

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
TRƯỜNG ĐIỆN – ĐIỆN TỬ



**BÁO CÁO ĐỒ ÁN 2**  
***THIẾT KẾ HỆ THỐNG TRẠM QUAN***  
***TRẮC KHÔNG KHÍ***

**Lớp:** *ET – E4 01 K65*

**GVHD:** *PSG.TS. Nguyễn Hữu Phát*

**Sinh viên:** *Trần Đình Nhật Thăng – 20200606*

*Nguyễn Minh Quang - 202020491*

Hanoi, 03/02/2024

# LỜI MỞ ĐẦU

Trong thời đại hiện đại, vấn đề ô nhiễm không khí ngày càng trở nên nghiêm trọng, ảnh hưởng đến sức khỏe cộng đồng và gây thiệt hại đáng kể cho môi trường. Để đối mặt với thách thức này, việc xây dựng và thiết kế hệ thống trạm quan trắc không khí đóng vai trò quan trọng trong việc giám sát và đánh giá chất lượng không khí một cách hiệu quả. Đồng thời, việc sử dụng dữ liệu từ hệ thống này để dự đoán xu hướng và biến động trong tương lai đóng vai trò quan trọng để phát triển các biện pháp ngăn chặn và cải thiện chất lượng không khí.

Đề tài "**Thiết kế hệ thống trạm quan trắc không khí**" mà chúng em trình bày trong báo cáo này không chỉ tập trung vào việc phát triển một hệ thống trạm quan trắc hiện đại mà còn nhấn mạnh vào khả năng dự đoán tình hình không khí trong tương lai. Chúng em đã kết hợp các công nghệ tiên tiến và mô hình dự đoán để cung cấp thông tin chính xác và đầy đủ về chất lượng không khí.

Chúng em tin rằng báo cáo này sẽ không chỉ đem lại cái nhìn sâu sắc về hệ thống trạm quan trắc không khí mà còn giới thiệu một phương pháp mới để dự đoán tình hình không khí trong tương lai, từ đó giúp cộng đồng và các nhà quản lý đưa ra những quyết định thông minh và có hiệu quả để bảo vệ môi trường sống của chúng ta.

# MỤC LỤC

<b>1. GIỚI THIỆU CHUNG.....</b>	<b>4</b>
1.1 Đặt vấn đề .....	4
1.2 Ý tưởng và mục tiêu hướng tới.....	4
<b>2. KẾ HOẠCH VÀ PHÂN CÔNG CÔNG VIỆC .....</b>	<b>5</b>
2.1 Kế hoạch chi tiết .....	5
<b>3. CƠ SỞ LÝ THUYẾT .....</b>	<b>5</b>
3.1 Giao thức MQTT.....	5
3.2 Giao tiếp UART .....	6
3.3 Giao tiếp I2C.....	7
3.4 Hệ điều hành RTOS .....	7
<b>3.5 Mô hình mạng LSTM: .....</b>	<b>8</b>
3.5.1. Khái niệm Deep Learning: .....	8
3.5.2. Tiền xử lý dữ liệu trong Deep Learning .....	8
3.5.3. Tổng quan LSTM.....	9
3.5.4. Kiến trúc LSTM.....	10
3.5.5. Ứng dụng LSTM .....	13
<b>3.6 Tổng quan về AQI:.....</b>	<b>13</b>
3.6.1. Khái niệm: .....	13
3.6.2. Tính toán AQI: .....	13
3.6.3. AQI tại nước ta:.....	14
<b>4. YÊU CẦU THIẾT KẾ .....</b>	<b>15</b>
<b>5. THIẾT KẾ HỆ THỐNG.....</b>	<b>16</b>
5.1 Phân tích, lựa chọn linh kiện .....	16
5.2 Thiết kế phần cứng.....	20
<b>6. KẾT LUẬN .....</b>	<b>22</b>
<b>6.1 Đánh giá kết quả.....</b>	<b>22</b>
5.2.3. Đánh giá kết quả dự án.....	22
5.2.4. Đánh giá kết quả làm việc nhóm.....	Error! Bookmark not defined.
<b>6.2 Hướng phát triển tương lai.....</b>	<b>25</b>
<b>6.3 Lời cảm ơn .....</b>	<b>Error! Bookmark not defined.</b>

**TÀI LIỆU THAM KHẢO.....** ERROR! BOOKMARK NOT DEFINED.

**PHỤ LỤC ..... 26**

# 1. GIỚI THIỆU CHUNG

## 1.1 Đặt vấn đề

Trong thời đại đầy thách thức và biến động hiện nay, vấn đề về chất lượng không khí đặt ra những thách thức lớn, đòi hỏi sự chú ý và giải pháp nhóm từ cả cộng đồng và các chuyên gia ngành môi trường. Ô nhiễm không khí không chỉ là một vấn đề sức khỏe cộng đồng mà còn là mối đe dọa đối với tính bền vững của môi trường sống.

Chất lượng không khí bị ảnh hưởng đáng kể bởi nhiều yếu tố, từ nguồn gốc tự nhiên như các sự kiện thời tiết đến các hoạt động con người như sản xuất, giao thông và năng lượng. Trong bối cảnh này, việc quan trắc và đánh giá chính xác chất lượng không khí trở nên thiết yếu, nhằm cung cấp cơ sở dữ liệu đầy đủ cho quá trình đưa ra quyết định và triển khai biện pháp can thiệp hiệu quả.

Hệ thống trạm quan trắc không khí không chỉ là công cụ quan trắc mà còn là bước quan trọng để hiểu rõ hơn về tác động của ô nhiễm không khí đối với cộng đồng. Bằng cách này, nó không chỉ đáp ứng nhu cầu thông tin hiện tại mà còn mở ra khả năng dự đoán, giúp chúng ta chuẩn bị trước cho những biến động trong chất lượng không khí và phát triển các chiến lược phòng ngừa.

Trong ngữ cảnh này, đề tài "Thiết kế Hệ thống Trạm Quan Trắc Không Khí và Đưa ra Dự Đoán" đặt ra nhiệm vụ không chỉ làm rõ hơn về chất lượng không khí hiện tại mà còn mở ra cánh cửa của sự hiểu biết và dự đoán. Mục tiêu của chúng tôi là tạo ra một hệ thống không chỉ quan sát mà còn dự đoán được, đóng góp tích cực vào việc xây dựng một môi trường sống an toàn và bền vững cho thế hệ tương lai.

## 1.2 Ý tưởng và mục tiêu hướng tới

Bắt đầu từ nhu cầu từ Đề tài "Thiết kế Hệ thống Trạm Quan Trắc Không Khí và Đưa ra Dự Đoán" đặt ra mục tiêu quan trọng là phát triển một hệ thống toàn diện không chỉ giám sát chất lượng không khí mà còn có khả năng dự đoán và đưa ra thông báo về biến động trong tương lai. Ý tưởng chính của chúng tôi tập trung vào các khía cạnh sau:

- **Tối ưu Hóa Hệ Thống Quan Trắc:** Chúng tôi đề xuất nâng cao hiệu suất của hệ thống trạm quan trắc không khí thông qua việc sử dụng các cảm biến và thiết bị đo hiện đại. Điều này bao gồm cả việc tối ưu hóa vị trí lắp đặt để đảm bảo sự đại diện chính xác cho môi trường cụ thể.
- **Phát Triển Mô Hình Dự Đoán:** Chúng tôi sẽ triển khai các mô hình máy học và trí tuệ nhân tạo để phân tích dữ liệu quan trắc và xây dựng mô hình dự đoán. Điều này giúp chúng ta hiểu rõ hơn về xu hướng, biến động và tác động của các yếu tố khác nhau đối với chất lượng không khí.
- **Tích Hợp Hệ Thống Thông Tin:** Chúng tôi đặt mục tiêu tích hợp hệ thống với các nền tảng thông tin và ứng dụng di động để cung cấp dữ liệu một cách linh hoạt và dễ tiếp cận cho cộng đồng. Điều này giúp tăng cường sự tham gia của cộng đồng và nhận thức về vấn đề ô nhiễm không khí.
- **Thực Hiện Hệ Thống Cảnh Báo:** Chúng tôi đề xuất phát triển hệ thống cảnh báo sớm, thông qua việc kết hợp dữ liệu quan trắc và mô hình dự đoán. Điều này sẽ giúp cảnh báo cộng đồng về những biến động tiềm ẩn trong chất lượng không khí, giúp họ chuẩn bị và ứng phó một cách nhanh chóng.

Qua ý tưởng này, chúng tôi hy vọng rằng đề tài của chúng tôi không chỉ giúp cải thiện chất lượng không khí mà còn đưa ra những giải pháp hiệu quả, hướng tới một môi trường sống bền vững và an toàn.

## 2. KẾ HOẠCH VÀ PHÂN CÔNG CÔNG VIỆC

### 2.1 Kế hoạch chi tiết

Để đạt được các mục tiêu đã nêu khi triển khai, dự án sẽ tuân theo một bộ kế hoạch toàn diện được thiết kế để giải quyết các khía cạnh khác nhau của quá trình phát triển. Các phương án được đề xuất như sau:

1. Điều tra, phân tích nhu cầu, đối tượng sử dụng:

- Xuất phát từ nhu cầu thực tiễn, đối tượng sử dụng, vị trí, ...
- Xây dựng các yêu cầu chức năng, phi chức năng

2. Tìm hiểu tài liệu:

• Tiến hành đánh giá chuyên sâu các tài liệu và tài liệu nghiên cứu hiện có liên quan đến việc triển khai hệ thống và ESP32.

• Khám phá các phương pháp, kiến trúc và tối ưu hóa khác nhau được sử dụng trong các dự án trước đó, cố gắng phát huy những ưu điểm và hạn chế nhược điểm.

3. Phân tích hệ thống:

- Lựa chọn cảm biến, module và các thiết bị phù hợp đề hướng tới mục tiêu của dự án.

4. Thiết kế:

- Mô phỏng, thử nghiệm kết nối chân giữa các thiết bị thông qua các phần mềm mô phỏng.
- Xây dựng mô hình, sơ đồ tuần tự, biểu đồ vận hành

5. Triển khai:

- Thực hiện code và kết nối cho phần cứng, kiểm thử hiệu suất của từng cảm biến.
- Hiệu chuẩn dữ liệu và áp dụng thuật toán.

6. Kiểm tra và đánh giá

- Đánh giá hiệu năng của toàn bộ hệ thống.

Bằng cách tuân theo các kế hoạch đề xuất này, dự án nhằm mục đích triển khai mô hình hệ thống quan trắc không khí sử dụng ESP32 được ghi chép đầy đủ, tối ưu hóa và đáng tin cậy, mở đường cho các dự án sau này trong các hệ thống tương tự.

## 3. CƠ SỞ LÝ THUYẾT

Trong phần này, chúng em sẽ tập trung tìm hiểu những giao thức, kỹ thuật và công nghệ để sử dụng trong đề tài. Nội dung sẽ gồm những giao thức, công nghệ chính có trong sản phẩm và khái niệm, các chức năng cơ bản của chúng.

### 3.1 Giao thức MQTT

MQTT (Message Queuing Telemetry Transport) là một giao thức truyền thông nhẹ, mở, và linh hoạt, được thiết kế đặc biệt để truyền thông dữ liệu giữa các thiết bị IoT (Internet of Things). Được phát triển bởi IBM vào những năm 1990 và sau đó trở thành một tiêu chuẩn mở với sự hỗ trợ của nhiều nhóm và tổ chức trong ngành công nghiệp IoT.

### 3.1.1. Nguyên Tắc Hoạt Động:

- MQTT là một giao thức publish/subscribe, trong đó các thiết bị có thể đăng ký để nhận thông điệp từ một chủ đề (topic) cụ thể và cũng có thể gửi thông điệp tới một chủ đề đó.
- Các thông điệp được chuyển đến qua một trung tâm thông điệp, thường được gọi là Broker. Broker đóng vai trò như một trung tâm điều phối, giúp thiết bị gửi và nhận thông điệp một cách hiệu quả.

### 3.1.2. Cấu Trúc Tin Nhắn:

- Tin nhắn MQTT bao gồm ba phần chính: Header, Payload, và Topic. Header chứa các thông tin quan trọng như kiểu tin nhắn, cấp độ chất lượng (QoS), và thông số khác. Payload là dữ liệu thực sự của tin nhắn, trong khi Topic là địa chỉ nơi mà tin nhắn được gửi hoặc nhận.

### 3.1.3. Chất Lượng Dịch Vụ (QoS):

MQTT hỗ trợ ba cấp độ chất lượng dịch vụ (QoS):

- QoS 0: Gửi một lần, không đảm bảo giao nhận.
- QoS 1: Gửi ít nhất một lần và đảm bảo ít nhất một lần nhận được.
- QoS 2: Gửi đúng một lần và đảm bảo nhận được đúng một lần.

### 3.1.4. Tiết Kiệm Băng Thông:

- MQTT được thiết kế để tiết kiệm băng thông. Giao thức này sử dụng kỹ thuật giữ kết nối (keep-alive) để duy trì kết nối mở khi không có dữ liệu truyền tải. Nó cũng hỗ trợ nén tin nhắn và sử dụng gói tin nhỏ để giảm lượng dữ liệu truyền tải.

### 3.1.5. Bảo Mật:

- MQTT hỗ trợ cơ chế bảo mật thông qua việc sử dụng SSL/TLS để mã hóa dữ liệu truyền tải giữa thiết bị và broker. Điều này đảm bảo an toàn cho dữ liệu, đặc biệt là khi sử dụng trong các ứng dụng IoT yêu cầu độ bảo mật cao.

### 3.1.6. Khả Năng Mở Rộng:

- MQTT có khả năng mở rộng tốt. Với cơ sở hạ tầng broker, nó có thể hỗ trợ hàng triệu thiết bị kết nối đồng thời.

Giao thức MQTT là một trong những công nghệ truyền thông quan trọng, mang lại tính linh hoạt và hiệu suất cao, đặc biệt là trong môi trường IoT đòi hỏi sự đơn giản và hiệu quả.

## 3.2 Giao tiếp UART

UART (Universal Asynchronous Receiver-Transmitter) là một giao thức truyền thông seri, được sử dụng để kết nối và truyền dữ liệu giữa các thiết bị trong các hệ thống nhúng và vi điều khiển. Dưới đây là một số khái niệm và nguyên tắc cơ bản của giao thức UART:

### 3.2.1. Đặc Điểm Cơ Bản:

- Sự Asynchronous (Bất đồng bộ): Trong UART, dữ liệu được truyền đi không có tín hiệu xung đồng hồ chung giữa các thiết bị. Thay vào đó, nó sử dụng các khung dữ liệu được chia thành các bit dữ liệu, bit start, bit stop, và thậm chí có thể chứa thêm bit kiểm tra (parity bit).
- Hình Thức Truyền:
  - Truyền 1 bit start: Bắt đầu mỗi khung truyền bằng một bit start để đánh dấu sự bắt đầu của dữ liệu.
  - Dữ liệu 8 bit (thường): Dữ liệu thường được truyền dưới dạng 8 bit, nhưng có thể có số bit khác tùy thuộc vào yêu cầu cụ thể.
  - 1 hoặc nhiều bit stop: Kết thúc mỗi khung bằng một hoặc nhiều bit stop để đảm bảo đồng bộ và tạo khoảng trống cho dữ liệu tiếp theo.

### 3.2.2. Tốc Độ Truyền (Baud Rate):

- Baud Rate: Là số lượng bit dữ liệu được truyền mỗi giây. Việc đồng bộ hóa giữa truyền và nhận dữ liệu yêu cầu cả hai bên đều cài đặt cùng một tốc độ baud.
- Baud Rate Tính Bằng Hz: Thường được tính bằng Hz, chẳng hạn 9600 bps (bits per second), 115200 bps, v.v.

### 3.2.3. Nguyên Tắc Hoạt Động:

- Truyền Dữ Liệu:
  - Mỗi khung dữ liệu bắt đầu bằng bit start để đánh dấu sự bắt đầu của dữ liệu.
  - Bit dữ liệu được truyền từ bit thấp đến bit cao.
  - Bit kiểm tra (parity bit) có thể được thêm vào để kiểm tra lỗi.
  - Một hoặc nhiều bit stop kết thúc mỗi khung dữ liệu.
- Nhận Dữ Liệu:
  - Thiết bị nhận sự bắt đầu của một khung bằng việc phát hiện bit start.
  - Đọc các bit dữ liệu theo thứ tự từ thấp đến cao.
  - Kiểm tra bit kiểm tra (nếu có).
  - Đọc các bit stop để xác nhận kết thúc khung dữ liệu.

### 3.2.4. Ưu Điểm và Nhược Điểm:

- Ưu Điểm:
  - Đơn giản và linh hoạt: UART là giao thức truyền thông đơn giản và phổ biến, có thể kết nối nhiều loại thiết bị khác nhau.
  - Không đòi hỏi đồng bộ hóa: Việc không yêu cầu xung đồng hồ chung giữa các thiết bị làm cho UART linh hoạt và dễ triển khai.
- Nhược Điểm:
  - Không Kiểm soát Lỗi Tốt: Vì không có xác nhận hoặc kiểm soát lỗi mạnh mẽ, UART có thể dễ bị nhiễu sóng và mất dữ liệu trong môi trường có nhiều tác nhân gây nhiễu.
  - Khả Năng Truyền Dữ Liệu Ngắn: Do sự không đồng bộ và thiếu các cơ chế kiểm soát lỗi mạnh mẽ, việc truyền dữ liệu trên khoảng cách lớn có thể gặp khó khăn.

Cơ sở lý thuyết về giao thức UART cung cấp cơ sở hành động để hiểu và triển khai giao thức này trong các ứng dụng truyền thông.

## 3.3 Giao tiếp I2C

Giao tiếp qua chuẩn I2C (Inter-Integrated Circuit) là một phương pháp hiệu quả và linh hoạt được sử dụng rộng rãi trong việc kết nối giữa các vi điều khiển và các thiết bị ngoại vi trong các hệ thống điện tử. I2C giúp đơn giản hóa quá trình truyền thông dữ liệu giữa các thành phần, đặc biệt là trong các ứng dụng nhúng và IoT.

Giao tiếp I2C sử dụng hai dây truyền thông chính: dây dữ liệu (SDA) và dây đồng hồ (SCL). Điều này cho phép nhiều thiết bị được kết nối trên cùng một bus, với mỗi thiết bị có một địa chỉ duy nhất. Quy trình truyền dữ liệu bao gồm các tín hiệu START và STOP, cùng với các khung dữ liệu chứa thông tin từ master đến slave hoặc ngược lại.

Giao tiếp I2C được ưa chuộng bởi sự linh hoạt và khả năng kết nối nhiều thiết bị trên cùng một bus, giảm thiểu việc sử dụng chân đầu ra trên vi điều khiển. Tốc độ truyền dữ liệu có thể linh hoạt điều chỉnh tùy thuộc vào yêu cầu của hệ thống. Việc hiểu rõ về giao tiếp I2C là quan trọng để tích hợp hiệu quả các thiết bị và mô-đun vào hệ thống của họ, đồng thời tối ưu hóa hiệu suất và khả năng mở rộng của hệ thống.

## 3.4 Hệ điều hành RTOS

RTOS (Real-Time Operating System) là hệ điều hành được thiết kế đặc biệt để hỗ trợ và quản lý ứng dụng yêu cầu thời gian thực, nơi việc xử lý và phản hồi phải được thực hiện trong một khoảng thời gian cố định. Đặc điểm chính của RTOS bao gồm đáp ứng thời gian cố định, quản lý ưu tiên, và quản lý tài nguyên. Scheduler



được sử dụng để quản lý việc lập lịch thực hiện nhiệm vụ, trong khi Semaphore và Mutex được sử dụng để đồng bộ hóa truy cập vào tài nguyên chia sẻ. Interrupts được xử lý một cách hiệu quả, và Message Queues được sử dụng để truyền thông tin giữa các nhiệm vụ. Hệ điều hành thời gian thực cũng phải đối mặt với các vấn đề như deadlock và starvation. Kiến trúc của RTOS có thể là Monolithic Kernel hoặc Microkernel, tùy thuộc vào cách mà các dịch vụ cơ bản của hệ điều hành được tổ chức.

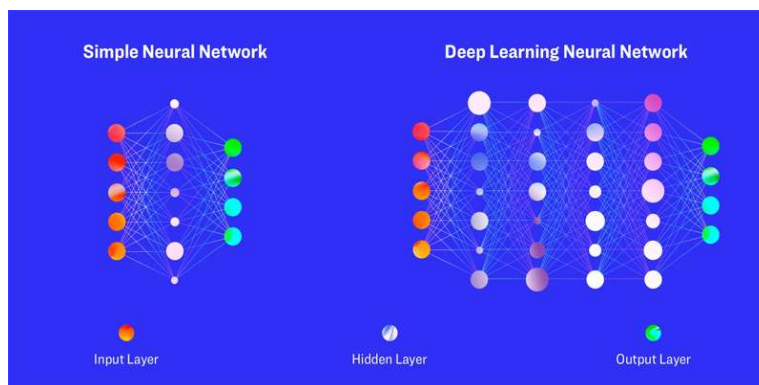
### 3.5 Mô hình mạng LSTM:

#### 3.5.1. Khái niệm Deep Learning:

Deep Learning là định nghĩa thuộc về một phần các thuật toán trong Machine Learning (máy học) với đặc thù mang độ phức tạp cao hơn. Vì vậy có thể nói hai khái niệm giữa Deep Learning và Machine Learning hoàn toàn có liên hệ mật thiết với nhau.

Trong thực tế, Deep Learning bao gồm nhiều lớp ẩn trong một mạng lưới thần kinh và thuộc lớp sau cùng. Việc đi qua nhiều số lượng lớp và mạng phức tạp được cho là độ sâu.

Ngày nay, sự thay đổi lớn nhất trong học tập sâu là độ sâu của mạng lưới thần kinh đã phát triển từ một vài lớp đến hàng trăm trong số chúng. Độ sâu hơn có nghĩa là khả năng nhận dạng các mẫu lớn hơn, với nguồn thông tin lớn hơn giúp tăng khả năng tiếp nhận các đối tượng trở nên rộng hơn, chi tiết hơn.



Hình 3-1: Mạng nơ ron đơn và mạng nơ ron học sâu

Cách thức hoạt động của thuật toán Deep Learning diễn ra như sau:

Các dòng thông tin sẽ được trải qua nhiều lớp cho đến lớp sau cùng. Lấy quy trình học của con người làm ví dụ cụ thể. Qua các lớp đầu tiên sẽ tập trung vào việc học các khái niệm cụ thể hơn trong khi các lớp sâu hơn sẽ sử dụng thông tin đã học để nghiên cứu và phân tích sâu hơn trong các khái niệm trừu tượng. Quy trình xây dựng biểu diễn dữ liệu này được gọi là trích xuất tính năng.

Kiến trúc phức tạp của việc học sâu được cung cấp từ mạng lưới thần kinh sâu với khả năng thực hiện trích xuất tính năng tự động. Ngược lại, trong học máy thông thường còn gọi là học nông, nhiệm vụ này được thực hiện khi truy xuất các thuật toán cụ thể.

#### 3.5.2. Tiền xử lý dữ liệu trong Deep Learning

Vì các thông số có dải khác nhau giá trị khác nhau, các thông số có giá trị lớn hơn sẽ có ảnh hưởng lớn hơn đến kết quả dự đoán của mô hình. Chuẩn hóa dữ liệu giúp cân bằng trọng số các thông số. Để phù hợp hơn cho việc đưa dữ liệu vào mô hình. Một số các chuẩn hóa để xử lý dữ liệu thông dụng như: chuẩn hóa Min Max, chuẩn hóa trung bình, bình thường hóa. Sau đây là ví dụ chuẩn hóa Min Max:

Áp dụng chuẩn hóa Min Max cho từng thông số bằng công thức:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3.5.2)$$

Trong đó:

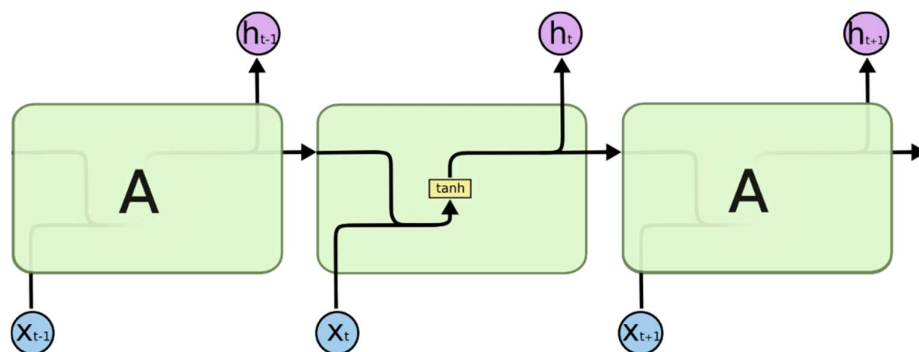
- $x'$  là giá trị được chuẩn hóa, nằm trong  $[0,1]$ .
- $x$  là giá trị ban đầu của một thông số trong tập dữ liệu
- $x_{\min}, x_{\max}$  lần lượt là giá trị nhỏ nhất và giá trị lớn nhất của từng thông số.

### 3.5.3. Tổng quan LSTM

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

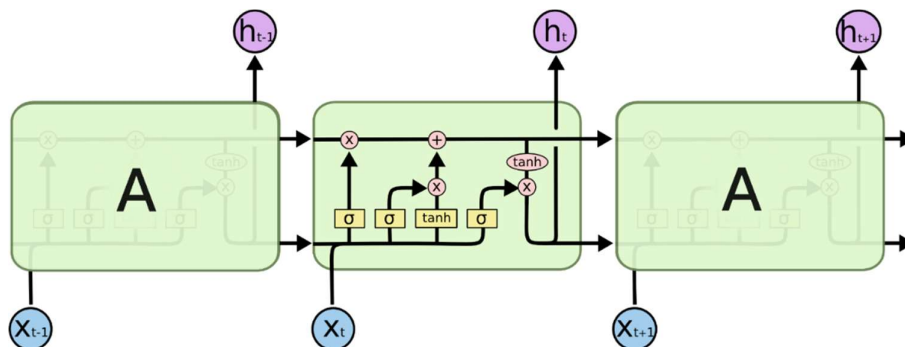
LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-đun này có cấu trúc rất đơn giản, thường là một tầng tanh.



Hình 3-2: Cấu trúc mạng RNN

LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.



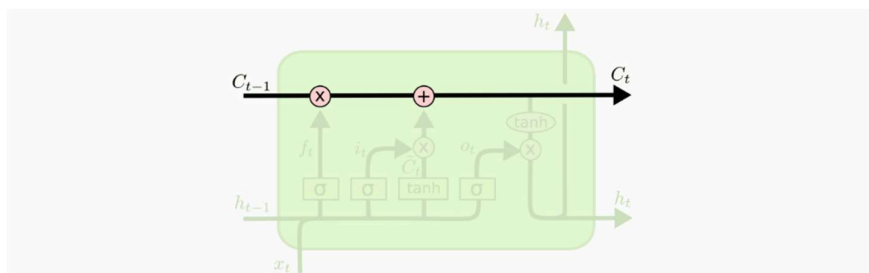
Hình 3-3: Cấu trúc mạng LSTM

Ở sơ đồ trên, mỗi một đường mang một véc-tơ từ đầu ra của một nút tới đầu vào của một nút khác. Các hình trong màu hồng biểu diễn các phép toán như phép cộng véc-tơ chẳng hạn, còn các ô màu vàng được sử dụng để biểu diễn các phép toán khác.

dụng để học trong các từng mạng nơ-ron. Các đường hợp nhau kí hiệu việc kết hợp, còn các đường rẽ nhánh ám chỉ nội dung của nó được sao chép và chuyển tới các nơi khác nhau.

Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ.

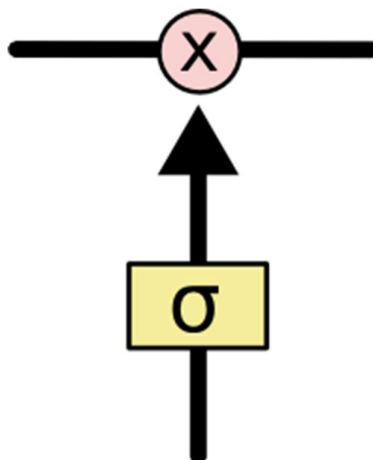
Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.



Hình 3-4: Đường trạng thái tế bào trong LSTM

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế bào, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).

Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.



Hình 3-5: Tầng sigmoid và phép nhân

Tầng sigmoid sẽ cho đầu ra là một số trong  $[0,1]$ , mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 1 thì có nghĩa là cho tất cả các thông tin đi qua nó. Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

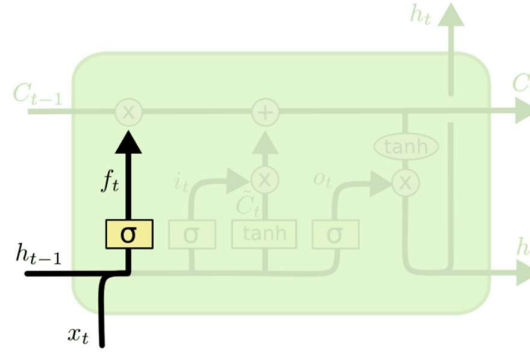
### 3.5.4. Kiến trúc LSTM

#### 3.5.4.1. Cổng quên (Forget Gate)

Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là “tầng cổng quên” (forget gate layer). Nó sẽ lấy đầu vào là  $h_{t-1}$  và  $x_t$  rồi đưa ra kết quả là một số trong  $[0,1]$  cho mỗi số trong trạng thái tế bào  $C_{t-1}$ . Đầu ra là 1 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 0 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi.

Ví dụ một mô hình ngôn ngữ dự đoán từ tiếp theo dựa trên tất cả các từ trước đó, với những bài toán như vậy, thì trạng thái tế bào có thể sẽ mang thông tin về giới tính của một nhân vật nào đó giúp ta

sử dụng được đại từ nhân xưng chuẩn xác. Tuy nhiên, khi đề cập tới một người khác thì ta sẽ không muốn nhớ tới giới tính của nhân vật nữa, vì nó không còn tác dụng gì với chủ thể mới này.



Hình 3-6: Cổng quên

Công thức tính trọng số quyết định loại bỏ thông tin cũ:

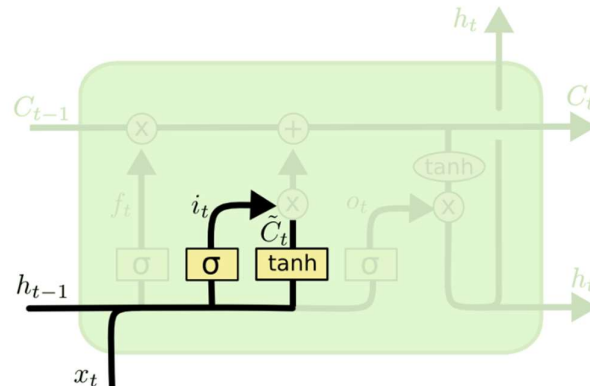
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Trong đó:

- $f_t$  là forget gate (cổng quên) tại thời điểm  $t$
- $\sigma$  là hàm sigmoid, được sử dụng để chuyển đổi giá trị thành một giá trị nằm trong khoảng từ 0 đến 1.
- $W_f$  là ma trận trọng số liên kết với forget gate.
- $h_{t-1}$  là trạng thái ẩn (hidden state) tại thời điểm trước đó ( $t-1$ ).
- $x_t$  là đầu vào tại thời điểm  $t$ .
- $b_f$  là véc-tơ độ dời (bias) liên quan đến forget gate.

#### 3.5.4.2. Cổng vào (Input gate)

Bước tiếp theo là quyết định xem thông tin mới nào ta sẽ lưu vào trạng thái tế bào. Việc này gồm 2 phần. Đầu tiên là sử dụng một tầng sigmoid được gọi là “tầng cổng vào” (input gate layer) để quyết định giá trị nào ta sẽ cập nhật. Tiếp theo là một tầng tanh tạo ra một véc-tơ cho giá trị mới  $\tilde{C}_t$  nhằm thêm vào cho trạng thái. Trong bước tiếp theo, ta sẽ kết hợp 2 giá trị đó lại để tạo ra một cặp nhập cho trạng thái. Chẳng hạn với ví dụ mô hình ngôn ngữ của ta, ta sẽ muốn thêm giới tính của nhân vật mới này vào trạng thái tế bào và thay thế giới tính của nhân vật trước đó.



Hình 3-7: Cổng vào (Input gate)

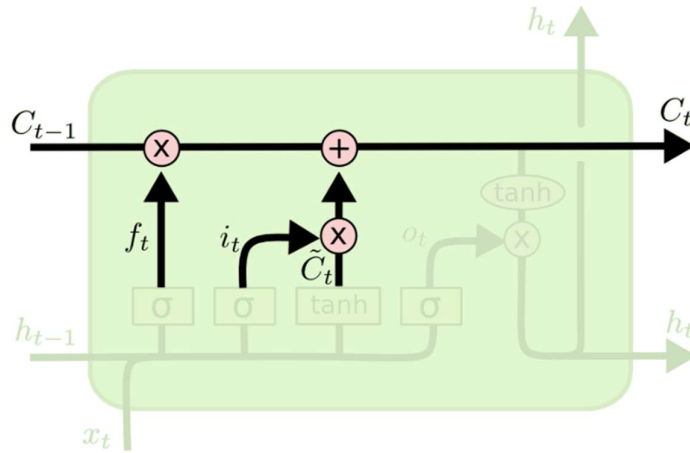
Công thức quyết định lưu thông tin mới nào sẽ được lưu vào trạng thái tế bào của công vào được mô tả qua 2 công thức:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Giờ là lúc cập nhật trạng thái tế bào cũ  $C_{t-1}$  thành trạng thái mới  $C_t$ . Ở các bước trước đó đã quyết định những việc cần làm, nên giờ ta chỉ cần thực hiện là xong. Ta sẽ nhân trạng thái cũ với  $f_t$  để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm  $i_t * \tilde{C}_t$ . Trạng thái mới thu được này phụ thuộc vào việc ta quyết định cập nhật mỗi giá trị trạng thái ra sao.

Với bài toán mô hình ngôn ngữ, chính là việc ta bỏ đi thông tin về giới tính của nhân vật cũ, và thêm thông tin về giới tính của nhân vật mới như ta đã quyết định ở các bước trước đó.



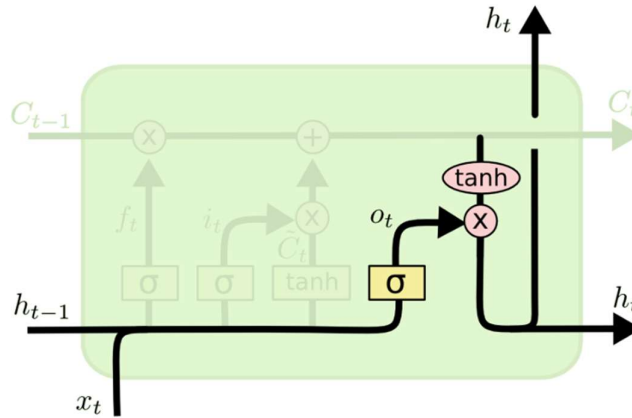
Hình 3-8: Cập nhật trạng thái mới

Công thức mô tả cho hình Hình 17, cập nhật trạng thái mới:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

#### 3.5.4.3. Cổng ra (Output Gate)

Cuối cùng, ta cần quyết định xem ta muốn đầu ra là gì. Giá trị đầu ra sẽ dựa vào trạng thái tế bào, nhưng sẽ được tiếp tục sàng lọc. Đầu tiên, ta chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào ta muốn xuất ra. Sau đó, ta đưa nó trạng thái tế bào qua một hàm tanh để có giá trị nó về  $[-1, 1]$ , và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn. Với ví dụ về mô hình ngôn ngữ, chỉ cần xem chủ thể mà ta có thể đưa ra thông tin về một trạng từ đi sau đó. Ví dụ, nếu đầu ra của chủ thể là số ít hoặc số nhiều thì ta có thể biết được dạng của trạng từ đi theo sau nó phải như thế nào.



Hình 3-9: Cổng ra

Đầu ra sẽ được thực hiện lần lượt theo hai bước ứng với công thức:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

### 3.5.5. Ứng dụng LSTM

LSTM sinh ra xử lý dữ liệu dạng chuỗi có khả năng biến đổi và xử lý thông tin lâu dài. LSTM có rất nhiều ứng dụng một số ví dụ như:

- Dự báo chuỗi thời gian: LSTM được sử dụng để dự báo chuỗi thời gian như dự báo thời tiết, dự báo giá cổ phiếu, dự báo lưu lượng nước trên một dòng sông dựa vào chuỗi dữ liệu có từ quá khứ.
- Tự động sinh văn bản: LSTM có thể sử dụng để tạo ra văn bản tự động từ văn bản trước đó, sử dụng một lượng lớn tập dữ liệu là các văn bản, sau đó tạo ra văn bản mới dựa vào văn bản hiện có.
- Dịch máy: LSTM trong dịch máy có khả năng hiểu ngôn ngữ tự nhiên. Sử dụng đầu vào bằng cách chia văn bản thành các từ riêng lẻ rồi tạo ra các dự đoán cho từ tiếp theo.

## 3.6 Tổng quan về AQI:

### 3.6.1. Khái niệm:

Chỉ số chất lượng không khí (viết tắt là AQI) là chỉ số được tính toán từ các thông số quan trắc các chất ô nhiễm trong không khí, nhằm cho biết tình trạng chất lượng không khí và mức độ ảnh hưởng đến sức khỏe con người. AQI tập trung vào các vấn đề sức khỏe mà chúng ta có thể gặp phải trong vài giờ hoặc vài ngày sau khi hít phải khí ô nhiễm. Thông thường AQI được tính toán với các yếu tố: NO<sub>2</sub>, SO<sub>2</sub>, O<sub>3</sub>, CO, PM<sub>10</sub>, PM<sub>2.5</sub>.

### 3.6.2. Tính toán AQI:

AQI được tính toán theo từng thành phần chất ô nhiễm trong không khí. Mỗi thông số sẽ xác định được một giá trị AQI cụ thể, giá trị AQI cuối cùng là giá trị lớn nhất trong các giá trị AQI của mỗi thông số (ở đây không dùng phương pháp tính giá trị trung bình vì chỉ cần có một thông số vượt quá ngưỡng cho phép là có thể kết luận môi trường đã bị ô nhiễm và có ảnh hưởng đến sức khỏe của cộng đồng).

Hiện nay trên thế giới rất nhiều quốc gia đã xây dựng phương pháp tính toán và công bố AQI. Phương pháp tính toán AQI khá đa dạng, tuy nhiên các phương pháp đều được tính toán dựa trên nồng

độ các khí gây ô nhiễm như: O<sub>3</sub>, CO, NO<sub>2</sub>, nồng độ bụi: TSP, PM<sub>10</sub>, PM<sub>2.5</sub> được lấy trung bình trong khoảng 1 giờ và 8 giờ hoặc 1 giờ – 24 giờ.

Có thể chia các phương pháp tính toán AQI thành 3 nhóm cơ bản sau:

- Sử dụng các bảng thông số đối chiếu (Anh, Pháp, Canada)
- Sử dụng các công thức tính toán đơn giản (Australia, Thành phố Hồ Chí Minh)
- Sử dụng các công thức tính toán phức tạp (Mỹ, Braxin, Hồng Kông, Hàn Quốc, Thái Lan, Bồ Đào Nha)

### 3.6.3. AQI tại nước ta:

Tại Việt Nam phương pháp tính toán AQI do Tổng cục môi trường ban hành được phát triển dựa theo phương pháp thứ 2: sử dụng các công thức đơn giản. Đồng thời việc tính toán AQI phải đảm bảo các yêu cầu sau:

- AQI được tính toán riêng cho số liệu của từng trạm quan trắc không khí tự động cố định liên tục đối với môi trường không khí xung quanh;
- AQI được tính toán cho từng thông số quan trắc. Mỗi thông số sẽ xác định được một giá trị AQI cụ thể. Giá trị AQI cuối cùng là giá trị lớn nhất trong các giá trị AQI của mỗi thông số;
- Thang đo giá trị AQI được chia thành các khoảng nhất định. Khi giá trị AQI nằm trong một khoảng nào đó, thì thông điệp cảnh báo cho cộng đồng ứng với khoảng giá trị đó sẽ được đưa ra.
- Các thông số thường được sử dụng là các thông số trong QCVN 05:2009/BTNMT bao gồm: SO<sub>2</sub>, CO, NO<sub>x</sub>, O<sub>3</sub>, PM<sub>10</sub>, TSP. Số liệu quan trắc được đưa vào tính toán đã qua xử lý, đảm bảo đã loại bỏ các giá trị sai lệch, đạt yêu cầu đối với quy trình quy phạm về đảm bảo kiểm soát chất lượng số liệu.

Quy trình tính toán AQI bao gồm:

- Tính toán giá trị AQI theo giờ theo công thức:

$$AQI_x^h = \frac{TS_x}{QC_x} \times 100$$

$TS_x$ : Giá trị quan trắc trung bình mỗi giờ của chất X

$QC_x$ : Giá trị quy chuẩn trung bình mỗi giờ của chất X

Lưu ý: Đối với thông số PM<sub>10</sub>: do không có quy chuẩn trung bình 1 giờ, vì vậy lấy quy chuẩn của TSP trung bình 1 giờ thay thế cho PM<sub>10</sub>.

- Giá trị AQI theo giờ là giá trị lớn nhất của các thông số trong cùng một thời gian:

$$AQI_x^h = \max AQI_x^h$$

- Tính toán giá trị AQI theo ngày: Tính toán giá trị AQI theo ngày của mỗi thông số theo công thức sau:

$$AQI_x^{24h} = \frac{TS_x}{QC_x} \times 100$$

$TS_x$ : Giá trị quan trắc trung bình 24 giờ của chất X

$QC_x$ : Giá trị quy chuẩn trung bình 24 giờ của chất X

Giá trị AQI làm tròn thành số nguyên.

Giá trị AQI theo ngày của từng thông số được xác định là giá trị lớn nhất trong số các giá trị AQI theo giờ của thông số đó trong 01 ngày và giá trị AQI trung bình 24 giờ của thông số đó.

$$AQI_x^d = \max(AQI_x^{24h}, AQI_x^h)$$

Trong đó,

$AQI_x^d$  là giá trị AQI theo ngày của thông số X

Giá trị AQI theo ngày là giá trị AQI lớn nhất của các thông số được lấy theo ngày của trạm quan trắc đó.

$$AQI_x^d = \max AQI_x^d$$

So sánh AQI tính được với bảng công bố thông tin về AQI cho cộng đồng:

Khoảng giá trị AQI	Chất lượng không khí	Ảnh hưởng sức khỏe	Màu
0 – 50	Tốt	Không ảnh hưởng đến sức khỏe	Xanh
51 – 100	Trung bình	Nhóm nhạy cảm nên hạn chế thời gian ở bên ngoài	Vàng
101 – 200	Kém	Nhóm nhạy cảm cần hạn chế thời gian ở bên ngoài	Da cam
201 – 300	Xấu	Nhóm nhạy cảm tránh ra ngoài. Những người khác hạn chế ở bên ngoài	Đỏ
Trên 300	Nguy hại	Mọi người nên ở trong nhà	Nâu

Bảng 3-1: Bảng giá trị AQI tương ứng với màu sắc

## 4. YÊU CẦU THIẾT KẾ

Ở phần này, nhóm tập trung tìm hiểu các mô hình sản phẩm đã có trên thị trường, phân tích điểm mạnh, điểm yếu của những sản phẩm này. Từ đó đưa ra những yêu cầu cho thiết kế của nhóm.

### 4.1 Yêu cầu chức năng sản phẩm của nhóm

Sau khi phân tích các ưu, nhược điểm của những sản phẩm đã có trên thị trường. Nhóm quyết định xây dựng sản phẩm với các đặc tính ưu điểm của các sản phẩm hiện có và cố gắng tối đa hạn chế các nhược điểm.

Với vai trò là một hệ thống quan trắc không khí, thiết bị của nhóm cần phải đáp ứng được các yêu cầu sau:

- Hệ thống phát hiện nhanh chóng, kịp thời các thông số liên quan: Nhiệt độ, độ ẩm, PM2.5, PM10
- Có màn hình LCD để hiển thị các dữ liệu, đồng thời gửi dữ liệu lên server.
- Sử dụng thuật toán, mô hình đưa ra dự đoán về thông số môi trường trong tương lai.
- Mạch có jack DC để cấp nguồn và pin dự phòng



## 4.2 Yêu cầu phi chức năng sản phẩm của nhóm

Để hệ thống tối ưu nhất và có thể tiếp cận với người dùng nhanh chóng, đơn giản, cần đáp ứng những yêu cầu sau:

- Sản phẩm có kích thước nhỏ gọn, thiết bị sẽ được lắp đặt linh hoạt hơn
- Hoạt động ổn định trong khoảng nhiệt độ từ  $10 \div 100^{\circ}\text{C}$
- Sản phẩm tiết kiệm điện, công suất hoạt động nhỏ.
- Nồng độ PM2.5, PM10 có thể đo từ  $0 \div 999 \mu\text{g}/\text{m}^3$
- Chu kỳ thu thập data khoảng 10 giây
- Cập nhật dữ liệu lên hệ thống liên tục 30 phút một lần
- Giao diện web hiển thị dữ liệu rõ ràng, bố cục đơn giản

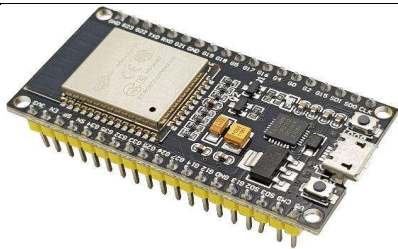
## 5. THIẾT KẾ HỆ THỐNG

### 5.1 Phân tích, lựa chọn linh kiện

#### 5.1.1. Khối điều khiển: Module ESP32-WROOM-32:

ESP32 là một vi điều khiển tích hợp mạnh mẽ được sử dụng phổ biến trong các ứng dụng IoT. Nó kết hợp Wi-Fi và Bluetooth, có tốc độ xử lý cao, khả năng tiết kiệm năng lượng, và hỗ trợ nhiều giao tiếp như UART, SPI, I2C. ESP32 có thể lập trình bằng Arduino IDE, MicroPython và ESP-IDF. Đây là một nền tảng linh hoạt và mạnh mẽ cho các dự án IoT. ESP32 là sản phẩm kế thừa từ vi điều khiển ESP8266.

Ta có thể lập trình bằng nhiều ngôn ngữ khác nhau như C/C++, Python, NodeJs, Lua, Ngoài ra, hiện tại Espressif Systems hỗ trợ chính thức SDK cho ESP32 với Arduino IDE, vì vậy ta có thể yên tâm xây dựng các ứng dụng với Arduino ESP32.

Thông số	Module ESP32-WROOM-32
Hình ảnh	
MCU	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
Điện áp nguồn	5V DC
Đầu vào/ra điện áp	3.3V DC
Băng tần	80 - 240 MHz
Giao thức WiFi	802.11B/g/n/E/I (802.11N @ 2.4 GHz lên đến 150 Mbit/s)
Tần số WiFi	2.4 – 2.5 GHz

Bluetooth	V4.2 - BLE
Bộ nhớ	448 Kbyte ROM, 520 Kbyte SRAM, 6 Kbyte SRAM trên RTC và QSPI Hỗ trợ đèn flash / SRAM chip
Chip USB-Serial	CP2102
Anten không dây	PCB
Giá thành	120.000 VNĐ

Bảng 5-1: Thông số module ESP32-WROOM-32

### 5.1.2. Khởi hiển thị: Màn hình LCD 2004 – I2C



Hình 5-1: LCD 2004 và module I2C PCF8574

Nhằm phục vụ việc theo dõi các hoạt động thu thập dữ liệu từ các cảm biến, trong đề tài em sử dụng màn hình LCD2004 để hiển thị dữ liệu. Với việc hỗ trợ Giao tiếp I2C từ module PCF8574, tối ưu chân cắm cho vi điều khiển hơn.

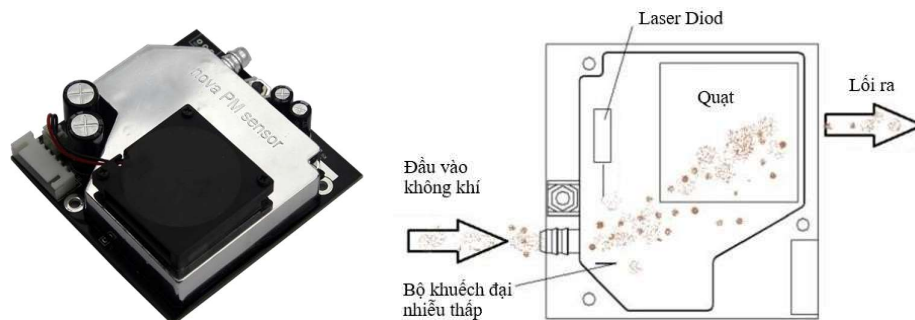
Đặc điểm	Thông số
Điện áp hoạt động (V)	5
Màu chữ, nền	Đen, xanh lục / Trắng, xanh dương
Điều chỉnh độ tương phản	Biến trở hoặc xung PWM
Khả năng hiển thị	(20 kí tự / dòng) x 4 dòng
Chip điều khiển	HD77480
Giao tiếp sử dụng	I2C (Sử dụng mạch chuyển đổi PCF8574)

Bảng 5-2: Thông số kỹ thuật LCD2004

### 5.1.3. Khởi cảm biến

#### 5.1.4. Cảm biến đo bụi Nova SDS011:

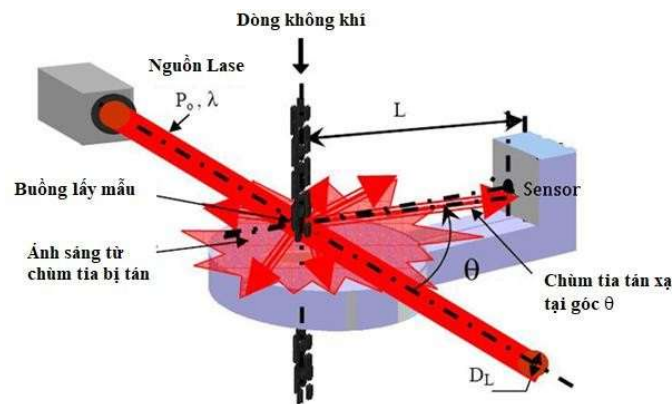
Cảm biến SDS011 là cảm đo nồng độ bụi sử dụng công nghệ nhiễu xạ laser được phát triển gần đây bởi inovafit, một công ty con từ trường đại học Tế Nam (Sơn Đông). Đây có thể coi là cảm biến đo bụi có độ chính xác cao trong các dòng cảm biến đo bụi kích thước nhỏ. Trong khi các cảm biến khác có xu hướng tập trung vào thu hẹp kích thước cảm biến, SDS011 lựa chọn giải pháp cân bằng giữa hiệu năng và kích thước trang bị thêm một quạt hút để tạo dòng khí đối lưu cho cảm biến, khác với các dòng cảm biến tương tự khác như Shinyei PPD24NS (một cảm biến từ Nhật Bản) sử dụng nhiệt điện trở để tạo dòng đối lưu tự nhiên.



Nguyên lý đo bụi của SDS011:

Cảm biến đo bụi sử dụng công nghệ nhiễu xạ laser (tán xạ ánh sáng). Nguyên tắc của phương pháp này như sau: khi một chùm laser đi qua không khí không chứa bụi, ánh sáng từ chùm tia không bị tán xạ. Khi trong không khí có bụi ánh sáng từ chùm tia sẽ bị tán xạ ra xung quanh. Ánh sáng tán xạ được thu nhận bởi đầu thu và chuyển thành tín hiệu điện và các tín hiệu này sẽ được khuếch đại và xử lý. Số lượng và đường kính của các hạt có thể thu được bằng cách phân tích tín hiệu bởi vì dạng sóng tín hiệu có quan hệ nhất định với đường kính hạt.

Cảm biến đo bụi như vậy sử dụng nguồn phát quang hồng ngoại gần (diode laser). Đầu thu là một diod quang kiểu thác với bộ khuếch đại. Nguồn phát hồng ngoại được sử dụng trong trường hợp này để tránh nhiễu với ánh sáng ban ngày vào buồng.



Hình 5-3: Minh họa đo bụi theo phương pháp tán xạ

Mật độ bụi phụ thuộc chủ yếu vào lưu lượng không khí. Nguồn laser, cảm biến và ống kính chuẩn trực được đặt ở vị trí bên trên để ngăn chặn sự lắng đọng của bụi lên bề mặt khi dòng khí không lưu thông qua buồng đo. (Bên trong buồng lấy mẫu có một nhiệt điện trở đặt trước cảm biến được sử dụng để sưởi ấm dòng khí tạo ra một dòng chảy liên tục qua cảm biến).

Góc tối ưu giữa nguồn Laser và cảm biến được tối ưu từ kết quả thực nghiệm. Mỗi hạt đi qua chùm tia laser khuếch tán một phần của chùm tia này tới cảm biến và lưu lượng không khí là không đổi, độ rộng xung của tín hiệu từ cảm biến được sử dụng để phân loại các hạt theo kích thước. Số lượng trung bình các hạt được thu thập và hiển thị với chu kỳ khoảng 30s.

Các hạt có kích thước lớn hơn 10μm mặc dù có hiện diện trong không khí nhưng không gây hại tới sức khỏe nên được bỏ qua.

#### 5.1.5. Cảm biến nhiệt độ và độ ẩm SHT31:

Cảm biến nhiệt độ và độ ẩm SHT31-d là một thiết bị hiệu suất cao được sử dụng để đo và giám sát nhiệt độ cùng độ ẩm trong môi trường. Thiết kế và chất lượng của cảm biến này đảm bảo tính chính xác và độ tin cậy trong việc cung cấp thông tin về điều kiện môi trường.

Nguyên tắc hoạt động của cảm biến nhiệt độ và độ ẩm SHT31-D dựa trên nguyên tắc hoạt động của cảm biến nhiệt độ và độ ẩm dựa vào sự thay đổi điện trở của chất hấp thụ nước trong môi trường xung quanh. Cụ thể, cảm biến SHT31 sử dụng hai cảm biến chất hấp thụ nhạy cảm đến nhiệt độ và độ ẩm để thực hiện đo lường.

- Cảm biến Nhiệt độ: Cảm biến nhiệt độ của SHT31-D là một cảm biến nhiệt trở (thermistor) chất lỏng. Điện trở của cảm biến thay đổi theo nhiệt độ. Khi nhiệt độ tăng lên, điện trở của cảm biến cũng tăng lên. Sự thay đổi này được đo lường và chuyển đổi thành giá trị nhiệt độ tương ứng.

- **Cảm biến Độ ẩm:** Cảm biến độ ẩm của SHT31-D sử dụng một lớp chất hấp thụ dựa trên polyme polymer có khả năng hấp thụ và giải phóng hơi nước tùy theo độ ẩm môi trường. Khi độ ẩm tăng, chất hấp thụ nước sẽ hấp thụ hơi nước từ môi trường và khi độ ẩm giảm, chất hấp thụ nước sẽ giải phóng hơi nước. Sự thay đổi về khối lượng của chất hấp thụ nước này tạo ra sự thay đổi về điện trở, từ đó cảm biến đo lường được độ ẩm.

Cảm biến SHT31-D kết hợp thông tin từ cảm biến nhiệt độ và độ ẩm để cung cấp dữ liệu nhiệt độ và độ ẩm của môi trường. Thông qua việc đo lường và so sánh sự thay đổi điện trở của cảm biến, nó có khả năng cung cấp dữ liệu chính xác về nhiệt độ và độ ẩm môi trường trong thời gian thực. Các giá trị này thường được truyền qua giao thức I2C hoặc SPI để được sử dụng và xử lý trong các ứng dụng và hệ thống khác nhau.



Hình 5-4: Cảm biến SHT31

#### 5.1.6. Khối thời gian thực: Module RTC DS3231

Khối thời gian thực sử dụng DS3231 dùng để căn chỉnh thời gian gửi và lưu trữ dữ liệu định kỳ, mạch cảm biến có thể bị mất điện khi không có nguồn điện cấp. Tuy nhiên DS3231 sử dụng pin lưu trữ CR2032 khiến cho việc không cấp nguồn điện cho toàn mạch nhưng vẫn lưu trữ thời gian, đảm bảo khi cấp nguồn lại vẫn có thể gửi và lưu trữ dữ liệu định kỳ theo thời gian mong muốn. Bảng 5-3 Mô tả thông số kỹ thuật cơ bản của DS3231.

Đặc điểm	Thông số
Điện áp hoạt động (V)	3.3 – 5.5
Pin tích hợp dự phòng	CR2032
Nhiệt độ hoạt động (°C)	-40 - 85
Nhiệt độ hoạt động chính xác cao (°C)	0 - 40
Thời gian	Giờ, phút, giây, ngày, tháng, năm

Bảng 5-3: Thông số kỹ thuật DS3231



Hình 5-5: Module RTC 3231

#### 5.1.7. Khối lưu trữ: Module SD Card SPI



Hình 5-6: Module thẻ nhớ SD Card SPI

Module thẻ nhớ micro-SD sử dụng để lưu trữ dữ liệu khi hệ thống không thể gửi dữ liệu qua MQTT do một số vấn đề trực trực nào đó. Sử dụng thẻ micro-SD để lưu trữ và có thể lấy ra và sao chép dữ liệu khi module truyền dữ liệu lên internet gặp vấn đề.

Đặc điểm	Thông số
Điện áp hoạt động (V)	4.5 -5.5
Dòng sử dụng (mA)	0.2 - 200
Hỗ trợ	Hệ thống tệp FAT, thẻ nhớ Micro SD lên đến 2GB, SDHC lên đến 32GB

Bảng 5-4: Thông số kỹ thuật module thẻ nhớ

### 5.1.8. Khối nguồn: Module chỉnh áp LM2596 và module ổn áp AMS1117

Như đã đề cập ở trên, cảm biến Nova SDS011 và màn hình LCD 20x4 chỉ hoạt động ổn định và chính xác ở mức điện áp 5V+, mà trên vi điều khiển ESP32 không có chân 5V. Đồng thời, để cấp nguồn cho ESP32, bộ thiết bị này có khối nguồn sử dụng module chỉnh áp LM2596, được cấp nguồn bởi bộ Jack DC 5.5x2.1mm. Bên cạnh đó, việc kết hợp với module ổn áp AMS1117 giúp chuyển đổi điện áp đầu ra 5V từ LM2596 về mức 3.3 VDC dùng để cung cấp cho cảm biến SHT31.

Module chỉnh áp LM2596 là một thiết bị điện tử được sử dụng để điều chỉnh và ổn định điện áp đầu ra từ một nguồn cung cấp điện áp đầu vào có thể biến đổi. Được dựa trên vi mạch điều chỉnh áp đa mức (buck), module này giúp biến đổi lên từ mức cao hơn xuống mức thấp hơn một cách hiệu quả.



Hình 5-7: Module chỉnh áp LM2596



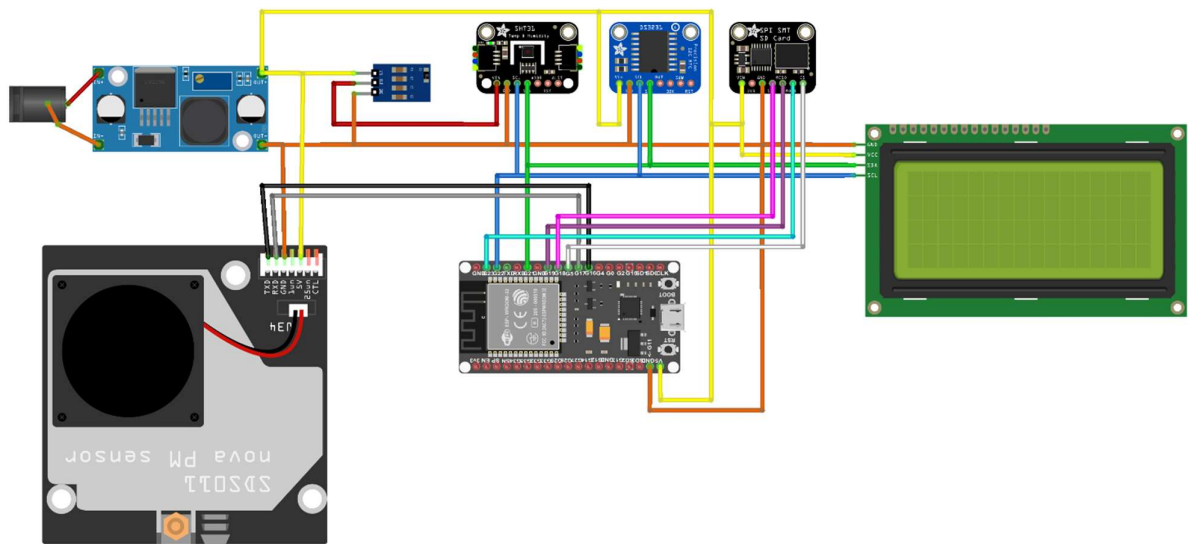
Hình 5-8: Module AMS1117

Thông số	AMS1117	LM2596
Điện áp ngõ ra	3.3-3.6V	1.5-20V
Dòng điện ngõ ra	1-1.9 A	3A(max)
Điện áp ngõ vào	15 - 15.9V	3-30V

Hình 5-5: Thông số kỹ thuật module AMS1117 và LM2596

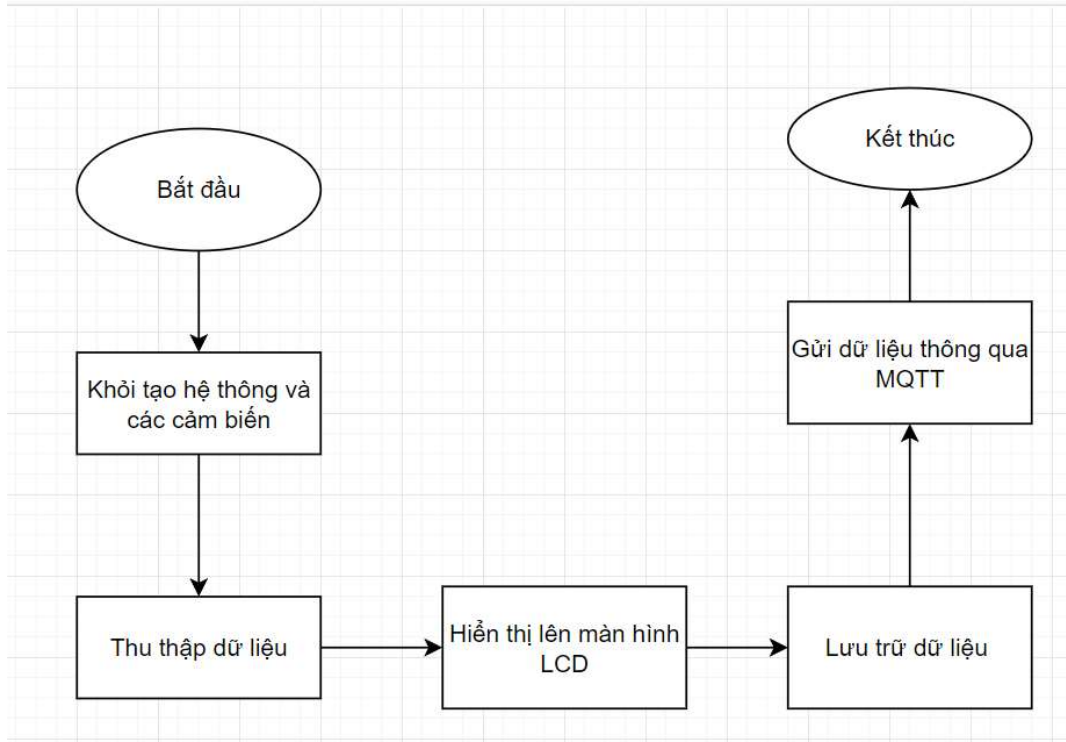
## 5.2 Thiết kế phần cứng

### 5.2.1. Sơ đồ chân kết nối:



Hình 5-9: Sơ đồ kết nối chân

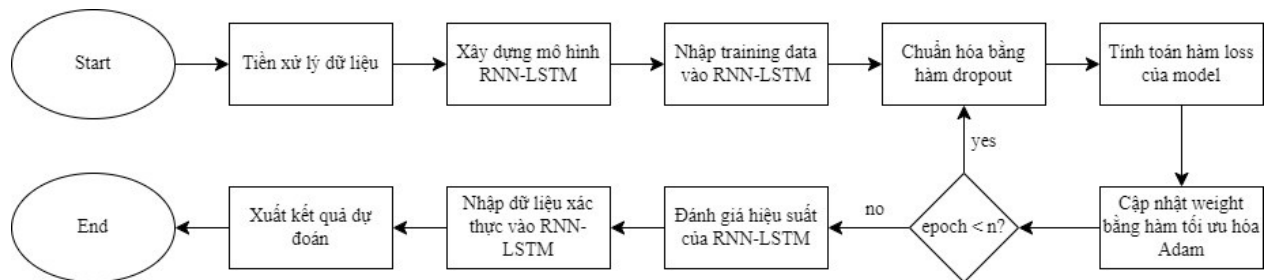
### 5.2.2. Lưu đồ thuật toán phần cứng



Hình 5-30: Lưu đồ thuật toán phần cứng.

## 5.3 Thiết kế model dự đoán thông số LSTM

### 5.3.1. Lưu đồ thuật toán





### 5.2.3. Xây dựng model LSTM

```
class LSTM(nn.Module):
    def __init__(self, num_features=5, hidden_layer_size=512, output_size=1, num_layers=1, batch_size=64):
        super(LSTM, self).__init__()
        self.batch_size = batch_size
        self.num_features = num_features
        self.hidden_layer_size = hidden_layer_size
        self.output_size = output_size
        self.num_layers = num_layers

        self.lstm = nn.LSTM(self.num_features, self.hidden_layer_size, self.num_layers, bidirectional=True, batch_first=True)
        self.linear = nn.Linear(2*hidden_layer_size, self.output_size)

    def init_hidden(self, current_batch_size):
        return (torch.zeros(self.num_layers*2, current_batch_size, self.hidden_layer_size).to(device),
                torch.zeros(self.num_layers*2, current_batch_size, self.hidden_layer_size).to(device))

    def forward(self, input_seq):
        current_batch_size = input_seq.size(0)
        self.hidden = self.init_hidden(current_batch_size)
        lstm_out, _ = self.lstm(input_seq, self.hidden)
        predictions = self.linear(lstm_out[:, -1, :])
        return predictions

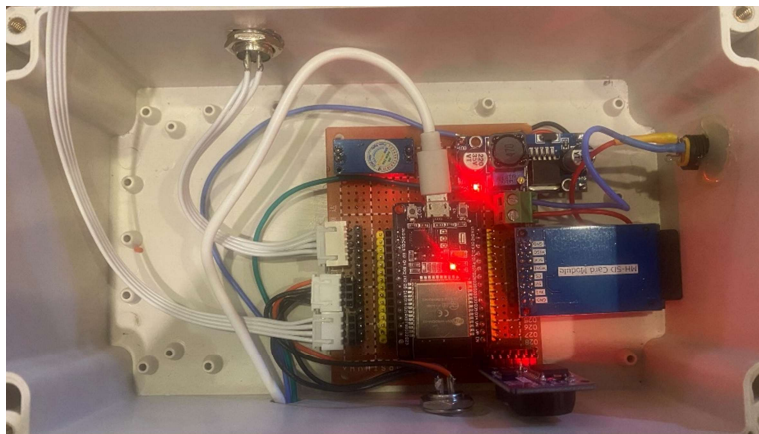
model = LSTM().to(device)
model.hidden = model.init_hidden(1024)
loss_function = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)
```

## 6. KẾT LUẬN

### 6.1 Đánh giá kết quả

#### 6.1.1. Kết quả phần cứng

Mạch có tính ổn định tương đối cao, tuy nhiên có một số lỗi về hiển thị như mất chữ trên LCD khi khởi động nhưng không ảnh hưởng tới việc thu thập dữ liệu từ môi trường không khí, quá trình thu thập dữ liệu và gửi, lưu trữ vẫn diễn ra bình thường.



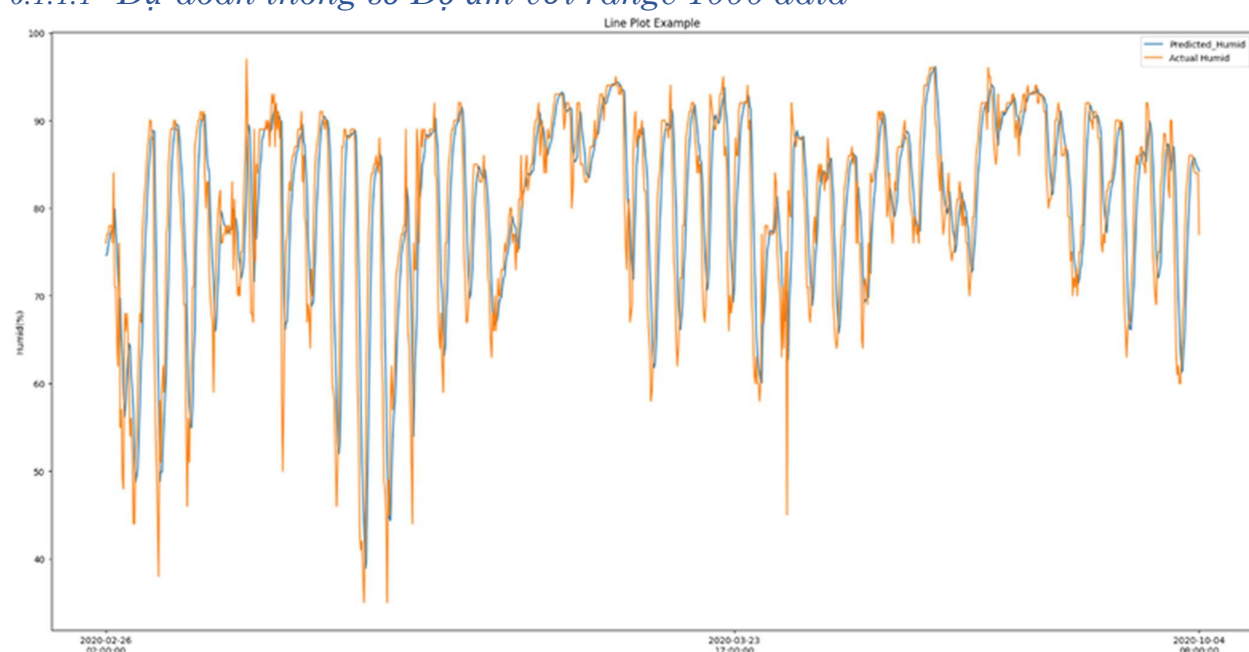
Hình 6-1: Hình ảnh bên trong sản phẩm



Hình 6-2: Hình ảnh bên ngoài của sản phẩm

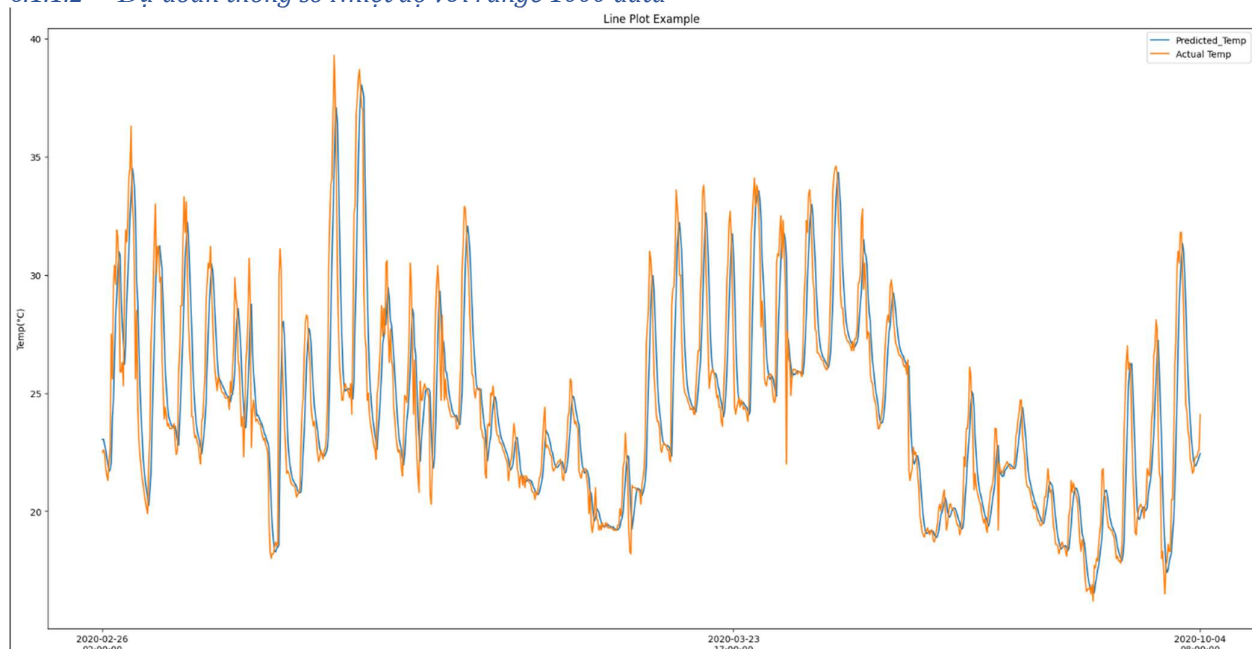
## 6.1.2. Kết quả phần mềm

### 6.1.1.1 Dự đoán thông số Độ ẩm với range 1000 data

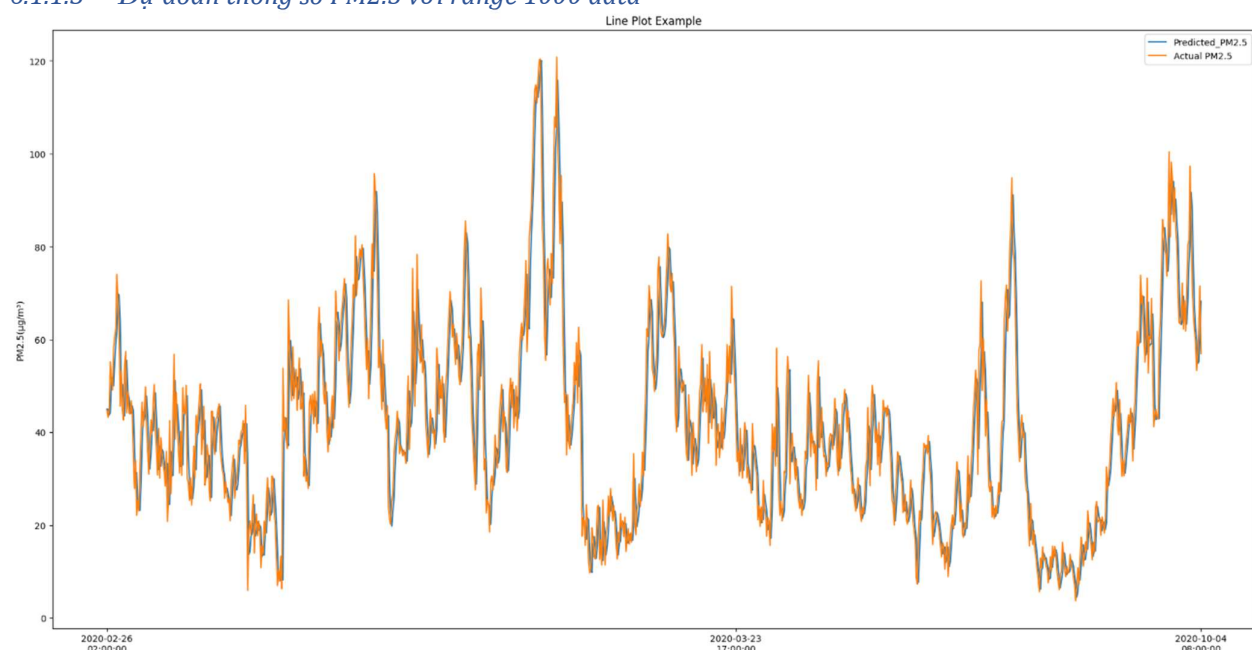




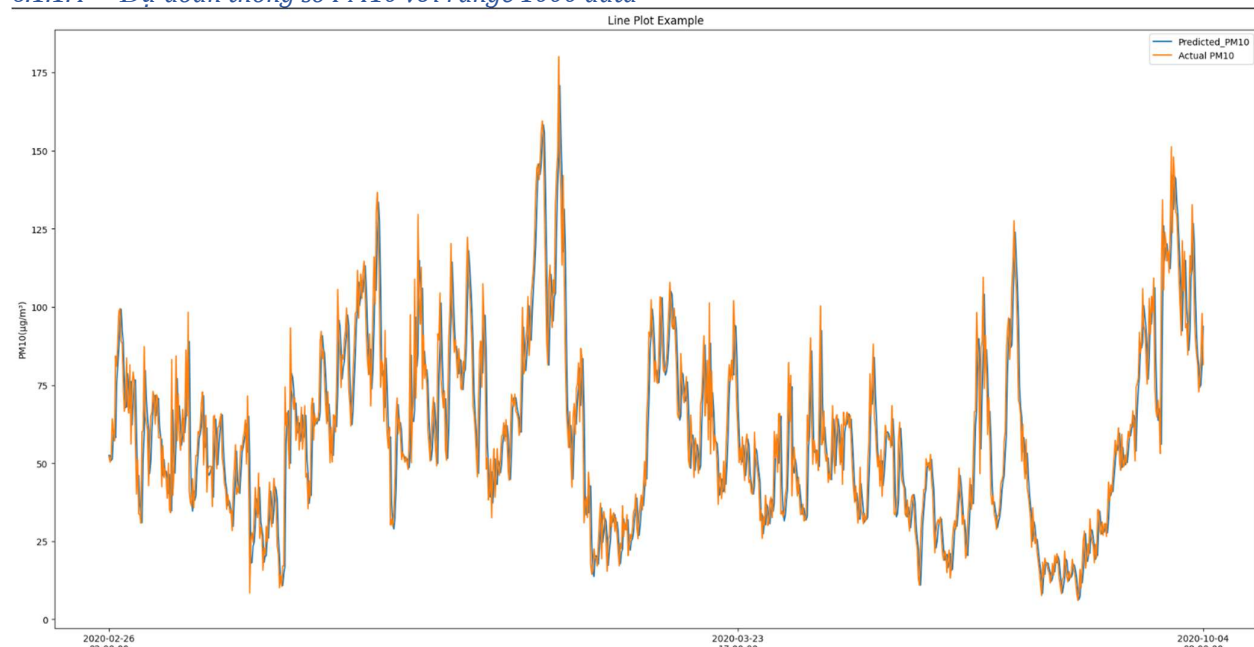
### 6.1.1.2 Dự đoán thông số Nhiệt độ với range 1000 data



### 6.1.1.3 Dự đoán thông số PM2.5 với range 1000 data



#### 6.1.1.4 Dự đoán thông số PM10 với range 1000 data



## 6.2 Hướng phát triển tương lai

Với những kết quả thu được của đồ án, có những mặt tích cực nhưng vẫn còn đó nhiều điểm hạn chế, cần được tiếp tục nghiên cứu giải pháp và hoàn thiện:

- Mở rộng thêm quy mô cho đồ án, từ một node cảm biến thành nhiều node cảm biến di động.
- Sử dụng máy chủ hiệu năng tốc độ cao hơn để xử lý nhiều tác vụ cùng lúc.
- Thiết kế giao diện người dùng có nhiều chức năng hơn.
- Phát triển những phương thức truyền thông khác ví dụ như LoRa, SigFox, ...

## PHỤ LỤC

- [1] Link Github: <https://github.com/quangnmhust/DA2.git>

----- The End -----