữ liệu nhiệt độ, độ ẩm của phòng máy gửi trực tiếp lên server bằng giao thức MQTT và tin nhắn telegram. Giúp người quản lý nắm bắt thông tin nhanh chóng cũng như đưa ra cảnh báo khi nhiệt độ vượt ngưỡng cho phép. Ngoài ra có tích hợp AI dự đoán dữ liệu tương lai.

## 1.2 Lý do chọn đề tài

Hiện nay, Thực trạng các phòng server ngày càng mở rộng quy mô và tần suất hoạt động. Việc quản lý cũng như giám sát môi trường nhằm đảm điều kiện tốt nhất để hệ thống vận hành một cách trơn tru là nhiệm vụ tiên quyết và cấp bách nhất hiện tại. Do đó, nhằm phục vụ lợi ích cũng như hỗ trợ người quản lý, việc áp dụng một hệ thống giám sát hiện đại sử dụng IoT tích hợp AI có thể giải quyết triệt để vấn đề trên.

## 1.3 Mục tiêu của báo cáo

Trình bày rõ những công việc, quá trình thiết kế và triển khai hệ thống IoT giám sát nhiệt độ độ ẩm có tích hợp mô hình AI. Hướng phát triển hệ thống cũng như việc áp dụng đề tài vào trong thực tế đời sống.

## 1.4 Phương pháp thực hiện

**Thu thập dữ liệu**: Sử dụng cảm biến DHT22 để đo nhiệt độ và độ ẩm trong phòng máy.

**Truyền dữ liệu**: ESP32 kết nối Wi-Fi để gửi dữ liệu đo được lên Adafruit thông qua giao thức MQTT.

**Lưu trữ và xử lý**: Dữ liệu được lưu trữ trên đám mây Adafruit và truyền dữ liệu về máy tính để train và vận hành mô hình AI

**Gửi cảnh báo**: Khi nhiệt độ vượt ngưỡng cho phép, hệ thống tự động gửi thông báo qua Telegram tới người quản lý.

**Dự đoán dữ liệu**: Tích hợp mô hình AI đơn giản (Random forest) để dự đoán nhiệt độ và độ ẩm trong tương lai, giúp đưa ra cảnh báo sớm.

# 2 Cơ sở lý thuyết (nếu có)

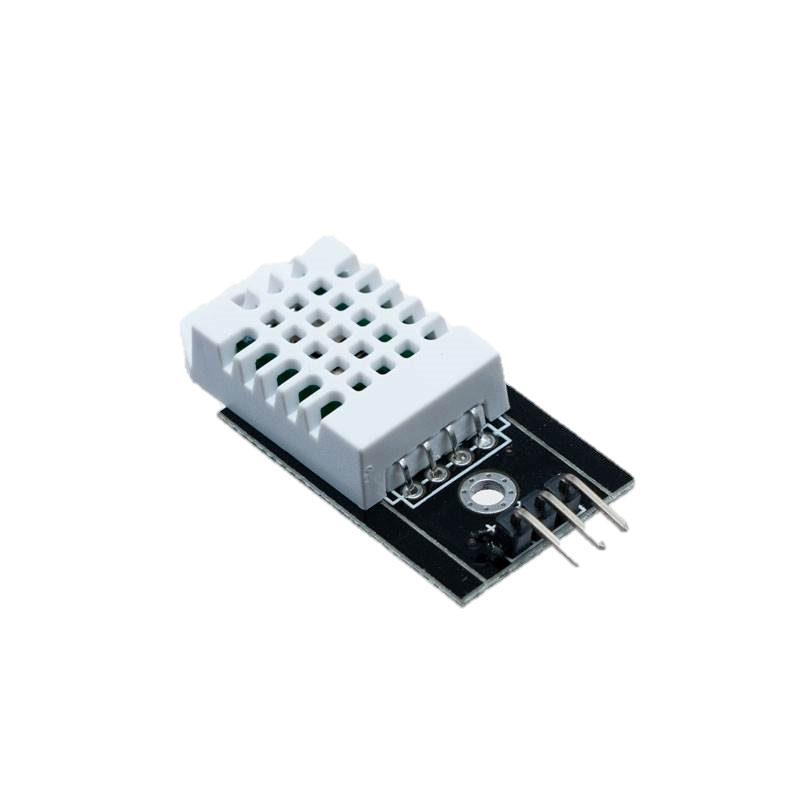
## 2.1. Khái niệm IoT (Internet of Things)

* ***Các thành phần cơ bản của một hệ thống IoT:*** *Ngoài định nghĩa chung, bạn có thể liệt kê và mô tả ngắn gọn các thành phần chính cấu tạo nên một hệ thống IoT, ví dụ như:*
  + ***Thiết bị (Devices):*** *Các cảm biến, bộ chấp hành, máy móc, hoặc bất kỳ đối tượng vật lý nào có khả năng thu thập và/hoặc tác động đến môi trường. Trong trường hợp này, đó là các cảm biến nhiệt độ và độ ẩm.*
  + ***Kết nối (Connectivity):*** *Các giao thức và mạng lưới cho phép các thiết bị kết nối và trao đổi dữ liệu với nhau và với nền tảng trung tâm (ví dụ: Wi-Fi, Bluetooth, Zigbee, Ethernet, mạng di động).*
  + ***Nền tảng (Platform):*** *Phần mềm hoặc dịch vụ đám mây chịu trách nhiệm thu thập, xử lý, lưu trữ và phân tích dữ liệu từ các thiết bị, cũng như quản lý các thiết bị.*
  + ***Ứng dụng (Applications):*** *Giao diện người dùng hoặc các hệ thống khác sử dụng dữ liệu đã được xử lý để cung cấp thông tin hữu ích hoặc thực hiện các hành động cụ thể (ví dụ: hiển thị nhiệt độ/độ ẩm, gửi cảnh báo).*
* ***Đặc điểm nổi bật của IoT:*** *Nhấn mạnh thêm các đặc trưng quan trọng của IoT, chẳng hạn như:*
  + ***Khả năng kết nối:*** *Số lượng lớn các thiết bị được kết nối.*
  + ***Thu thập và xử lý dữ liệu:*** *Khả năng thu thập lượng lớn dữ liệu từ môi trường và xử lý chúng để đưa ra thông tin có giá trị.*
  + ***Tính tự động hóa:*** *Khả năng tự động thực hiện các tác vụ dựa trên dữ liệu thu thập được.*
  + ***Tính thông minh:*** *Khả năng học hỏi và thích ứng dựa trên dữ liệu.*
* ***Ứng dụng rộng rãi của IoT:*** *Mở rộng ra các lĩnh vực khác mà IoT đang được ứng dụng để làm nổi bật tiềm năng của công nghệ này (ví dụ: nhà thông minh, thành phố thông minh, nông nghiệp thông minh, công nghiệp thông minh, y tế thông minh). Điều này giúp người đọc có cái nhìn rộng hơn về bối cảnh của đề tài.*

## 2.2 Các công nghệ sử dụng trong hệ thống

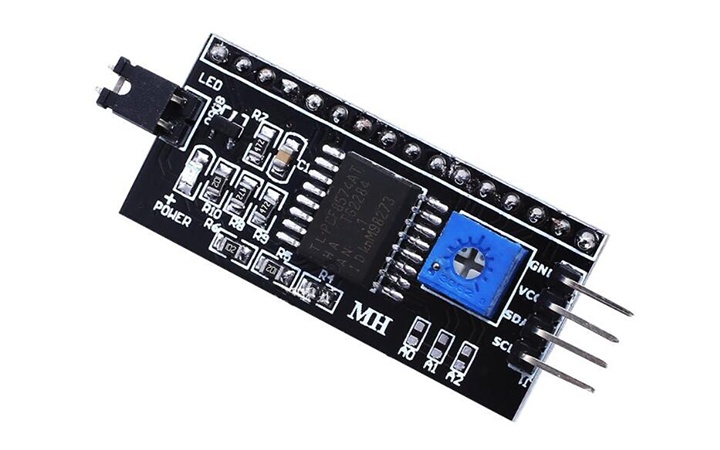
### 2.2.1 Cảm biến và vi điều khiển.

#### a. Cảm biến DHT22



* ***Nguyên lý hoạt động:*** *Giải thích ngắn gọn về cách DHT22 đo nhiệt độ và độ ẩm. Ví dụ, nó thường sử dụng một cảm biến điện dung để đo độ ẩm và một nhiệt điện trở (thermistor) để đo nhiệt độ. IC bên trong sẽ xử lý tín hiệu và xuất ra dữ liệu số.*
* ***Thông số kỹ thuật quan trọng:*** *Liệt kê một vài thông số kỹ thuật chính của DHT22 để làm rõ hơn về khả năng của nó, ví dụ như:*
  + ***Dải đo nhiệt độ:*** *(ví dụ: -40°C đến 80°C)*
  + ***Độ chính xác nhiệt độ:*** *(ví dụ: ±0.5°C)*
  + ***Dải đo độ ẩm:*** *(ví dụ: 0% đến 99.9% RH)*
  + ***Độ chính xác độ ẩm:*** *(ví dụ: ±2% RH)*
  + ***Điện áp hoạt động:*** *(ví dụ: 3.3V - 5V)*
  + ***Giao tiếp:*** *(thường là giao tiếp kỹ thuật số một dây)*
* ***Ưu điểm khi sử dụng DHT22 trong ứng dụng này:*** *Nhấn mạnh lại tại sao DHT22 là lựa chọn phù hợp, ví dụ: độ chính xác tốt hơn DHT11, dải đo rộng, giá thành hợp lý, dễ dàng giao tiếp với vi điều khiển.*
* ***Lưu ý khi sử dụng:*** *Đề cập đến một vài lưu ý khi làm việc với DHT22, chẳng hạn như thời gian lấy mẫu tối thiểu, ảnh hưởng của nhiễu, và khoảng cách kết nối tối đa.*

#### b . Cảm biến LDC I2C



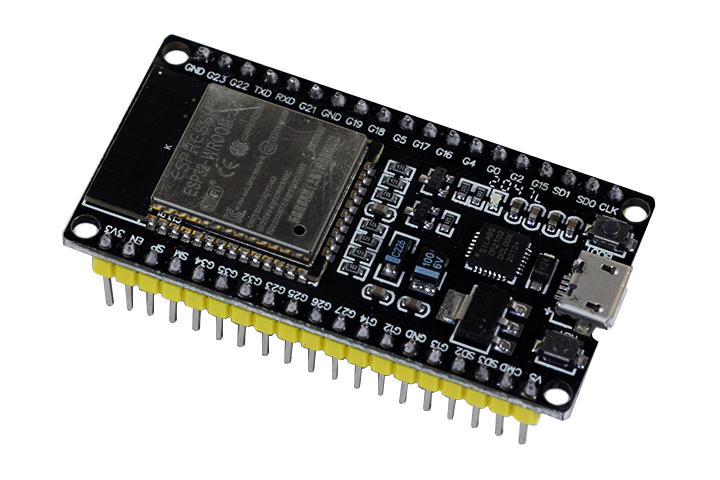
* ***Nguyên lý hoạt động:*** *Cảm biến LDC hoạt động dựa trên nguyên lý cảm ứng điện từ. Nó thường bao gồm một cuộn dây (inductor) và một mạch điện tử để đo lường sự thay đổi trở kháng hoặc tần số cộng hưởng của cuộn dây khi có một vật dẫn kim loại (target) di chuyển đến gần hoặc ra xa cuộn dây. IC bên trong sẽ xử lý sự thay đổi này và chuyển đổi nó thành dữ liệu số, sau đó truyền qua giao tiếp I2C.*
* ***Thông số kỹ thuật quan trọng:*** *(Các thông số này sẽ thay đổi tùy thuộc vào cảm biến LDC I2C cụ thể bạn sử dụng. Đây chỉ là ví dụ dựa trên dòng LDC1612/LDC1614)*
  + ***Dải đo khoảng cách:*** *(ví dụ: vài micromet đến vài milimet, tùy thuộc vào kích thước cuộn dây và target)*
  + ***Độ phân giải:*** *(ví dụ: có thể đạt đến hàng nanomet)*
  + ***Độ chính xác:*** *(tùy thuộc vào cấu hình và môi trường)*
  + ***Điện áp hoạt động:*** *(ví dụ: 3.3V hoặc 5V)*
  + ***Giao tiếp:*** *I2C (giao thức giao tiếp nối tiếp hai dây)*
  + ***Tốc độ lấy mẫu:*** *(có thể điều chỉnh được)*
  + ***Số kênh:*** *(một số chip LDC có thể hỗ trợ nhiều kênh cảm biến)*
* ***Ưu điểm khi sử dụng cảm biến LDC I2C trong ứng dụng này (ví dụ, nếu dùng để phát hiện xâm nhập hoặc rung động):***
  + ***Độ phân giải cao:*** *Có khả năng phát hiện những thay đổi rất nhỏ về vị trí hoặc sự hiện diện của vật kim loại.*
  + ***Không tiếp xúc:*** *Đo lường không cần tiếp xúc trực tiếp với đối tượng, tăng độ bền và giảm hao mòn.*
  + ***Khả năng hoạt động trong môi trường khắc nghiệt:*** *Ít bị ảnh hưởng bởi bụi bẩn, dầu mỡ so với các cảm biến quang học.*
  + ***Giao tiếp I2C:*** *Dễ dàng kết nối và giao tiếp với nhiều vi điều khiển, bao gồm ESP32, chỉ với hai dây tín hiệu.*
  + ***Tiết kiệm năng lượng:*** *Một số cảm biến LDC có chế độ hoạt động tiết kiệm năng lượng.*

#### *c. Còi báo động (Buzzer)*



* ***Nguyên lý hoạt động:*** *Còi báo động là một thiết bị âm thanh đơn giản, thường hoạt động dựa trên nguyên lý điện từ hoặc áp điện.*
  + ***Còi điện từ:*** *Khi có dòng điện chạy qua cuộn dây bên trong, nó tạo ra một trường điện từ làm rung một màng kim loại, tạo ra âm thanh.*
  + ***Còi áp điện:*** *Sử dụng hiệu ứng áp điện, khi một điện áp được đặt vào vật liệu áp điện (thường là gốm), nó sẽ bị biến dạng cơ học, tạo ra dao động và do đó tạo ra âm thanh.*
* ***Thông số kỹ thuật quan trọng:***
  + ***Điện áp hoạt động:*** *(ví dụ: 5V, 12V) - cần phù hợp với nguồn điện của hệ thống.*
  + ***Dòng điện tiêu thụ:*** *(thường là vài chục đến hàng trăm miliampe) - cần đảm bảo vi điều khiển hoặc mạch điều khiển có thể cung cấp đủ dòng.*
  + ***Cường độ âm thanh (SPL - Sound Pressure Level):*** *(đơn vị dB - decibel, ví dụ: 85dB ở khoảng cách 10cm) - cho biết độ lớn của âm thanh phát ra.*
  + ***Tần số âm thanh:*** *(đơn vị Hz - Hertz) - xác định âm vực của tiếng còi. Một số còi có tần số cố định, trong khi một số khác có thể thay đổi tần số để tạo ra các âm thanh khác nhau.*
  + ***Loại còi:***
    - ***Còi thụ động (Passive buzzer):*** *Cần được cấp xung vuông (PWM) từ vi điều khiển để tạo ra âm thanh với tần số mong muốn. Cho phép tạo ra nhiều âm điệu khác nhau.*
    - ***Còi chủ động (Active buzzer):*** *Đã tích hợp sẵn mạch dao động bên trong, chỉ cần cấp nguồn DC là sẽ phát ra âm thanh ở một tần số cố định. Dễ sử dụng hơn cho các cảnh báo đơn giản.*
* ***Ưu điểm khi sử dụng còi báo động trong ứng dụng này:***
  + ***Cảnh báo trực quan và dễ nhận biết:*** *Âm thanh là một hình thức cảnh báo hiệu quả, đặc biệt khi người quản trị không nhìn trực tiếp vào màn hình giám sát.*
  + ***Đơn giản và dễ tích hợp:*** *Còi là một thiết bị tương đối đơn giản và dễ dàng kết nối với vi điều khiển như ESP32 thông qua một chân GPIO.*
  + ***Chi phí thấp:*** *Còi báo động có giá thành khá rẻ.*
  + ***Linh hoạt (đối với còi thụ động):*** *Có thể tạo ra nhiều kiểu âm thanh cảnh báo khác nhau để phân biệt các loại sự cố*

#### d. Vi điều khiển ESP32



* ***Kiến trúc và tính năng nổi bật:*** *Mô tả ngắn gọn về kiến trúc của ESP32, tốc độ xử lý, bộ nhớ, và các ngoại vi tích hợp khác ngoài Wi-Fi và Bluetooth (ví dụ: GPIO, ADC, DAC, SPI, I2C).*
* ***Ưu điểm khi lựa chọn ESP32 cho ứng dụng IoT:*** *Giải thích lý do ESP32 là một lựa chọn mạnh mẽ cho dự án này:*
  + ***Tích hợp Wi-Fi và Bluetooth:*** *Giúp việc kết nối internet và các thiết bị khác trở nên dễ dàng.*
  + ***Hiệu năng cao:*** *Đủ sức mạnh để xử lý dữ liệu cảm biến, giao tiếp mạng và có thể thực hiện các tác vụ phức tạp hơn trong tương lai.*
  + ***Giá thành cạnh tranh:*** *Là một lựa chọn kinh tế cho các dự án IoT.*
  + ***Cộng đồng hỗ trợ lớn:*** *Có nhiều tài liệu, thư viện và ví dụ mã nguồn mở, giúp quá trình phát triển dễ dàng hơn.*
  + ***Hỗ trợ nhiều giao thức:*** *Khả năng hỗ trợ MQTT và HTTP rất phù hợp cho việc gửi dữ liệu lên các nền tảng đám mây.*
* ***Các phương thức giao tiếp mạng:*** *Giải thích ngắn gọn về giao thức MQTT (Message Queuing Telemetry Transport) - một giao thức nhẹ nhàng, phù hợp cho các ứng dụng IoT có băng thông hạn chế và độ tin cậy không cao - và giao thức HTTP (Hypertext Transfer Protocol) - giao thức phổ biến cho việc truyền tải dữ liệu trên web. Nêu rõ lý do tại sao bạn có thể cân nhắc sử dụng một trong hai giao thức này.*

### 2.2.2 Gửi dữ liệu lên nền tảng giám sát

#### *Telegram Bot API*

* + ***Nguyên lý hoạt động:*** *Mô tả cách Telegram Bot API hoạt động. Bạn tạo một bot thông qua BotFather trên Telegram, sau đó sử dụng API (thường là thông qua các thư viện lập trình) để gửi và nhận tin nhắn, điều khiển bot.*
  + ***Ưu điểm cho việc gửi cảnh báo:*** *Nhanh chóng, dễ dàng tích hợp, thông báo trực tiếp đến người quản trị, có thể tùy chỉnh nội dung cảnh báo.*
  + ***Hạn chế:*** *Có thể không phù hợp để hiển thị dữ liệu trực quan hoặc lưu trữ lịch sử dữ liệu chi tiết.*

#### *Adafruit IO*

* + ***Kiến trúc và tính năng chính:*** *Mô tả Adafruit IO là một nền tảng đám mây được thiết kế đặc biệt cho các dự án IoT. Nhấn mạnh các tính năng quan trọng như tạo dashboard trực quan, lưu trữ dữ liệu theo thời gian, tạo trigger và actions (ví dụ: gửi email, webhook khi có sự kiện), và khả năng tích hợp với nhiều loại thiết bị và dịch vụ khác.*
  + ***Ưu điểm khi sử dụng Adafruit IO:*** *Dễ sử dụng, giao diện trực quan, miễn phí cho một số lượng dữ liệu và thiết bị nhất định, cung cấp các widget đa dạng để hiển thị dữ liệu (biểu đồ, số liệu, nút điều khiển...).*
  + ***Cách thức hoạt động trong hệ thống của bạn:*** *Giải thích cách ESP32 sẽ gửi dữ liệu nhiệt độ và độ ẩm lên các "feeds" (kênh dữ liệu) trên Adafruit IO, và cách bạn sẽ sử dụng dashboard để theo dõi dữ liệu này.*

### 2.2.3 Mô hình học máy – Random Forest (Giới thiệu)

#### 2.2.3.1 Lý thuyết

* ***Học máy (Machine Learning) là gì?*** *Nói một cách đơn giản, học máy là một lĩnh vực của trí tuệ nhân tạo (AI) cho phép máy tính học hỏi từ dữ liệu mà không cần được lập trình một cách rõ ràng cho từng tác vụ cụ thể. Thay vào đó, các thuật toán học máy sẽ tìm kiếm các mẫu và quy luật trong dữ liệu để đưa ra dự đoán hoặc quyết định.*
* ***Giới thiệu về mô hình cây quyết định (Decision Tree):*** *Cây quyết định là một loại mô hình học máy sử dụng cấu trúc dạng cây để đưa ra quyết định. Mỗi nút trên cây đại diện cho một thuộc tính (ví dụ: nhiệt độ hiện tại), mỗi nhánh đại diện cho một quyết định hoặc một phạm vi giá trị của thuộc tính đó, và mỗi lá đại diện cho một kết quả dự đoán. Cây quyết định dễ hiểu và trực quan, nhưng có thể dễ bị "quá khớp" (overfitting) với dữ liệu huấn luyện, dẫn đến hiệu suất kém trên dữ liệu mới.*
* ***Random Forest là gì?*** *Random Forest là một thuật toán học máy thuộc nhóm "học ансамбль" (ensemble learning). Thay vì sử dụng một cây quyết định duy nhất, Random Forest xây dựng một "rừng" gồm nhiều cây quyết định độc lập. Khi cần đưa ra dự đoán, mỗi cây trong rừng sẽ dự đoán một kết quả, và kết quả cuối cùng thường là kết quả phổ biến nhất (đối với bài toán phân loại) hoặc trung bình (đối với bài toán hồi quy) của tất cả các cây.*

#### 2.2.3.2 Ứng dụng

* ***Ứng dụng Random Forest trong dự đoán chuỗi thời gian:*** *Random Forest có thể được áp dụng cho các bài toán dự đoán chuỗi thời gian, trong đó mục tiêu là dự đoán các giá trị trong tương lai dựa trên các giá trị trong quá khứ. Để làm điều này, chúng ta có thể coi các giá trị nhiệt****ách Random Forest được sử dụng trong đề tài:*** *Trong hệ thống của bạn, mô hình Random Forest sẽ được huấn luyện trên dữ liệu lịch sử về nhiệt độ và độ ẩm đã được thu thập từ cảm biến DHT22. Sau khi được huấn luyện, mô hình này có thể nhận các giá trị nhiệt độ và độ ẩm hiện tại (và có thể cả các giá trị ở một vài thời điểm trước đó) làm đầu vào, và dự đoán giá trị nhiệt độ và độ ẩm trong tương lai gần (ví dụ: 3 phút tới).*
* ***Lợi ích của việc dự đoán:*** *Việc dự đoán nhiệt độ và độ ẩm trong tương lai có thể mang lại nhiều lợi ích:*
  + ***Cảnh báo sớm hơn:*** *Nếu mô hình dự đoán nhiệt độ hoặc độ ẩm có xu hướng vượt quá ngưỡng an toàn trong tương lai gần, hệ thống có thể gửi cảnh báo sớm hơn, cho phép người quản trị có thời gian để can thiệp trước khi sự cố thực sự xảy ra.*
  + ***Điều khiển hệ thống làm mát thông minh hơn:*** *Dựa trên dự đoán, hệ thống có thể tự động điều chỉnh hoạt động của hệ thống làm mát một cách chủ động hơn, thay vì chỉ phản ứng khi nhiệt độ đã vượt ngưỡng. Ví dụ, nếu dự đoán nhiệt độ sẽ tăng nhanh, hệ thống có thể kích hoạt quạt làm mát sớm hơn và mạnh hơn.*
  + ***Tối ưu hóa năng lượng:*** *Việc điều khiển hệ thống làm mát một cách thông minh dựa trên dự đoán có thể giúp tối ưu hóa việc sử dụng năng lượng.*
* ***Luồng dữ liệu và tần suất cập nhật:*** *Bạn đã đề cập rằng dữ liệu từ cảm biến sẽ được gửi lên Adafruit IO sau mỗi 15 giây. Dữ liệu này có thể được sử dụng để liên tục cập nhật và huấn luyện lại mô hình Random Forest (hoặc sử dụng một mô hình đã được huấn luyện trước đó để đưa ra dự đoán). Việc dự đoán cho 3 phút tới có nghĩa là mô hình sẽ cố gắng ước tính giá trị nhiệt độ và độ ẩm tại thời điểm 3 phút sau thời điểm hiện tại.*

# 3. Nội dung chính (thực hiện đề tài)

## *3.1 Thiết kế phần cứng*

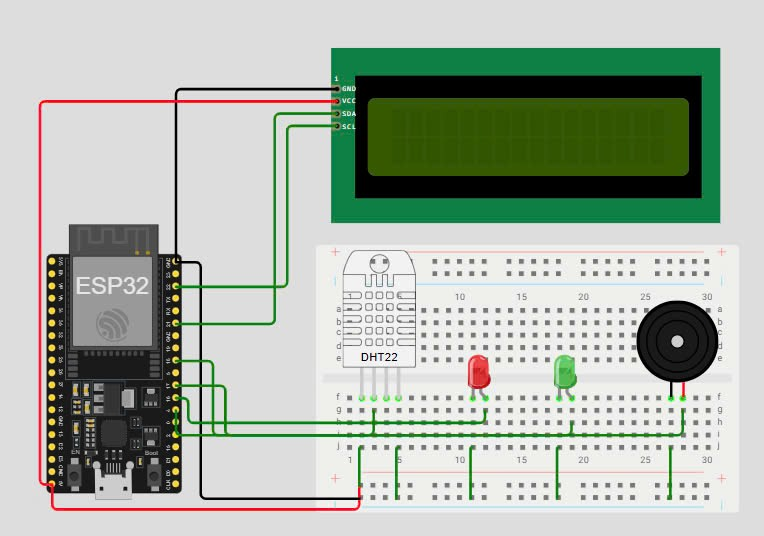
## *Công cụ sử dụng: Wokwi (*[*https://wokwi.com/*](https://wokwi.com/)*)*

***Mô tả:*** *Dựng sơ đồ mạch sử dụng các linh kiện sau:*

* *ESP32*
* *Cảm biến nhiệt độ & độ ẩm DHT22*
* *Màn hình LCD I2C*
* *Buzzer*
* *LED cảnh báo*

*Sơ đồ được lưu dưới định dạng .json để dễ dàng nhập vào PlatformIO.*

Hình ảnh mạch được lắp gắp:



Cách lắp đặt:

#### 1. DHT22 – Cảm biến nhiệt độ và độ ẩm

* **DATA (chân giữa)** → GPIO **4**
* **VCC** → **3.3V**
* **GND** → **GND**

#### 2. LCD I2C (LiquidCrystal\_I2C)

* Giao tiếp I2C sử dụng 2 chân:
  + **SDA** → GPIO **21**
  + **SCL** → GPIO **22**
* **VCC** → **5V**
* **GND** → **GND**

#### 3. LED Cảnh báo (2 LED)

* **Cực dương (Anode) của LED** → GPIO **16 (đỏ), 18(xanh)**
* **Cực âm (Cathode)** → **GND**

#### 4. Buzzer (còi báo động)

* **Chân dương (+)** → GPIO **17**
* **Chân âm (–)** → **GND**

## 3.2 Lập trình

* **Môi trường phát triển:** PlatformIO (Visual Studio Code)
* **Ngôn ngữ sử dụng:** C++
* **Các thư viện sữ dụng:**
  + WiFi.h để kết nối WiFi
  + PubSubClient.h để giao tiếp MQTT với Adafruit IO
  + DHT.h để đọc cảm biến DHT22
  + LiquidCrystal\_I2C.h để điều khiển màn hình LCD
  + HTTPClient.h và WiFiClientSecure.h để gửi dữ liệu qua Telegram

**Tính năng chính:**

* Đọc nhiệt độ, độ ẩm từ cảm biến DHT22
* Hiển thị lên LCD
* Gửi dữ liệu mỗi 15s lên Adafruit IO (MQTT)
* Gửi thông báo mỗi 60s lên Telegram
* Nếu thay đổi nhiệt độ > 2°C hoặc độ ẩm > 5%, gửi cảnh báo Telegram
* Nếu nhiệt độ > 30°C, buzzer và LED nhấp nháy cảnh báo
* Đồng bộ thời gian với NTP để ghi log chính xác

3.3 triển khai

Trước khi triển khai ta cần tải 2 extensions: platform IO và Wokwi simulator.

Sau khi đã có 2 extensions và file json chứa mạch cần làm. Ta vào platform IO tạo dự án mới và thực hiện các bước sau:

Bước 1: Lưu copy file json vào dự án đặt tên là diagram.json

Bước 2: Lập trình trong phần scr/main.cpp

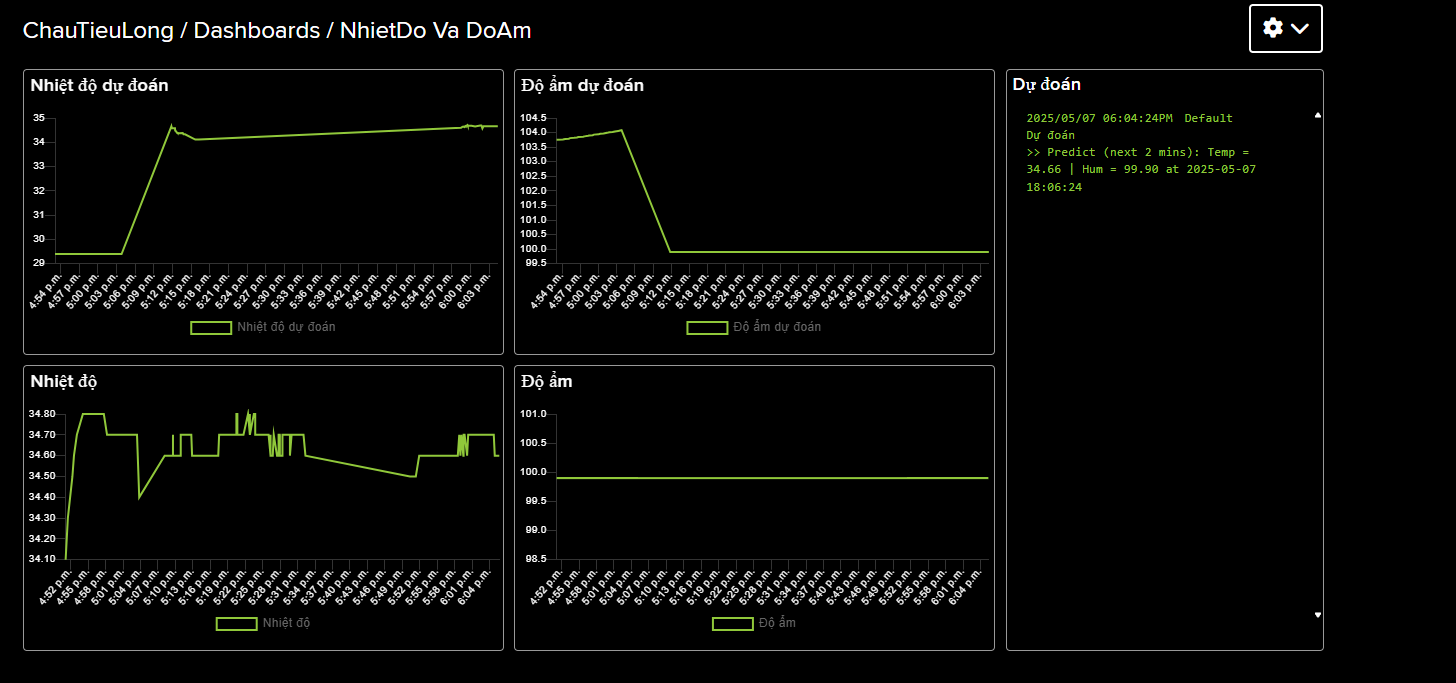
Bước 3:

Trên web [io.adafruit.com](http://io.adafruit.com):

Tạo dự án đặt tên là NhietDo va DoAm. Tiếp theo tạo 5 feed đặt tên lần lượt là

* Nhiệt đô
* Độ ẩm
* Nhiệt độ dự đoán
* Độ ấm dự đoán
* Dự đoán (Chứa txt dự đoán)

Vào dashboard tạo 5 biễu đồ char tương ứng với 5 feed vừa tạo, được như hình sau:



Lấy Usename và key để lấy địa chỉ 5 feed đưa dữ liệu lên và lập trình

Trên telegram:

Truy cập **https://t.me/BotFather** để tạo một **Telegram Bot** mới.

Sử dụng lệnh /newbot, đặt tên và username cho bot. Sau khi hoàn tất, BotFather sẽ cung cấp **Token** dùng để gửi tin nhắn.

Truy cập **https://web.telegram.org/** để lấy **Chat ID**:  
 + Tìm kiếm và gửi tin nhắn bất kỳ cho bot vừa tạo.  
 + Truy cập đường dẫn sau để lấy thông tin chat\_id:

|  |
| --- |
| https://api.telegram.org/bot<YOUR\_BOT\_TOKEN>/getUpdates |

* Thay <YOUR\_BOT\_TOKEN> bằng token nhận được từ BotFather.

Sau khi có BOT\_TOKEN và CHAT\_ID tiến hành lập trình, được hình như sau:



Bước 4 Tiến hành lập trình trên thiết bị thật.

Đâu tiên kiểm tra máy tính có driver kết nối cổng COM chưa.

Nếu chưa tiến hành tải driver CH341SER từ web sau:

<https://www.wch.cn/downloads/ch341ser_exe.html>

Sau khi có driver thì lắp đặt thiết bị ngoài như mạch ảo (Hình mạch lắp đặt)

Build mô hình trong visual studio code sau đó upload lên ESP32. Phải có điện thì thiết bị esp32 mới hoạt động liên tục

# *4. Kết quả và đánh giá*

* *Trình bày kết quả đạt được*
* Hệ thống đã đưa dữ liệu lên màn hình ldc\_i2c, web adafruit sau mỗi 15s và 60s trên telegram đúng với thực tế.
* Hệ thống đưa ra thông báo nếu nhiệt độ quá cao hay bất thường nó sẽ gửi thông bằng cách dèn đỏ sáng chớp nháy và còi sẽ hú. Ngoài ra, nó cũn thông báo biến động trên telegram
* *Đánh giá kết quả so với mục tiêu ban đầu*
* Hệ thống vận hành tốt, không xảy ra lỗi. Đạt được mục tiêu hoàn thiện ban đầu
* Mô hình AI hoạt động khá tốt, dự đoán được sinh ra chính xác và sát với thời gian thực. Vượt kì vọng ban đầu chỉ hoạt động được và ổn.
* *Hạn chế và khó khăn gặp phải*
* Mô hình AI đã hoạt động tốt tuy nhiên chưa đủ dữ liệu để train nên dữ liệu đầu ra có phần hạn chế.
* Chập mạch khi nối sai chân.
* Phải đảm bảo kết nối wifi ổn định trong suốt quá trình thực hiện hệ thống.

# *5. Kết luận và hướng phát triển*

* *Tổng kết nội dung đã thực hiện*
* Trong khuôn khổ đồ án, nhóm đã hoàn thành việc xây dựng hệ thống IoT giám sát nhiệt độ và độ ẩm trong phòng máy. Hệ thống sử dụng vi điều khiển ESP32 kết nối với cảm biến DHT22, màn hình LCD I2C và các thiết bị cảnh báo (LED, buzzer) để hiển thị và phản ứng với dữ liệu môi trường theo thời gian thực. Dữ liệu được gửi lên server thông qua giao thức MQTT và đồng thời thông báo đến người dùng qua ứng dụng Telegram khi phát hiện sự thay đổi bất thường. Ngoài ra, hệ thống có khả năng mở rộng với mô hình trí tuệ nhân tạo dự đoán xu hướng nhiệt độ, độ ẩm trong tương lai, giúp nâng cao khả năng giám sát và ứng phó sớm.
* *Nêu hướng phát triển nếu tiếp tục mở rộng đề tài*
* Tích hợp phần tự động điều khiển thiết bị: như máy quạt, điều hòa, máy hút ẩm nhằm kiểm soát nhiệt độ, độ ẩm
* Lưu trữ xữ lý dữ liệu lâu dài và vận hành AI trên server thay thế cho máy local
* Nâng cấp giao diện hiển thị dashboard thân thiện với người dùng, đẹp mắt hơn.
* Mở rộng giám sát nhiều điểm: sử dụng nhiều cảm biến hơn và kết nối chúng với nhau để thành một kiến trúc quản lý chặt chẽ.

*Tài liệu tham khảo*

*Adafruit. (2023). Adafruit IO MQTT API. Adafruit Documentation.* [*https://io.adafruit.com/api/docs/mqtt.html*](https://io.adafruit.com/api/docs/mqtt.html) *Adafruit. (2024). Adafruit IO documentation.* [*https://learn.adafruit.com/adafruit-io*](https://learn.adafruit.com/adafruit-io) *Aosong Electronics. (2018). DHT22 datasheet.* [*https://www.aosong.com*](https://www.aosong.com) *Bán Linh Kiện. (2024). Cài driver USB to COM CH340/CH341 trên Windows.* [*https://banlinhkien.com/cai-driver-usb-to-com-ch340-ch341-tren-windows-n85184.html*](https://banlinhkien.com/cai-driver-usb-to-com-ch340-ch341-tren-windows-n85184.html) *Điện Tử DAT. (n.d.). Module còi buzzer 5V.* [*https://dientudat.com/san-pham/module-coi-buzzer-5v/*](https://dientudat.com/san-pham/module-coi-buzzer-5v/) *Espressif Systems. (2024). ESP32 technical reference manual.* [*https://docs.espressif.com/projects/esp-idf/en/latest/esp32/*](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/) *Scikit-learn Developers. (2024). Random forest regression.* [*https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html*](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html) *Telegram. (2024). Telegram Bot API – Official documentation.* [*https://core.telegram.org/bots/api*](https://core.telegram.org/bots/api) *Tuyết Nga Electronics. (n.d.). LCD I2C 16x2 datasheet.* [*https://tuyetnga.com.vn/lcd-16x2-i2c*](https://tuyetnga.com.vn/lcd-16x2-i2c)