

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN IOT CƠ BẢN**

# **HỆ THỐNG DỰ BÁO THỜI TIẾT VÀ TÌNH HÌNH GIAO THÔNG TRONG KHU VỰC**

*Người hướng dẫn:* **GV. PHÙ TRẦN TÍN**

*Người thực hiện:* **TRẦN THỊ VỆ - 52100674**

**NGUYỄN ĐÌNH DANH - 52100878**

**TRƯƠNG BÌNH THUẬN - 52100322**

**Lớp : 21050301**

**Khoá : 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN IOT CƠ BẢN**

# **HỆ THỐNG DỰ BÁO THỜI TIẾT VÀ TÌNH HÌNH GIAO THÔNG TRONG KHU VỰC**

*Người hướng dẫn:* **GV. PHÙ TRẦN TÍN**

*Người thực hiện:* **TRẦN THỊ VỆ – 52100674**

**NGUYỄN ĐÌNH DANH - 52100878**

**TRƯƠNG BÌNH THUẬN - 52100322**

**Lớp : 21050301**

**Khoá : 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

## LỜI CẢM ƠN

Trong suốt quá trình học tập và rèn luyện, chúng em đã nhận được rất nhiều sự giúp đỡ tận tình, sự quan tâm, chăm sóc của GV. Phù Trần Tín. Ngoài ra, chúng em còn được GV truyền đạt những kiến thức, phương pháp mới về toán hay ho và thú vị, thầy cô còn giúp sinh viên có được nhiều niềm vui trong việc học và cảm thấy thoải mái, ... Chúng em xin chân thành cảm ơn các thầy cô rất nhiều trong suốt quá trình học tập này!

Bởi lượng kiến thức của chúng em còn hạn hẹp và gặp nhiều vấn đề trong quá trình học nên báo cáo này sẽ còn nhiều thiếu sót và cần được học hỏi thêm. Chúng em rất mong em sẽ nhận được sự góp ý của quý thầy cô về bài báo cáo này để chúng em rút kinh nghiệm trong những môn học sắp tới. Cuối cùng, chúng em xin chân thành cảm ơn quý thầy cô.

TP Hồ Chí Minh, ngày 01 tháng 11 năm 2023

Sinh viên:

*Trần Thị Vẹn – 52100674*

*Nguyễn Đình Danh - 52100878*

*Trương Bình Thuận - 52100322*

## **ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của chúng tôi và được sự hướng dẫn của GV. Phù Trần Tín. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 01 tháng 11 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trần Thị Vẹn*

*Nguyễn Đình Danh*

*Trương Bình Thuận*

## **PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

### **Phần xác nhận của GV hướng dẫn**

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày    tháng    năm  
(kí và ghi họ tên)

### **Phần đánh giá của GV chấm bài**

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày    tháng    năm  
(kí và ghi họ tên)

## TÓM TẮT

Đây là bài báo cáo cuối kì môn IoT cơ bản và bài báo cáo này trình bày về quá trình thực hiện đồ án cuối kì của nhóm về hệ thống dự báo thời tiết và dự báo tình trạng giao thông, bao gồm:

- Mô tả về đề tài, lí do thực hiện đề tài, các thành phần hệ thống.
- Thiết kế mô hình hệ thống.
- Triển khai hệ thống trên thiết bị thật
- Tìm kiếm và tiền xử lí các tập data và dữ liệu api về thời tiết và giao thông.
- Xây dựng model học máy để đưa ra dự đoán.
- Xây dựng ứng dụng android để hiển thị dữ liệu tới người dùng.

Đồ án cuối kì cũng vận dụng tổng hợp các kiến thức về những vấn đề đã học trong môn học này như sử dụng MCU, các thiết bị IOT; đồng thời áp dụng kiến thức từ các môn học cùng ngành như Machine learning, Android Application để xây dựng một hệ thống hoàn chỉnh.

## MỤC LỤC

<b>TÓM TẮT .....</b>	<b>iv</b>
<b>MỤC LỤC .....</b>	<b>v</b>
<b>DANH MỤC HÌNH VẼ .....</b>	<b>ix</b>
<b>DANH MỤC BẢNG BIỂU .....</b>	<b>x</b>
<b>CHƯƠNG 1: TỔNG QUAN VỀ IOT .....</b>	<b>1</b>
1.1 Khái niệm về IoT .....	1
1.2 Cấu trúc của hệ thống IoT .....	2
1.3 Cách IoT hoạt động .....	3
1.4 Những đặc tính cơ bản của IoT .....	5
1.5 Ưu nhược điểm của IoT .....	6
<i>1.5.1 Ưu điểm của IoT.....</i>	<i>6</i>
<i>1.5.2 Nhược điểm của IoT.....</i>	<i>6</i>
1.6 Vai trò của IoT .....	6
<i>1.6.1 Đối với doanh nghiệp.....</i>	<i>6</i>
<i>1.6.2 Đối với người tiêu dùng.....</i>	<i>7</i>
1.7 Quyền riêng tư và bảo mật IoT .....	8
<b>CHƯƠNG 2: NỀN TẢNG IOT .....</b>	<b>9</b>
2.1 Định nghĩa IoT platform .....	9
2.2 Tính năng của IoT platform .....	9
2.3 Một số nền tảng IoT phổ biến .....	10
<b>CHƯƠNG 3: CÁC NỀN TẢNG IOT SỬ DỤNG .....</b>	<b>13</b>
3.1 ThingSpeak - IoT Cloud Platforms .....	13

3.1.1 Các tính năng quan trọng của <i>Thinkspeak</i> .....	13
3.1.2 Cách <i>Thinkspeak</i> hoạt động .....	13
3.2 Arduino IDE - Nền tảng thiết bị IoT .....	14
3.3 Python (Colab) - Data Platforms .....	15
3.4 Firebase – Real time database .....	16
3.5 Android Studio - Integrated Development Environment .....	17
<b>CHƯƠNG 4: GIỚI THIỆU HỆ THỐNG .....</b>	<b>19</b>
4.1 Tên đề tài .....	19
4.2 Lý do thực hiện đề tài .....	19
4.3 Đối tượng nghiên cứu .....	20
4.4 Thành phần hệ thống .....	20
4.5 Chức năng của các thiết bị .....	21
4.5.1 Cảm Biến Độ Ẩm, Nhiệt Độ <i>DHT11 Temperature Humidity Sensor</i> .....	21
4.5.2 Cảm Biến Cường Độ Ánh Sáng <i>TEMP6000 Analog Light Sensor</i> .....	22
4.5.3 Test board <i>MB102</i> .....	23
4.5.4 Kit RF Thu Phát Wifi <i>ESP8266 NodeMCU Lua V3 CH340</i> .....	23
4.5.6 Mạch Uno SMD <i>CH340 (Arduino Uno Compatible)</i> .....	25
4.5.7 Cảm biến mưa - <i>Rain Water Sensor</i> .....	27
4.5.8 Dây cắm Test Board Đục .....	28
4.5.9 Dây cắm Test Board Cái .....	28
<b>CHƯƠNG 5: THIẾT KẾ HỆ THỐNG .....</b>	<b>30</b>
5.1 Mô tả hệ thống .....	30
5.1.1 Thiết bị IoT .....	30



5.1.2 <i>Phía server xử lý</i> .....	30
5.1.3 <i>Xây dựng app</i> .....	30
5.2 <i>Mô phỏng trên TinkerCad</i> .....	30
5.3 <i>Mạch điện mô phỏng</i> .....	31
5.4 <i>Lưu đồ xử lý</i> .....	31
5.5 <i>Quá trình xử lý</i> .....	32
<b>CHƯƠNG 6: KẾT NỐI THIẾT BỊ THẬT</b> .....	<b>33</b>
6.1 <i>Các thông tin cần xác định</i> .....	33
6.2 <i>Nối mạch và sản phẩm hoàn thiện</i> .....	34
<b>CHƯƠNG 7: HOÀN THIỆN HỆ THỐNG</b> .....	<b>36</b>
7.1 <i>Data Processing</i> .....	36
7.1.1 <i>Tìm dữ liệu để train model</i> .....	36
7.1.2 <i>Merge dữ liệu để hoàn thành dữ liệu cho tập dữ liệu</i> .....	37
7.2 <i>Mã giả xử lý từng quy trình</i> .....	38
7.2.1 <i>Đọc dữ liệu từ các sensor</i> .....	38
7.2.2 <i>Arduino sử dụng ESP8266 (ESP-01) để đọc dữ liệu từ cảm biến</i> .....	41
7.2.3 <i>Train mô hình bằng python</i> .....	43
7.2.4 <i>Đẩy dữ liệu kết quả lên Firebase</i> .....	44
7.2.5 <i>Get dữ liệu qua Android Studio</i> .....	44
7.3 <i>Dữ liệu từ sensor được đưa lên Thingspeak</i> .....	47
7.4 <i>Kết quả demo của ứng dụng</i> .....	49
7.3.1 <i>Trang chủ của ứng dụng</i> .....	49
7.3.2 <i>Trang thống kê của hệ thống</i> .....	50

<b>TÀI LIỆU THAM KHẢO .....</b>	<b>I</b>
Tiếng việt .....	I
Tiếng Anh .....	I

## DANH MỤC HÌNH VẼ

Hình 1: Cấu trúc của hệ thống IoT .....	3
Hình 2: Cách hoạt động IoT .....	5
Hình 3: Các loại nền tảng IoT .....	11
Hình 4: Mô hình thực tế .....	21
Hình 5: DHT11 Temperature Humidity Sensor .....	22
Hình 6: Light Sensor .....	23
Hình 7: Test board MB102 .....	23
Hình 8: Thu Phát Wifi ESP8266 NodeMCU Lua V3 CH340 .....	24
Hình 9: Cảm Biến Chất Lượng Không Khí MQ-135 .....	25
Hình 10: Arduino Uno Compatible .....	27
Hình 11: Rain Water Sensor .....	28
Hình 12: Dây Cắm Breadboard đục đục .....	28
Hình 13: Dây Cắm Breadboard cái cái .....	29
Hình 14: Mô phỏng trên TinkerCad .....	31
Hình 15: Mạch điện mô phỏng .....	31
Hình 16: Lưu đồ xử lý .....	32
Hình 17: Xử lý cụ thể hệ thống .....	32
Hình 18: Kết nối giữa Arduino Uno R3 và Kit MCU 8266 .....	34
Hình 19: Mạch sau khi hoàn thành xong .....	35
Hình 22: Data cuối cùng để train model .....	38
Hình 23: Biểu đồ dự đoán mưa sau 30 phút tới .....	43
Hình 24: Dữ liệu lên Firebase .....	44
Hình 25: Lấy dữ liệu ở android app .....	45
Hình 26: Nhiệt độ .....	47
Hình 27: Độ ẩm .....	47
Hình 28: Mưa .....	48
Hình 29: Cường độ ánh sáng .....	48
Hình 30: Chất lượng không khí .....	49

Hình 31: Trang chủ của ứng dụng .....	49
Hình 32: Biểu đồ hiển thị các thông số trong ứng dụng .....	50
Hình 33: Biểu đồ hiển thị các thông số trong ứng dụng .....	51

## **DANH MỤC BẢNG BIỂU**

Bảng 1: Chi phí thực hiện thiết bị thật .....	21
Bảng 2: Đánh giá thời tiết qua nhiệt độ .....	45
Bảng 3: Đánh giá thời tiết qua độ ẩm .....	46
Bảng 4: Đánh giá thời tiết qua chất lượng không khí .....	46
Bảng 5: : Đánh giá thời tiết qua lượng mưa .....	46
Bảng 6: Đánh giá tình hình kẹt xe tại khu vực .....	47

# CHƯƠNG 1: TỔNG QUAN VỀ IOT

## 1.1 Khái niệm về IoT

IoT (Internet of Things) nghĩa là Internet vạn vật. Là mạng kết nối các đồ vật và thiết bị thông qua cảm biến, phần mềm và các công nghệ khác, cho phép các đồ vật và thiết bị thu thập và trao đổi dữ liệu với nhau.

Internet vạn vật lan tỏa lợi ích của mạng internet tới mọi đồ vật được kết nối, chứ không chỉ dừng lại ở phạm vi một chiếc máy tính. Khi một đồ vật được kết nối với internet, nó sẽ trở nên thông minh hơn nhờ khả năng gửi hoặc nhận thông tin và tự động hoạt động dựa trên các thông tin đó.

Các thiết bị IoT có thể là đồ vật được gắn thêm cảm biến để thu thập dữ liệu về môi trường xung quanh (giống như các giác quan), các máy tính/bộ điều khiển tiếp nhận dữ liệu và ra lệnh cho các thiết bị khác, hoặc cũng có thể là các đồ vật được tích hợp cả hai tính năng trên.

Tiềm năng ứng dụng của internet vạn vật (IoT) trải rộng trên mọi lĩnh vực. Tuy nhiên, mọi hệ thống IoT hoàn chỉnh đều có đủ 4 bước: thu thập dữ liệu, chia sẻ dữ liệu, xử lý dữ liệu và đưa ra quyết định.

**Các công nghệ được sử dụng trong hệ thống IoT có thể bao gồm:**

- Điện toán biên

Điện toán biên đề cập đến công nghệ được sử dụng để điều khiển các thiết bị thông minh thực hiện nhiều tác vụ hơn, không chỉ đơn thuần là gửi hay nhận dữ liệu từ nền tảng IoT của chúng. Công nghệ này tăng cường công suất điện toán tại biên của một mạng lưới IoT, giảm bớt độ trễ trong giao tiếp và cải thiện tốc độ phản hồi.

- Điện toán đám mây

Công nghệ đám mây được sử dụng để lưu trữ dữ liệu từ xa và quản lý thiết bị IoT, giúp nhiều thiết bị trong mạng lưới có thể truy cập dữ liệu.

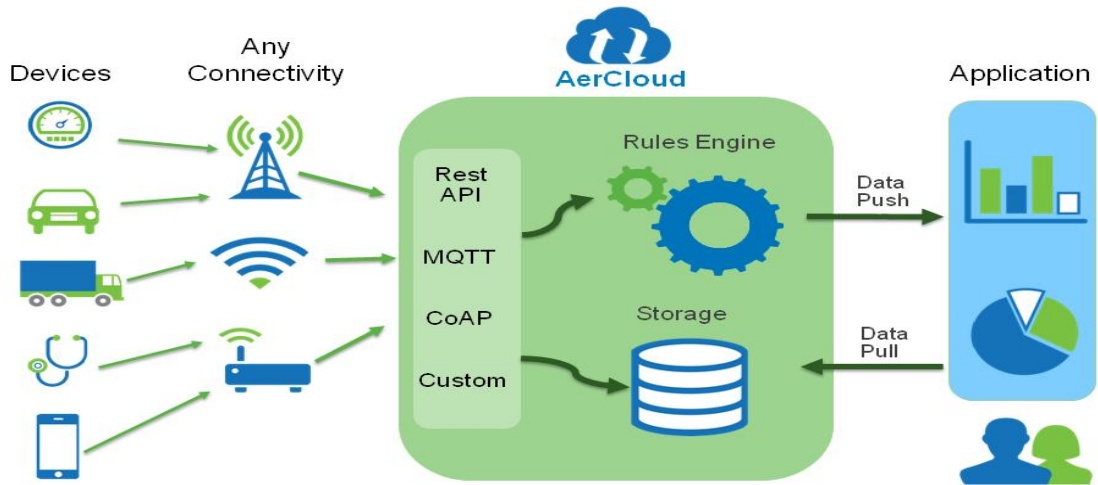
- Máy học

Máy học đề cập đến phần mềm và thuật toán được sử dụng để xử lý dữ liệu và đưa ra các quyết định theo thời gian thực dựa trên dữ liệu đó. Những thuật toán máy học này có thể được triển khai trên đám mây hoặc tại biên.

## **1.2 Cấu trúc của hệ thống IoT**

Một hệ thống IoT thường bao gồm các thành phần sau:

- + Thiết bị cảm biến (Sensor devices): Là những thiết bị thu thập thông tin và gửi dữ liệu đến mạng lưới IoT. Chúng có thể là các cảm biến nhiệt độ, độ ẩm, ánh sáng, hình ảnh, âm thanh, độ rung, v.v.
- + Thiết bị Gateway: Là thiết bị trung gian giữa các thiết bị cảm biến và mạng lưới IoT. Thiết bị này có chức năng thu thập và chuyển tiếp dữ liệu từ các thiết bị cảm biến lên mạng lưới IoT.
- + Mạng lưới IoT (IoT network): Là một mạng lưới kết nối các thiết bị thông minh với nhau và với internet. Các thiết bị có thể giao tiếp với nhau để thực hiện các tác vụ tự động.
- + Trung tâm điều khiển (Control center): Là nơi tổng hợp, xử lý và quản lý dữ liệu từ các thiết bị cảm biến và điều khiển các thiết bị thông minh trong mạng lưới IoT. Nó có thể là một phần mềm hoặc phần cứng.
- + Ứng dụng (Application): Là các ứng dụng được phát triển để sử dụng các dữ liệu và thông tin thu thập được từ mạng lưới IoT. Các ứng dụng này có thể được sử dụng để giám sát và điều khiển các thiết bị trong mạng lưới, hoặc để phân tích và xử lý dữ liệu để cung cấp các thông tin hữu ích cho người dùng.



Hình 1: Cấu trúc của hệ thống IoT

### 1.3 Cách IoT hoạt động

Mỗi phần trong số này cũng sẽ đóng vai trò là cơ cấu tổ chức của phần còn lại của sách điện tử này và chúng ta sẽ đi sâu hơn vào các thành phần này trong các chương để theo dõi.

- **Cảm biến/Thiết bị**

Đầu tiên, cảm biến hoặc thiết bị thu thập dữ liệu từ môi trường của chúng. Dữ liệu này có thể đơn giản như đọc nhiệt độ hoặc phức tạp như một nguồn cấp dữ liệu video đầy đủ. Sử dụng “cảm biến/ thiết bị” vì nhiều cảm biến có thể được kết hợp với nhau hoặc cảm biến có thể là một phần của thiết bị không chỉ cảm nhận mọi thứ.

Ví dụ: điện thoại của bạn là một thiết bị có nhiều cảm biến (máy ảnh, gia tốc kế, GPS, v.v.), nhưng điện thoại của bạn không chỉ là một cảm biến vì nó còn có thể thực hiện nhiều hành động.

Tuy nhiên, cho dù đó là một cảm biến độc lập hay một thiết bị đầy đủ, trong bước đầu tiên này, dữ liệu đang được thu thập từ môi trường bởi một thứ gì đó .

- **Kết nối**

Tiếp theo, dữ liệu đó được gửi đến đám mây, nhưng nó cần một cách để đạt được điều đó. Các cảm biến/ thiết bị có thể được kết nối với đám mây thông qua

nhiều phương thức bao gồm: mạng di động, vệ tinh, WiFi, Bluetooth, mạng diện rộng công suất thấp (LPWAN), kết nối qua cổng/ bộ định tuyến hoặc kết nối trực tiếp với internet qua ethernet...

Mỗi tùy chọn có sự cân bằng giữa mức tiêu thụ điện năng, phạm vi và băng thông. Việc chọn tùy chọn kết nối nào là tốt nhất phụ thuộc vào ứng dụng IoT cụ thể, nhưng tất cả chúng đều hoàn thành nhiệm vụ giống nhau: đưa dữ liệu lên đám mây.

- **Xử lý dữ liệu**

Khi dữ liệu được đưa lên đám mây, phần mềm sẽ thực hiện một số loại xử lý trên đó. Điều này có thể rất đơn giản, chẳng hạn như kiểm tra xem nhiệt độ có nằm trong phạm vi chấp nhận được không. Hoặc nó cũng có thể rất phức tạp, chẳng hạn như sử dụng thị giác máy tính trên video để xác định các đối tượng (chẳng hạn như những kẻ xâm nhập vào tài sản).

- **Giao diện người dùng**

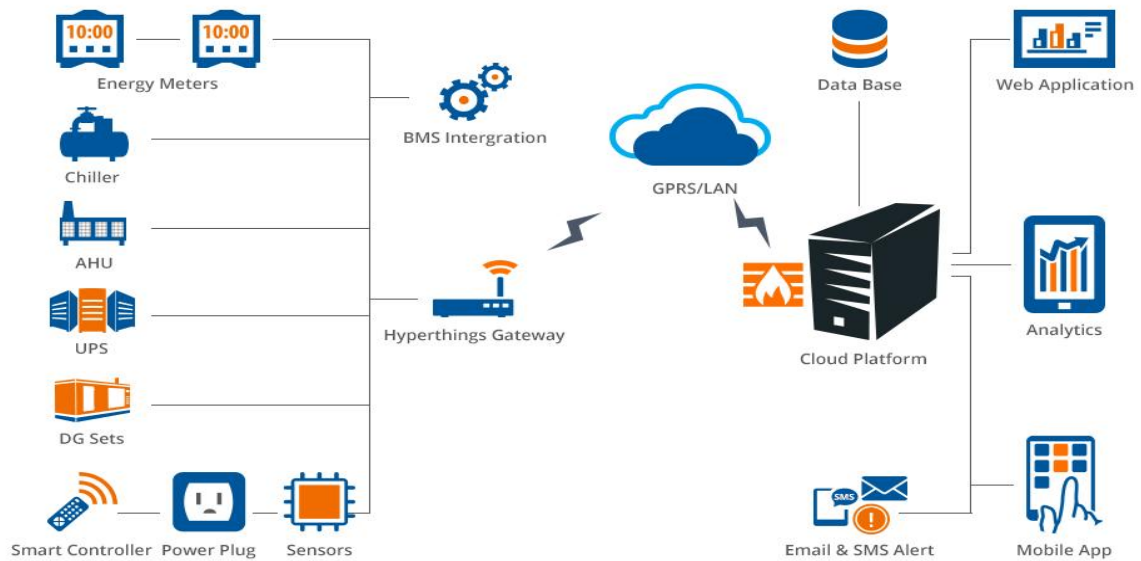
Tiếp theo, thông tin sẽ hữu ích cho người dùng cuối theo một cách nào đó. Điều này có thể thông qua một cảnh báo cho người dùng (email, văn bản, thông báo, v.v.).

Ví dụ, một văn bản cảnh báo khi nhiệt độ quá cao trong kho lạnh của công ty.

Người dùng có thể có một giao diện cho phép họ chủ động đăng ký trên hệ thống.

Ví dụ: người dùng có thể muốn kiểm tra nguồn cấp dữ liệu video trên các thuộc tính khác nhau thông qua ứng dụng điện thoại hoặc trình duyệt web.





Hình 2: Cách hoạt động IoT

#### 1.4 Những đặc tính cơ bản của IoT

- Tính không đồng nhất: các thiết bị trong IoT là không đồng nhất vì nó có phần cứng khác nhau cũng như network khác nhau. Các thiết bị giữa các network có thể tương tác với nhau nhờ vào sự liên kết của các network.
- Tính kết nối liên thông (interconnectivity): với hệ thống IoT thì bất cứ điều gì, vật gì, máy móc gì cũng có thể kết nối với nhau thông qua mạng lưới thông tin và cơ sở hạ tầng liên lạc tổng thể.
- Những dịch vụ liên quan đến “Things”: hệ thống IoT có khả năng cung cấp các dịch vụ liên quan đến “Things” chẳng hạn như bảo vệ sự riêng tư và nhất quán giữa Physical Thing và Virtual Thing. Để cung cấp được dịch vụ này, cả công nghệ phần cứng và công nghệ thông tin(phần mềm) sẽ phải thay đổi.
- Sẽ có quy mô lớn: Sẽ có một số lượng rất lớn các thiết bị, máy móc, được quản lý và giao tiếp với nhau. Số lượng này lớn hơn nhiều so với số lượng máy tính kết nối Internet hiện nay. Số lượng các thông tin được truyền bởi thiết bị sẽ lớn hơn nhiều so với được truyền bởi con người.
- Có thể thay đổi linh hoạt: các trạng thái của các thiết bị điện tử, máy móc có thể tự động thay đổi như ngủ và thức dậy, kết nối hoặc bị ngắt, vị trí thiết bị

đã thay đổi, và tốc độ đã thay đổi... Hơn nữa, số lượng thiết bị có thể tự động thay đổi tùy vào cách mà chúng ta muốn.

## **1.5 Ưu nhược điểm của IoT**

### **1.5.1 Ưu điểm của IoT**

- Khả năng truy cập thông tin từ mọi nơi, mọi lúc trên mọi thiết bị.
- Cải thiện kết nối giữa các thiết bị điện tử.
- Chuyển các gói dữ liệu qua mạng được kết nối tiết kiệm thời gian và tiền bạc.
- Tự động hóa các nhiệm vụ giúp cải thiện chất lượng dịch vụ của doanh nghiệp và giảm nhu cầu can thiệp của con người.

### **1.5.2 Nhược điểm của IoT**

- Khi số lượng thiết bị được kết nối ngày càng tăng và nhiều thông tin được chia sẻ giữa các thiết bị, thì khả năng hacker có thể lấy cắp thông tin bí mật cũng tăng lên.
- Việc thu thập và quản lý dữ liệu từ tất cả các thiết bị của các doanh nghiệp có quy mô cũng sẽ là một thách thức lớn.
- Nếu có lỗi trong hệ thống, có khả năng mọi thiết bị được kết nối sẽ bị hỏng.
- Vì không có tiêu chuẩn quốc tế về khả năng tương thích cho IoT, rất khó để các thiết bị từ các nhà sản xuất khác nhau liên kết với nhau.

## **1.6 Vai trò của IoT**

### **1.6.1 Đối với doanh nghiệp**

Việc ứng dụng IoT trong doanh nghiệp sẽ được chia thành hai phân khúc cụ thể dưới đây:

- Các dịch vụ dành riêng cho ngành như cảm biến trong nhà máy phát điện hoặc thiết bị định vị thời gian thực để tiến hành chăm sóc sức khỏe.
- Các thiết bị IoT sẽ được sử dụng trong tất cả các ngành công nghiệp, như điều hòa không khí thông minh hoặc hệ thống an ninh.

Lợi ích của IoT trong mỗi doanh nghiệp thông thường sẽ phụ thuộc vào việc triển khai cụ thể. Khi ứng dụng IoT cơ bản doanh nghiệp sẽ có quyền truy cập vào nhiều dữ liệu về các sản phẩm của họ.

Các nhà sản xuất đang thêm các cảm biến vào các thành phần của sản phẩm để qua đó có thể truyền đi dữ liệu về cách chúng hoạt động. Việc này sẽ giúp doanh nghiệp phát hiện ra lỗi trước khi thiệt hại xảy ra.

Thông thường nhà sản xuất giúp tạo thêm nhiều cảm biến trong thành phần sản phẩm để từ đó dễ dàng truyền đi nhiều dữ liệu. Chính điều này sẽ giúp các doanh nghiệp có thể nhanh chóng phát hiện ra lỗi và kịp thời sửa chữa trước khi những sai sót này gây ra những hậu quả lớn hơn.

Không chỉ vậy, các doanh nghiệp ứng dụng các thiết bị IoT là gì dựa trên cảm biến nhằm thực hiện chuỗi cung ứng từ đó làm tăng mức độ hoàn hảo của sản phẩm sản xuất ra. Ngoài ra, IoT khuyến khích các công ty cân nhắc lại về cách họ tiếp cận doanh nghiệp của mình và cung cấp cho họ các công cụ để cải thiện chiến lược kinh doanh.

### ***1.6.2 Đối với người tiêu dùng***

IoT hiện nay sẽ giúp tạo nên các thiết bị cảm biến thông minh theo dõi nhịp tim, thể trạng cũng như sức khỏe của con người. Chính nhờ công dụng tuyệt vời của IoT mà đồng hồ thông minh, kính thực tế ảo hay các dây đai theo dõi GPS,... đã được ra đời nhằm biến cuộc sống của con người trở nên dễ dàng và thuận tiện hơn.

Ngoài ra IoT cũng đang phát huy tối đa vai trò của mình để hình thành nên các thành phố thông minh – Smart City hình thành nên hệ thống giám sát thông minh, giao thông tự động, cũng như các hệ thống quản lý năng lượng tự động.

Ngoài ra trong tương lai IoT cũng được ứng dụng nhiều hơn trong việc phân phối nước, đảm bảo an ninh đô thị cũng như tiến hành giám sát môi trường, quan sát thời tiết...

Hiện nay, thiết lập hệ thống IoT đang là xu hướng được nhiều doanh nghiệp theo đuổi để tăng hiệu quả hoạt động và tăng khả năng chống chịu trước các rủi ro. Bởi vậy nên trong tương lai IoT chắc chắn sẽ ngày càng phổ biến và được ứng dụng

vô cùng rộng rãi. IoT sẽ mở ra nhiều cơ hội giúp các doanh nghiệp vận hành ngày càng hiệu quả, nâng cao hiệu suất, qua đó cuộc sống của con người cũng ngày càng hiệu quả hơn.

### **1.7 Quyền riêng tư và bảo mật IoT**

Bởi vì các thiết bị Internet of Things được kết nối chặt chẽ với nhau, tất cả những gì một hacker phải làm là khai thác một lỗ hổng để thao túng tất cả dữ liệu, khiến nó không thể sử dụng được. Các nhà sản xuất không cập nhật thiết bị của họ thường xuyên hoặc hoàn toàn không cập nhật khiến chúng dễ bị tấn công bởi các hacker. Ngoài ra, các thiết bị được kết nối thường yêu cầu người dùng nhập thông tin cá nhân, bao gồm tên, tuổi, địa chỉ, số điện thoại và thậm chí cả tài khoản mạng xã hội, những thông tin vô giá đối với các hacker.

Hacker không phải là mối đe dọa duy nhất đối với Internet of Things mà quyền riêng tư cũng là mối quan tâm lớn khác đối với người dùng IoT. Ngoài việc rò rỉ dữ liệu cá nhân, Internet vạn vật còn gây ra rủi ro cho cơ sở hạ tầng quan trọng, bao gồm điện, giao thông vận tải và các dịch vụ tài chính.

Tất cả các thiết bị gia dụng, máy móc công nghiệp, dịch vụ khu vực công và nhiều thiết bị khác đều được kết nối với Internet. Vì vậy, nó đã tạo ra một kho thông tin khổng lồ có sẵn trên các thiết bị đó và những thông tin này dễ bị tấn công bởi tin tặc. Sẽ rất nghiêm trọng nếu thông tin cá nhân cũng như bí mật của riêng bạn những kẻ xâm nhập trái phép lan truyền.

## CHƯƠNG 2: NỀN TẢNG IOT

IoT platform được tạo ra nhằm cung cấp chức năng chung cho các ứng dụng của người dùng. Điều này giúp bạn tập trung thiết lập điểm mới cho sản phẩm của mình và tăng tính trải nghiệm cho người dùng.

Sở hữu những tính năng ưu việt, IoT platform giúp giảm chi phí và rủi ro trong quá trình phát triển sản phẩm. Bên cạnh đó, nền tảng này còn thúc đẩy sản phẩm của bạn nhanh chóng ra mắt thị trường.

### 2.1 Định nghĩa IoT platform

IoT platform là một dạng phần mềm trung gian, kết nối giữa phần cứng và các ứng dụng. IoT platform có nhiệm vụ thu thập và trao đổi dữ liệu qua các giao thức, cấu trúc mạng liên kết khác nhau. Quản lý, điều khiển các thiết bị từ xa, cập nhật các chương trình qua mạng.

IoT platform tích hợp hầu hết trên các thiết bị có thể kết nối internet và hòa hợp với ứng dụng của bên thứ ba. Sự độc lập của IoT platform cho phép quản lý bất kỳ loại thiết bị được kết nối nào qua một mạng duy nhất, với cách đơn giản nhất. IoT platform hiện đại có nhiều tính năng có giá trị cho cả lớp phần cứng và phần mềm. Nó cung cấp giao diện người dùng, phân tích, xử lý dữ liệu dựa trên cơ sở điện toán đám mây. Một số thậm chí có thể triển khai các giải pháp IoT từ đầu đến cuối.

### 2.2 Tính năng của IoT platform

IoT platform sở hữu một số chức năng thiết yếu như:

- Có khả năng thu thập dữ liệu thông qua cảm biến trong thời gian thực.
- Xử lý và phân tích dữ liệu cục bộ.
- Truyền dữ liệu và nhận lệnh thông qua quá trình kết nối đám mây.
- Lưu trữ dữ liệu trên nền tảng đám mây.
- Cung cấp thông tin chi tiết cho người dùng dựa trên việc phân tích dữ liệu đám mây.
- Thực hiện mọi tác vụ dựa trên thông tin chi tiết.

- Báo cáo thông tin tổng quan cho người dùng.
- Đảm bảo mọi hoạt động diễn ra trên nền tảng IoT được thực hiện an toàn.
- Đảm nhiệm việc quản lý và xác nhận mọi thiết bị IoT của người dùng trên quy mô lớn.

#### Thành phần của IoT platform

- Thiết bị: Thành phần này có chức năng kết nối, thu thập, truyền và nhận dữ liệu. Những thiết bị này bao gồm: các loại cảm biến, máy móc hoặc thiết bị có các chức năng tương tự.
- Phương thức kết nối: Các phương thức này được chọn lựa qua thông số của các thiết bị và server/cloud
- Bộ phận xử lý dữ liệu: Những dữ liệu từ các thiết bị được thu thập bởi server/cloud sẽ được phân tích và thực hiện.
- Giao diện: Toàn bộ quá trình vận hành của hệ thống sẽ được hiển thị tại đây. Nhờ đó, người dùng có thể quan sát và tương tác với dữ liệu dễ dàng hơn.

### 2.3 Một số nền tảng IoT phổ biến

Đặc tính chung của các sản phẩm IoT phần lớn đều rất phức tạp vì chúng phải tích hợp nhiều thành phần “chồng chéo” nhau. Do đó, bạn rất khó tìm kiếm một nền tảng IoT phù hợp với mọi lĩnh vực công nghệ. Vậy nên, hầu hết người dùng đều cần đến những nền tảng IoT khác nhau để áp dụng vào quy trình làm việc.

Hiện nay, có 4 loại nền tảng IoT phổ biến là:

- Nền tảng Cloud

Đây còn được gọi là nền tảng IoT hỗ trợ ứng dụng. Nền tảng này thực hiện các chức năng như: vận chuyển, tiêu thụ, phân ích, lưu trữ và hiển thị dữ liệu. Những hoạt động này tạo nên một khối hoàn chỉnh xây dựng cốt lõi cho sản phẩm của bạn. Nền tảng cloud có khả năng tối ưu hóa tính phức tạp của quá trình thiết lập một giải pháp IoT.

- Nền tảng kết nối IoT

Đối với nền tảng công nghệ IoT, khả năng kết nối là một trong những nhân tố quan trọng. Nhờ vậy, các thiết bị IoT có thể liên kết với nền tảng đám mây hoặc

những kho lưu trữ khác. Nền tảng kết nối IoT có vai trò đảm bảo tính kết nối liên mạch giữa các thiết bị và nền tảng đám mây.

Các tính năng của chúng bao gồm: quản lý kết nối, cập nhật qua Internet, định vị vị trí, cung cấp thiết bị, giám sát giao thông,... Nền tảng này chỉ cung cấp một giao diện trực quan duy nhất. Tại đó, bạn có thể giám sát, quản lý và triển khai mọi thiết bị của mình.

- Nền tảng thiết bị IoT

Trong quá trình phát triển sản phẩm IoT, phần cứng đảm nhiệm vai trò cực kỳ quan trọng. Nền tảng thiết bị IoT có nhiệm vụ cung cấp “nguyên liệu” xây dựng phần cứng. Nhờ đó, người dùng có thể cải tiến và phát triển các thiết bị IoT của mình.

- Nền tảng phân tích

Các sản phẩm công nghệ IoT được tạo ra không phục vụ nhu cầu thu thập dữ liệu. Mục tiêu hàng đầu của chúng là cung cấp những thông tin chi tiết và hữu ích cho người dùng. Đây là lý do vì sao nền tảng phân tích IoT lại ra đời và tích hợp nhiều tính năng quan trọng.

Một số nền tảng nay cung cấp chức năng phân tích chung mà người dùng được phép tùy chỉnh. Bên cạnh đó, một số nền tảng phân tích khác mang đến các tính năng chuyên biệt như: theo dõi tài sản, bảo trì dự đoán, hậu cần,...

## The different types of IoT platforms



Hình 3: Các loại nền tảng IoT

Lợi ích của IoT platform đối với người tiêu dùng: Nền tảng IoT mang lại một cuộc sống thông minh và tiện nghi hơn cho người tiêu dùng.

- Hệ thống an ninh nhà ở: Giúp người dùng giám sát được bên trong và ngoài khuôn viên nhà của mình. Thậm chí, bạn có thể nhìn thấy và trò chuyện với khách đến thăm.
- Máy điều hòa thông minh: Có khả năng tự động làm ấm không gian trước khi người dùng quay về nhà.
- Bóng đèn thông minh: Tự động bật/tắt khi có người hoặc không.

Tiêu chuẩn chọn lựa dịch vụ cung cấp nền tảng IoT:

- Là một thương hiệu uy tín và nhận được sự tin dùng từ những khách hàng trước.
- Sở hữu hệ sinh thái rộng lớn.
- Cung cấp cho người dùng quyền truy cập giao diện lập trình ứng dụng.
- Có giải pháp mạnh mẽ, dịch vụ chuyên nghiệp.
- Có kiến thức về lĩnh vực của doanh nghiệp bạn.

Làm sao phân biệt được các nền tảng IoT khác nhau:

- Khả năng tùy chỉnh, khả năng mở rộng
- Kiểm soát mã
- Khả năng tích hợp với phần mềm của bên thứ 3
- Khả năng bảo mật dữ liệu, mức độ sử dụng dễ hay khó
- Các tùy chọn triển khai

Nền tảng IoT được ứng dụng trong các lĩnh vực:

Nhà máy thông minh: tình trạng hoạt động của máy móc, năng suất sản xuất, năng lượng, môi trường, tình trạng nguyên vật liệu,... Chúng giúp doanh nghiệp giảm bớt gánh nặng về nhân sự.

Giám sát đường ống: IoT platform báo cáo chi tiết các yếu tố ảnh hưởng đến độ bền đường ống như: nhiệt độ, áp suất, độ ăn mòn,... Việc này giúp người vận hành nhanh chóng nhận diện được các vấn đề đang gặp phải. Từ đó, họ có thể đưa ra các giải pháp làm giảm thiệt hại hiệu quả hơn.



## CHƯƠNG 3: CÁC NỀN TẢNG IOT SỬ DỤNG

### 3.1 ThingSpeak - IoT Cloud Platforms

ThingSpeak là một nền tảng trực quan hóa và phân tích dữ liệu trên đám mây. Nó là một sản phẩm Matlab và có thể xử lý và phân tích dữ liệu trực tiếp từ đám mây. Chủ yếu nó được sử dụng trong các dự án IoT cần phân tích để theo dõi những thay đổi về giá trị cảm biến trên đám mây.

Để làm việc với nó, bạn phải đăng nhập thông qua tài khoản Matlab, nó chỉ miễn phí cho người dùng phi thương mại. Phiên bản miễn phí có một số giới hạn nhưng nếu bạn là sinh viên hoặc thuộc bất kỳ cơ sở giáo dục nào thì bạn có thể có toàn quyền truy cập bằng cách sử dụng email từ cơ sở của bạn.

#### 3.1.1 Các tính năng quan trọng của ThingSpeak

ThingSpeak cung cấp các khả năng khác nhau để thu thập, trực quan hóa và phân tích dữ liệu trên đám mây. Sau đây là các tính năng chính:

- Các thiết bị có thể dễ dàng định cấu hình và gửi dữ liệu tới ThingSpeak bằng cách sử dụng các giao thức truyền thông.
- Có thể xem dữ liệu trong thời gian thực
- Nó có thể nhận dữ liệu từ phần mềm của bên thứ ba.
- Có thể sử dụng với Matlab để phân tích dữ liệu
- Nó không cần server và phần mềm web để xây dựng prototype hệ thống IoT.
- Tự động thực hiện các hành động và giao tiếp bằng phần mềm của bên thứ ba như twitter.

#### 3.1.2 Cách ThingSpeak hoạt động

ThingSpeak hoạt động với điện toán đám mây, tất cả các thiết bị muốn lấy dữ liệu phải nằm trong mạng với cơ sở dữ liệu đám mây. ThingSpeak cũng kết nối với cơ sở dữ liệu đám mây và hiển thị luồng dữ liệu.

Ví dụ: muốn theo dõi nhiệt độ và độ ẩm của phòng làm việc. Phải đặt một bộ điều khiển và các bộ cảm biến được thiết lập trong văn phòng của mình và kết nối nó thông qua các API REST. Bộ điều khiển sẽ thu thập dữ liệu của cảm biến nhiệt

độ và độ ẩm và gửi dữ liệu đó lên đám mây để hiển thị trực quan trên Thingspeak. có thể kiểm tra tình trạng phòng của mình thông qua điện thoại thông minh hoặc web chỉ bằng cách đăng nhập vào tài khoản Thingspeak.

Thingspeak hoạt động với các thiết bị sau:

- Arduino
- Raspberry pi
- Matlab
- Module ESP8266, module ESP32
- LoRaWAN và nhiều thiết bị khác.

### 3.2 Arduino IDE - Nền tảng thiết bị IoT

Arduino là một nền tảng điện tử với mã nguồn mở, dựa trên phần cứng và phần mềm, với thiết kế linh hoạt và dễ sử dụng. Arduino cũng có thể xem như một máy tính thu nhỏ, giúp người dùng lập trình, thực hiện các dự án điện tử dễ dàng hơn mà không cần đến các công cụ chuyên biệt hỗ trợ việc nạp code.

Arduino có thể tương tác với môi trường xung quanh thông qua các loại cảm biến, nút nhấn, ... Arduino cũng có thể được sử dụng để điều khiển các thiết bị như động cơ, đèn và gửi thông tin đến các thiết bị khác.

#### Phần mềm Arduino IDE

Được phát triển trên nền tảng IDE nên việc viết một chương trình (code) giờ đây thật đơn giản. Với Arduino IDE gần như bạn không cần phải code nhiều mà đã được các nhà sản xuất, doanh nghiệp, cộng đồng, cá nhân hỗ trợ mã chương trình (code), thư viện.

Hiện nay, Arduino được sử dụng rất rộng rãi với những ưu điểm độc đáo của mình.

- **Môi trường phát triển đơn giản, rõ ràng**

Đây là đặc điểm nổi bật nhất của Arduino với môi trường phát triển ứng dụng cực kỳ dễ sử dụng. Cách sử dụng board Arduino cũng như phần mềm Arduino IDE khá dễ dàng, các tập lệnh lập trình đơn giản dễ sử dụng ngay cả với những người ít am hiểu về điện tử hay người mới bắt đầu lập trình.

- **Giá thành rẻ**

Đây cũng là một điều làm nên sự phổ biến của Arduino, giá thành của Arduino khá thấp so với những dòng vi điều khiển khác, phù hợp cho những người mới bắt đầu lập trình và cả những bạn học sinh sinh viên.

- **Đa nền tảng**

Công cụ phát triển của Arduino có thể chạy trên các hệ điều hành như Windows, Mac OS hay Linux.

- **Phần cứng nguồn mở và dễ dàng mở rộng**

Không chỉ với mã nguồn mở mà Arduino còn cung cấp cả phần cứng nguồn mở. Bạn có thể hoàn toàn tự thiết kế một board Arduino riêng cho mình để phù hợp với nhu cầu sử dụng dựa trên ý tưởng ban đầu của Arduino.

### **3.3 Python (Colab) - Data Platforms**

Python (Colab) có thể được xem xét là một phần của loại "Data Platforms," nhất là khi nó được sử dụng để phân tích và xử lý dữ liệu. Dưới đây là một lý do và giải thích chi tiết:

#### **Chức năng phân tích dữ liệu**

- Colab, một phiên bản của Jupyter Notebook chạy trên cloud, cung cấp một môi trường thí nghiệm tương tác cho việc phân tích dữ liệu.
- Python, với các thư viện như NumPy, Pandas, Matplotlib, Seaborn, và SciPy, là một trong những ngôn ngữ lập trình phổ biến cho phân tích dữ liệu và học máy.
- Các chức năng của Python (Colab) trong việc thực hiện các phân tích và xử lý dữ liệu có thể đưa nó vào phạm vi của "Data Platforms."

#### **Trực quan hóa và báo cáo**

- Colab cung cấp khả năng trực quan hóa dữ liệu thông qua các thư viện như Matplotlib và Seaborn, giúp người dùng hiểu rõ hơn về dữ liệu.
- Việc trực quan hóa và tạo báo cáo từ dữ liệu là một phần quan trọng trong loại "Data Platforms."

#### **Tích hợp với dữ liệu từ nguồn khác**

- Python (Colab) có thể được sử dụng để tích hợp dữ liệu từ nhiều nguồn khác nhau, bao gồm cả dữ liệu từ ThingSpeak, MongoDB, và các nền tảng khác.
- Khả năng tích hợp và làm việc với nhiều nguồn dữ liệu là một khía cạnh quan trọng của "Data Platforms."

Lưu ý rằng việc phân loại có thể phụ thuộc vào cách mà Python (Colab) được sử dụng trong ngữ cảnh cụ thể của dự án hoặc mục đích nào đó. Python (Colab) có khả năng linh hoạt và có thể được sử dụng cho nhiều mục đích khác nhau, bao gồm cả phân tích dữ liệu và phát triển machine learning models.

### 3.4 Firebase – Real time database

Firebase ra đời dưới dạng Backend-as-a-Service với chức năng thời gian thực. Sau khi được Google mua lại vào năm 2014 thì Firebase nhanh chóng phát triển thành nền tảng phát triển ứng dụng đa năng của di động và website như ngày nay. Nền tảng này là sự kết hợp giữa cloud với hệ thống máy chủ của Google để tập trung chính cho 2 đối tượng là:

- Develop & test your app: phát triển và thử nghiệm các ứng dụng được thiết kế.
- Grow & engage your audience: phân tích dữ liệu và tối ưu hóa trải nghiệm với người dùng.

Firebase cung cấp cho chúng ta những API đơn giản, mạnh mẽ và đa nền tảng trong việc quản lý, sử dụng database, bởi vậy giờ đây chúng ta chỉ cần gọi API và phần server đã có Firebase lo!

Firebase Realtime Database là một cơ sở dữ liệu NoSQL được lưu trữ trên đám mây, giúp các nhà phát triển xây dựng ứng dụng thời gian thực bằng cách lưu trữ và đồng bộ dữ liệu trên nhiều thiết bị. Dưới đây là một số tính năng và khái niệm quan trọng liên quan đến Firebase Realtime Database:

#### **Đồng bộ dữ liệu thời gian thực:**

Firebase Realtime Database cung cấp tính năng đồng bộ dữ liệu thời gian thực. Điều này có nghĩa là mọi thay đổi trong dữ liệu được phản ánh ngay lập tức trên tất cả các thiết bị kết nối mà không cần làm mới thủ công.

**Cơ sở dữ liệu NoSQL:**

Đây là cơ sở dữ liệu NoSQL, có nghĩa là nó không sử dụng bảng và hàng SQL truyền thống. Thay vào đó, dữ liệu được lưu trữ dưới dạng JSON, cho phép cấu trúc dữ liệu linh hoạt và có thể mở rộng.

**Mô hình dữ liệu JSON:**

Dữ liệu được tổ chức dưới dạng cây JSON, trong đó mỗi mảnh dữ liệu có một đường dẫn duy nhất trong cơ sở dữ liệu. Cấu trúc phân cấp này cho phép truy vấn và truy xuất dữ liệu một cách hiệu quả.

**Hỗ trợ ngoại tuyến:**

Firebase Realtime Database cung cấp hỗ trợ ngoại tuyến, cho phép ứng dụng tiếp tục hoạt động ngay cả khi thiết bị không kết nối internet. Những thay đổi được thực hiện ngoại tuyến sẽ được đồng bộ khi thiết bị kết nối lại.

**Xác thực và quy tắc bảo mật:**

Firebase cung cấp các cơ chế xác thực tích hợp giúp bạn kiểm soát quyền đọc và ghi của người dùng đối với các phần cụ thể của cơ sở dữ liệu. Quy tắc bảo mật được định nghĩa bằng một ngôn ngữ dựa trên JSON.

**Tích hợp với xác thực Firebase:**

Nó có thể tích hợp dễ dàng với Xác Thực Firebase, cho phép bạn quản lý xác thực và ủy quyền người dùng cùng với dữ liệu thời gian thực.

**Kích hoạt với Cloud Functions:**

Firebase Realtime Database có thể kích hoạt Cloud Functions khi có sự kiện trong cơ sở dữ liệu, chẳng hạn như tạo, cập nhật hoặc xóa dữ liệu. Điều này cho phép bạn thực hiện logic phía máy chủ đáp ứng với các thay đổi trong cơ sở dữ liệu.

**Khả năng mở rộng:**

Firebase Realtime Database tự động mở rộng để xử lý các yêu cầu của ứng dụng. Nó được thiết kế để mở rộng theo chiều ngang qua nhiều hệ thống và thiết bị.

**3.5 Android Studio - Integrated Development Environment**

Android Studio là một IDE (Integrated Development Environment) được thiết kế chủ yếu cho việc phát triển ứng dụng di động Android, bao gồm cả các ứng dụng IoT (Internet of Things).

**Hỗ trợ nền tảng IoT và đám mây:**

- Có thể tích hợp với nền tảng IoT và các dịch vụ đám mây như Firebase để lưu trữ và quản lý dữ liệu.
- Hỗ trợ kết nối với các thiết bị IoT thông qua Bluetooth, Wi-Fi và các giao thức khác.

**Gradle Build System:**

- Android Studio sử dụng hệ thống quản lý dự án Gradle để quản lý và xây dựng ứng dụng.
- Gradle cung cấp khả năng linh hoạt trong việc cấu hình và xây dựng ứng dụng Android.

**Intelligent Code Editor:**

- Bao gồm một trình soạn thảo mã nguồn thông minh với tính năng tự động hoàn thành mã, kiểm tra lỗi và cung cấp gợi ý.
- Hỗ trợ cả ngôn ngữ lập trình Kotlin và Java.

**Visual Layout Editor:**

- Cung cấp trình chỉnh sửa giao diện người dùng (UI) để dễ dàng thiết kế giao diện của ứng dụng mà không cần viết mã.
- Hỗ trợ kéo và thả để xây dựng giao diện người dùng.

Android Studio đi kèm với Android Emulator, cho phép kiểm thử ứng dụng trên nhiều thiết bị ảo với các phiên bản Android khác nhau. Hỗ trợ cả việc kết nối và kiểm thử trên các thiết bị thật.

## CHƯƠNG 4: GIỚI THIỆU HỆ THỐNG

### 4.1 Tên đề tài

Hệ thống dự báo thời tiết và dự báo tình hình giao thông tại khu vực  
(Weather forecasting and traffic congestion prediction system in the area. )

### 4.2 Lý do thực hiện đề tài

Việt Nam, với địa thế nằm trong khu vực nhiệt đới gió mùa, thường phải đối mặt với một hiện tượng thời tiết đặc trưng và đôi khi khá khó lường: mưa bão. Mùa mưa không đơn thuần là sự thay đổi thông thường của thời tiết, mà đôi khi biến thành những trận mưa dữ dội và khó đoán. Mưa bão có thể ập đến một cách bất thành linh và kéo dài gây ra không ít khó khăn không chỉ cho những người nông dân mong chờ một ngày nắng để hong khô những tạ thóc mới gặt, mà còn là những phiền phức ảnh hưởng rõ đến cuộc sống hàng ngày tại các đô thị lớn.

Khi mưa bão kéo đến, tình hình giao thông thường trở nên hỗn loạn, và tình trạng kẹt xe nhiều giờ là không thể tránh khỏi. Điều này đặc biệt trầm trọng tại các thành phố lớn như Hà Nội hay Hồ Chí Minh. Những ngày mưa bão, các đường phố thường ngập nước, gây ra sự đau đầu cho người dân và tạo ra những khó khăn không nhỏ cho việc di chuyển và giao thông công cộng. Đôi khi chỉ muốn di chuyển từ nhà đến cơ quan 2km nhưng vì vô tình đi vào đoạn đường kẹt xe, thời gian đi hết 2km đó có thể tính bằng giờ.

Trên thị trường, hiện tại đã có rất nhiều app dự báo thời tiết, tuy nhiên thường đều dựa vào dữ liệu vệ tinh để dự báo, nghĩa là các khu vực dự báo thường chung chung và không cụ thể. Hơn thế nữa, dữ liệu vệ tinh cũng không cung cấp được thông tin chi tiết về tình hình giao thông. Điều này làm cho các ứng dụng dự báo thời tiết trở nên hạn chế khi người dùng cần biết thông tin cụ thể về tình trạng lưu thông trên các tuyến đường.

Nhận thức về những khó khăn này, nhóm chúng em đã quyết định thực hiện đề tài "Hệ thống dự báo thời tiết và dự báo tình trạng giao thông tại khu vực" với mong muốn thiết kế **một thiết bị có các cảm biến phù hợp** đặt tại ngã tư, và **một**

**ứng dụng giúp hiển thị tức thì các thông tin** về tình trạng thời tiết và tình trạng kẹt xe hiện tại cũng như sắp tới ở khu vực lắp thiết bị. Như vậy, chỉ với vài cú chạm trên điện thoại, người dùng đã ngay lập tức biết tình hình giao thông và tình hình mưa bão để sắp xếp di chuyển trên một tuyến đường khác. Điều này không chỉ tiết kiệm hàng giờ thời gian di chuyển tránh tắc nghẽn tại các đô thị lớn, mà còn giúp người dân sinh sống quanh khu vực chủ động sắp xếp các hoạt động trong gia đình để giảm thiểu ảnh hưởng do thay đổi thời tiết gây ra.

### 4.3 Đối tượng nghiên cứu

Để thực hiện đề tài này, các đối tượng cần nghiên cứu bao gồm:

- Các cảm biến sử dụng trong đo đạc môi trường phục vụ cho dự báo thời tiết.
- Các loại mô hình học máy và các phương pháp train trên tập dữ liệu.
- Dữ liệu thời tiết các năm trước và dữ liệu dự báo; dữ liệu về tình hình giao thông tại khu vực.
- Cách xây dựng app hiển thị dự báo và các biểu đồ, thông tin liên quan.
- Người dùng cuối, bao gồm tài xế, cơ quan quản lý giao thông và những người quan tâm đến tình hình thời tiết và tình hình giao thông tại khu vực.

### 4.4 Thành phần hệ thống

a) Thiết bị và chi phí:

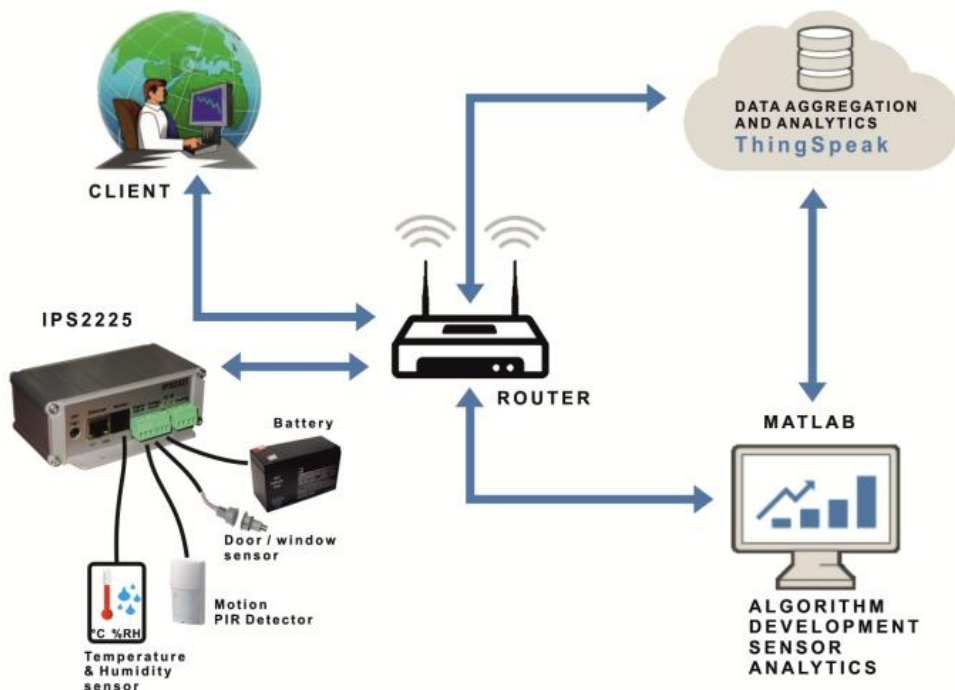
Thiết bị	Số lượng	Giá tiền (VNĐ)
Cảm biến nhiệt độ, độ ẩm DHT 11	1	21.000
Cảm biến cường độ ánh sáng Light sensor	1	23.000
Test Board MB102	1	25.000
Dây cắm Test Board Đục	1	28.000
Dây cắm Test Board Cái	1	28.000
Kit RF Thu Phát Wifi ESP8266 NodeMCU Lua V3 CH340	1	72.000



Cảm biến MQ135	1	31.000
Mạch Uno SMD CH340 (Arduino Uno Compatible)	1	99.000
Cảm biến mưa	1	11.000
Tổng	9	338.000

Bảng 1: Chi phí thực hiện thiết bị thật

## b) Mô hình thực tế



Hình 4: Mô hình thực tế

## 4.5 Chức năng của các thiết bị

### 4.5.1 Cảm Biến Độ Ẩm, Nhiệt Độ DHT11 Temperature Humidity Sensor

Cảm biến độ ẩm và nhiệt độ DHT11 Temperature Humidity Sensor là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1 wire (giao tiếp digital 1 dây truyền dữ liệu duy nhất). Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không phải qua bất kỳ tính

toán nào. So với cảm biến đời mới hơn là DHT22 thì DHT11 cho khoảng đo và độ chính xác kém hơn rất nhiều.

Thông tin kỹ thuật:

- Nguồn: 3 -> 5 VDC.
- Dòng sử dụng: 2.5mA max (khi truyền dữ liệu).
- Khoảng đo độ ẩm: 20%-90% RH (sai số 5%RH)
- Khoảng đo nhiệt độ: 0-50°C (sai số 2°C)
- Tần số lấy mẫu tối đa: 1Hz (1 giây / lần)
- Kích thước 15mm x 12mm x 5.5mm.



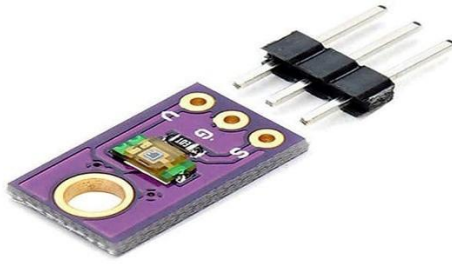
Hình 5: DHT11 Temperature Humidity Sensor

#### 4.5.2 *Cảm Biến Cường Độ Ánh Sáng TEMT6000 Analog Light Sensor*

Cảm biến cường độ ánh sáng TEMT6000 Analog Light Sensor được sử dụng để đo cường độ ánh sáng của môi trường xung quanh, cảm biến là dạng Phototransistor được mắc nối tiếp với điện trở tạo thành cầu phân áp trả ra giá trị điện áp Analog tuyến tính với cường độ ánh sáng, phù hợp với các ứng dụng đo cường độ sáng trong nông nghiệp, nhà thông minh,...

Thông số kỹ thuật:

- IC chính: TEMT6000 từ hãng Vishay
- Điện áp sử dụng: Max 6VDC
- Chuẩn giao tiếp: trả giá trị điện áp Analog tuyến tính với cường độ ánh sáng.
- Kích thước: 60 x 80mm



Hình 6: Light Sensor

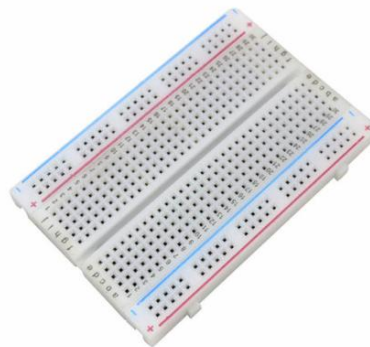
#### 4.5.3 Test board MB102

Breadboard MB-102 830 lỗ 165x55x10mm được sử dụng để gắn các module hoặc linh kiện điện tử, kết nối chúng với nhau bằng các loại dây cắm, dây nối test board giúp test, kiểm tra tính năng 1 cách dễ dàng trước khi tạo thành các thành phẩm hoàn chỉnh.

Breadboard MB-102 830 lỗ 165x55x10mm là loại kích thước chuẩn quốc tế, chất lượng và độ bền cao, có thể sử dụng với 1 số module cấp nguồn được thiết kế chuyên biệt.

Thông số kỹ thuật:

- Chất liệu: Nhựa, mối tiếp xúc bằng đồng mạ.
- Số điểm trên test board: 830 điểm.
- Kích thước: 165 x 55 x 10 mm



Hình 7: Test board MB102

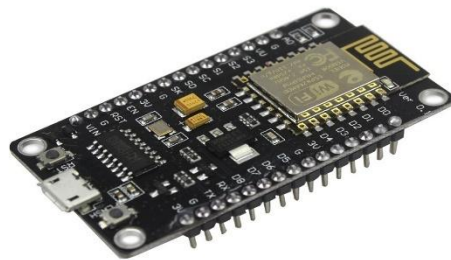
#### 4.5.4 Kit RF Thu Phát Wifi ESP8266 NodeMCU Lua V3 CH340

Kit RF thu phát Wifi ESP8266 NodeMCU Lua V3 CH340 là phiên bản NodeMCU sử dụng IC nạp CH340 từ Lolin với bộ xử lý trung tâm là module Wifi SoC ESP8266, kit có thiết kế dễ sử dụng và đặc biệt là có thể sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code, điều này khiến việc sử dụng và lập trình các ứng dụng trên ESP8266 trở nên rất đơn giản.

Kit RF thu phát Wifi ESP8266 NodeMCU Lua V3 CH340 được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua sóng Wifi, đặc biệt là các ứng dụng liên quan đến IoT.

Thông số kỹ thuật:

- IC chính: ESP8266 Wifi SoC.
- Phiên bản firmware: NodeMCU Lua
- Chip nạp và giao tiếp UART: CH340
- GPIO tương thích hoàn toàn với firmware Node MCU.
- Cấp nguồn: 5VDC MicroUSB hoặc Vin.
- GPIO giao tiếp mức 3.3VDC
- Tích hợp Led báo trạng thái, nút Reset, Flash.
- Tương thích hoàn toàn với trình biên dịch Arduino.
- Kích thước: 59 x 32mm



Hình 8: Thu Phát Wifi ESP8266 NodeMCU Lua V3 CH340

#### ***4.5.5 Cảm Biến Chất Lượng Không Khí MQ-135***

Mô tả:

Thường được dùng trong các thiết bị kiểm tra chất lượng không khí bên trong cao ốc, văn phòng, thích hợp để phát hiện NH<sub>3</sub>, NO<sub>x</sub>, Ancol, Benzen, khói, CO<sub>2</sub>,...

Thông số kỹ thuật:

- Điện áp nguồn: 5V DC
- Điện áp của heater:  $5V \pm 0.1$  AC/DC
- Điện trở tải: thay đổi được ( $2k\Omega$ - $47k\Omega$ )
- Điện trở của heater:  $33\Omega \pm 5\%$
- Công suất tiêu thụ của heater: ít hơn 800mW
- Khoảng phát hiện: 10 - 300 ppm NH<sub>3</sub>, 10 - 1000 ppm Benzene, 10 - 300 Alcol
- Kích thước: 32mm\*20mm
- Khoảng đo rộng
- Bền, tuổi thọ cao
- Phát hiện nhanh, độ nhạy cao
- Mạch đơn giản



Hình 9: Cảm Biến Chất Lượng Không Khí MQ-135

#### **4.5.6 Mạch Uno SMD CH340 (Arduino Uno Compatible)**

Mạch Uno SMD CH340 (Arduino Uno Compatible) có thiết kế tương thích với Arduino Uno sử dụng IC nạp và giao tiếp UART CH340 và vi điều khiển

ATMEGA328P loại IC dán khó thay thế khi hư hỏng, tuy nhiên sẽ có thêm 2 chân Analog A6, A7 so với Arduino Uno phiên bản sử dụng IC cắm.

Thông số kỹ thuật:

- Chip điều khiển chính: ATMEGA328P loại IC dán
- Chip nạp và giao tiếp UART: CH340
- Nguồn nuôi mạch: 5VDC từ cổng USB hoặc nguồn ngoài cắm từ giắc tròn DC (nếu sử dụng nguồn ngoài từ giắc tròn DC Hshop.vn khuyên bạn nên cấp nguồn từ 6~9VDC để đảm bảo mạch hoạt động tốt, nếu bạn cắm 12VDC thì IC ỏn áp rất nóng, dễ cháy và gây hư hỏng mạch).
- Số chân Digital I/O: 14 (trong đó 6 chân có khả năng xuất xung PWM).
- Số chân PWM Digital I/O: 6
- Số chân Analog Input: 6
- Dòng điện DC Current trên mỗi chân I/O: 20 mA
- Dòng điện DC Current chân 3.3V: 50 mA
- Flash Memory: 32 KB (ATmega328P), 0.5 KB dùng cho bootloader.
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Clock Speed: 16 MHz
- LED\_BUILTIN: 13
- Kích thước: 68.6 x 53.4 mm



Hình 10: Arduino Uno Compatible

#### 4.5.7 Cảm biến mưa - Rain Water Sensor

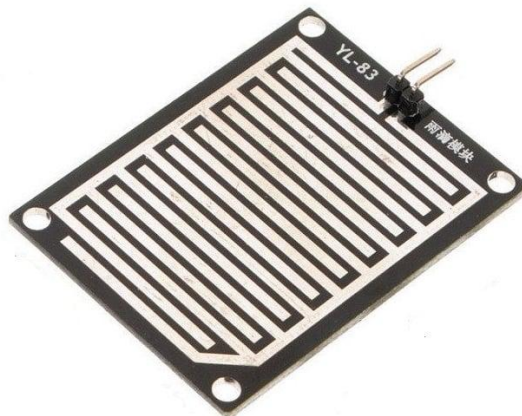
Cảm biến nước mưa (Rain Water Sensor ) được sử dụng để phát hiện mưa, nước hoặc các dung dịch dẫn điện tiếp xúc với bề mặt cảm biến sẽ phát ra tín hiệu để làm các ứng dụng tự động: phát hiện mưa, báo mực nước tự động,...

Thông số kỹ thuật:

- Điện áp sử dụng: 5VDC
- Kích thước tấm cảm biến mưa: 54 x 40mm
- Kích thước board PCB: 30 x 16mm
- Tín hiệu đầu ra: Digital TTL (0VDC / 5VDC) và đầu ra Analog A0 trả giá trị điện áp tuyến tính theo lượng nước tiếp xúc với cảm biến.
- Lỗ cố định bu lông dễ dàng để cài đặt
- Có đèn báo hiệu nguồn và đầu ra
- Độ nhạy có thể được điều chỉnh thông qua chiết áp
- LED sáng lên khi không có mưa đầu ra cao, có mưa, đầu ra thấp LED tắt.

Chế độ kết nối:

- VCC: Nguồn
- GND: Đất
- D0: Đầu ra tín hiệu TTL chuyển đổi
- A0: Đầu ra tín hiệu Analog



Hình 11: Rain Water Sensor

#### 4.5.8 Dây cắm Test Board Đục

Dây Cắm Breadboard đục đục 20cm 40 sợi loại tốt (M-M Jumper Wire) được sử dụng với Breadboard để kết nối các module và linh kiện điện tử với nhau, dây có chất lượng tốt với lõi đồng giúp dây mềm, khó đứt và không bị oxy hóa so với các loại chất lượng kém giá rẻ lõi nhôm trên thị trường (dây cứng, cắt ra thấy bị oxy hóa và rất dễ đứt).

Thông số kỹ thuật:

- Một bó có 40 dây.
- Dây lõi đồng nhiều sợi có độ dẫn điện cao, mỗi tiếp xúc chắc chắn, độ dài 20cm, có nhiều màu sắc khác nhau (10 màu).



Hình 12: Dây Cắm Breadboard đục đục

#### 4.5.9 Dây cắm Test Board Cái

Dây Cắm Breadboard cái cái 10cm 40 sợi loại tốt (F-F Jumper Wire) được sử dụng với Breadboard để kết nối các module và linh kiện điện tử với nhau, dây có chất lượng tốt với lõi đồng giúp dây mềm, khó đứt và không bị oxy hóa so với các loại chất lượng kém giá rẻ lõi nhôm trên thị trường (dây cứng, cắt ra thấy bị oxy hóa và rất dễ đứt).

Thông số kỹ thuật:

- Một bó có 40 dây.



- Dây lõi đồng nhiều sợi có độ dẫn điện cao, mỗi tiếp xúc chắc chắn, độ dài 10cm, có nhiều màu sắc khác nhau (10 màu).



Hình 13: Dây Cắm Breadboard cái cái

## CHƯƠNG 5: THIẾT KẾ HỆ THỐNG

### 5.1 Mô tả hệ thống

Hệ thống chia làm 3 phần chính:

#### 5.1.1 Thiết bị IoT

Khảo sát và lắp đặt một thiết bị IoT hoàn chỉnh có khả năng thu thập thông tin thời tiết, bao gồm các cảm biến như đã liệt kê ở phần trước. Thiết bị này sau đó gửi các data thu thập được lên server ThingSpeak để tiến hành xử lý.

#### 5.1.2 Phía server xử lý

- **Model học máy:** đã được train trên tập dữ liệu về thời tiết và giao thông trên các tuyến đường và thời gian cụ thể, được sử dụng để đưa ra dự đoán về tình hình thời tiết và giao thông trong 1 giờ sắp tới.

- **Thingspeak:** Thu nhận data gửi lên từ thiết bị IoT, biểu diễn data dưới dạng các biểu đồ theo thời gian thực, refresh mỗi 15 giây.

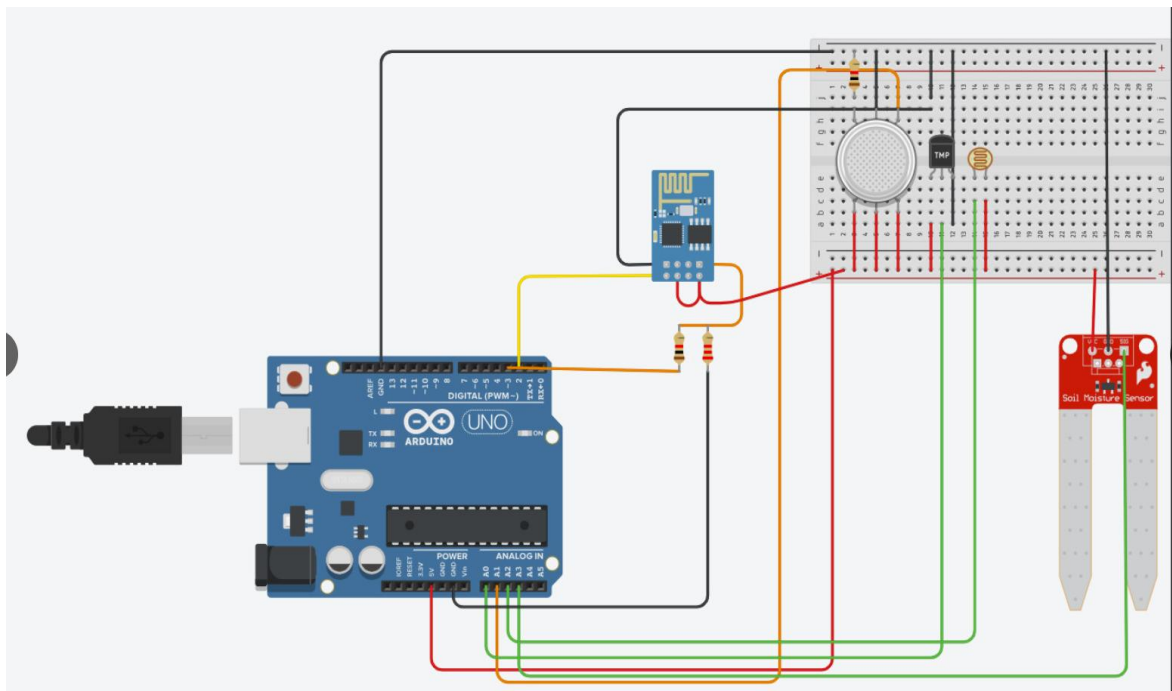
- **Firebase:** Chứa data dự đoán đầu ra, bao gồm số liệu của các cảm biến hiện tại và số liệu dự đoán trong tương lai.

#### 5.1.3 Xây dựng app

Xây dựng một app mobile hiển thị các báo cáo tình hình thời tiết và giao thông hiện tại ở khu vực cũng như các dự báo sắp tới. App cũng sẽ hiển thị các biểu đồ cho data thu thập được tương tự như trên Thingspeak.

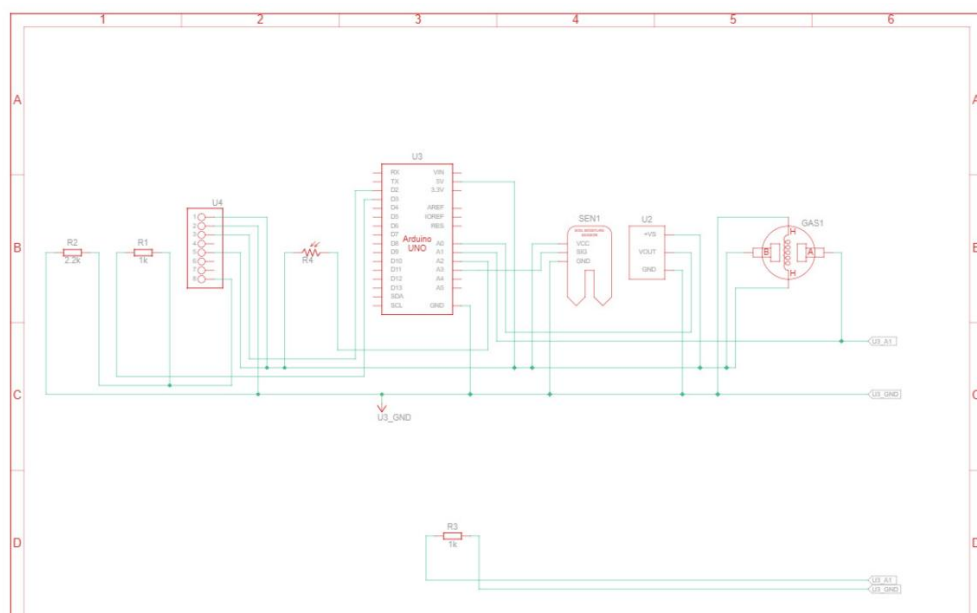
Hệ thống hoàn thiện sẽ kiểm thử lại nhiều lần trên thực tế để đánh giá độ chính xác, sau đó tiến hành sửa đổi cải thiện cho tới khi đạt độ tin cậy mong muốn.

### 5.2 Mô phỏng trên TinkerCad



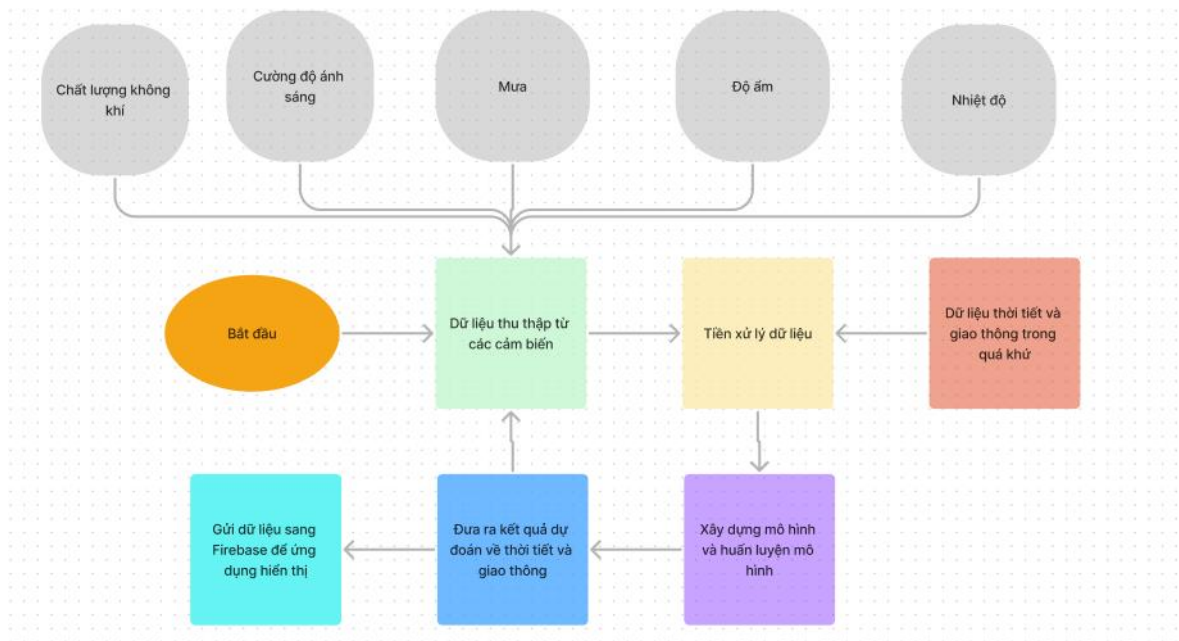
Hình 14: Mô phỏng trên TinkerCad

### 5.3 Mạch điện mô phỏng



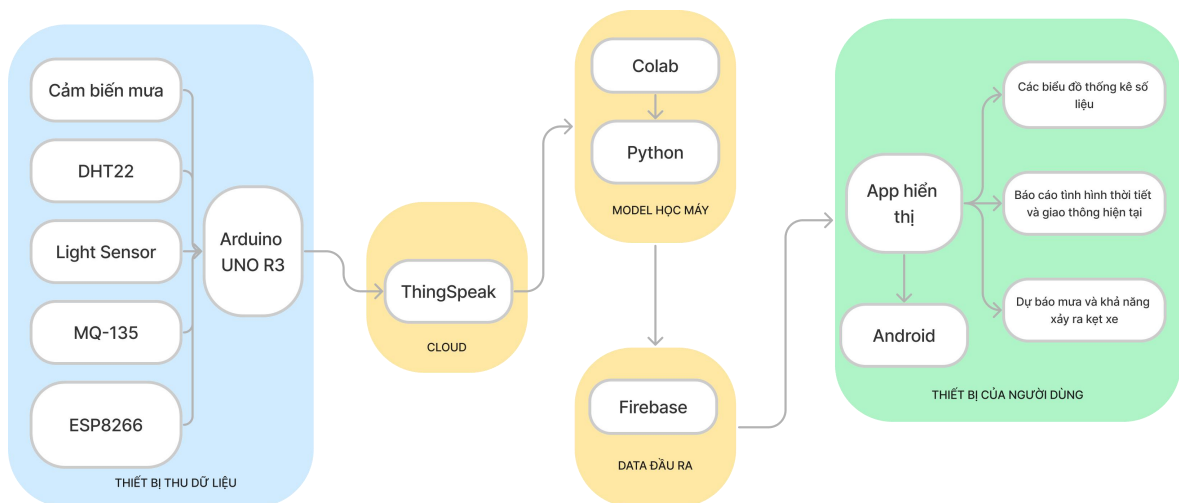
Hình 15: Mạch điện mô phỏng

### 5.4 Lưu đồ xử lý



Hình 16: Lưu đồ xử lý

## 5.5 Quá trình xử lý



Hình 17: Xử lý cụ thể hệ thống

## CHƯƠNG 6: KẾT NỐI THIẾT BỊ THẬT

### 6.1 Các thông tin cần xác định

- **Cảm biến Mưa (Rain):**

Đọc giá trị từ cảm biến mưa (rainAnalogPin và rainDigitalPin).

In giá trị analog và digital lên Serial.

In trạng thái mưa hoặc không mưa dựa trên giá trị digital.

- **Cảm biến Ánh sáng TEMT6000:**

Đọc giá trị từ cảm biến ánh sáng.

Chuyển đổi giá trị analog sang phần trăm và in lên Serial.

- **Cảm biến Nhiệt độ và Độ ẩm DHT11:**

Đọc giá trị nhiệt độ và độ ẩm từ cảm biến DHT.

In giá trị lên Serial.

- **Cảm biến Khí MQ135:**

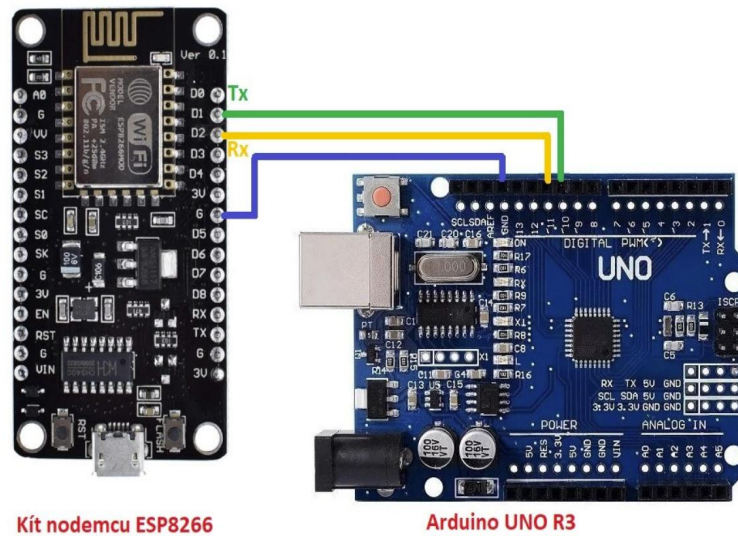
Đọc các giá trị từ cảm biến MQ135 (RZero, Corrected RZero, Resistance, PPM, và Corrected PPM).

In các giá trị lên Serial.

- **Gửi dữ liệu lên Serial và cổng nối tiếp ảo:**

Gán giá trị nhiệt độ, độ ẩm, giá trị digital từ cảm biến mưa, giá trị ánh sáng, và corrected PPM từ cảm biến khí vào đối tượng JSON (doc).

Sử dụng serializeJson để chuyển đổi đối tượng JSON thành chuỗi và in lên cả Serial và cổng nối tiếp ảo (mySerial).



Hình 18: Kết nối giữa Arduino Uno R3 và Kit MCU 8266

## 6.2 Nối mạch và sản phẩm hoàn thiện

### Kết nối Cảm Biến Mưa (Rain Sensor):

- Chân analog của cảm biến mưa (rainAnalogPin) được kết nối với chân A0 trên Arduino.
- Chân digital của cảm biến mưa (rainDigitalPin) được kết nối với chân số 2 trên Arduino.

### Kết nối Cảm Biến Ánh Sáng TEMT6000:

Chân analog của cảm biến ánh sáng (LIGHTSENSORPIN) được kết nối với chân A1 trên Arduino.

### Kết nối Cảm Biến Nhiệt Độ và Độ Ẩm DHT11:

Chân dữ liệu (DHTPIN) của cảm biến DHT11 được kết nối với chân số 4 trên Arduino.

### Kết nối Cảm Biến Khí MQ135:

Chân analog của cảm biến khí (PIN\_MQ135) được kết nối với chân A2 trên Arduino.

### Kết nối Cổng Nối Tiếp Ảo (SoftwareSerial):

- Chân RX (RX\_PIN, chân 10 theo mã nguồn của bạn) của cổng nối tiếp ảo được kết nối với chân TX của một module giao tiếp (ví dụ: module Bluetooth hoặc ESP8266).
- Chân TX (TX\_PIN, chân 11 theo mã nguồn của bạn) của cổng nối tiếp ảo được kết nối với chân RX của module giao tiếp.

#### **Nối 5V từ Arduino:**

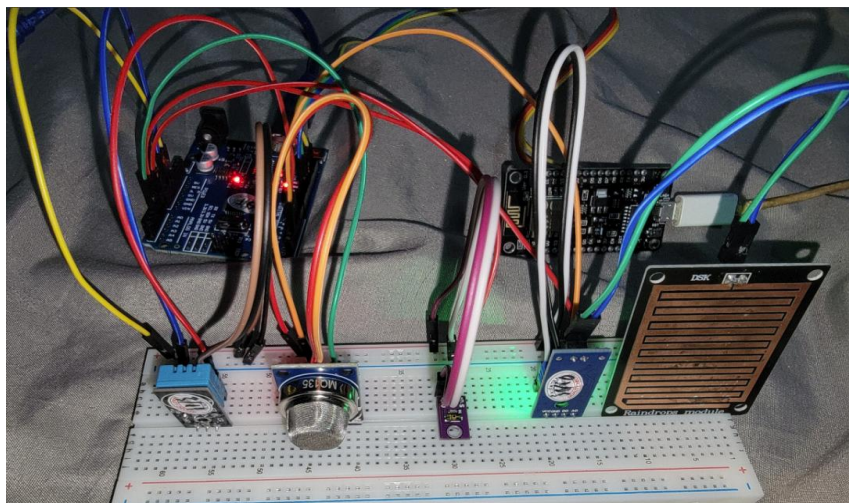
- Chèn một dây nối (ví dụ: dây màu đỏ) từ chân 5V trên Arduino vào một cột (hàng) trên breadboard.
- Kết nối đầu khác của dây nối này với các linh kiện mà bạn cần cung cấp nguồn 5V, chẳng hạn như cảm biến ánh sáng, cảm biến nhiệt độ và độ ẩm, cảm biến khí, vv.

#### **Nối GND từ Arduino:**

- Chèn một dây nối (ví dụ: dây màu đen) từ chân GND trên Arduino vào một cột (hàng) khác trên breadboard.
- Kết nối đầu khác của dây nối này với các linh kiện mà bạn cần kết nối với đất (GND).

#### **Kiểm tra kết nối:**

Đảm bảo các linh kiện cảm biến mưa, cảm biến ánh sáng, cảm biến nhiệt độ và độ ẩm, cảm biến khí, đều được kết nối đúng cách với breadboard.



Hình 19: Mạch sau khi hoàn thành xong



## CHƯƠNG 7: HOÀN THIỆN HỆ THỐNG

### 7.1 Data Processing

#### 7.1.1 Tìm dữ liệu để train model

Sau quá trình nghiên cứu và tìm kiếm, do không có tập data có chứa dữ liệu về cả 2 mảng là thời tiết và giao thông, chúng em quyết định tìm từng tập data riêng cho từng mảng, sau đó tiến hành các bước tiền xử lý cần thiết kết hợp với một số data lấy từ API để thu được tập data hoàn chỉnh.

Các tập data tìm kiếm được bao gồm:

**Data của tập tình trạng giao thông (32.000 dòng dữ liệu):**

- Đặc điểm của con đường tại TPHCM:
- Kinh độ và vĩ độ của điểm đầu và điểm cuối đường.
- Cấp độ của con đường.
- Độ dài của con đường.
- Tên đường và các thông tin khác.
- Feature Liên Quan Đến Giao Thông:
- Thời gian đo: chia thành 48 segment một ngày tương ứng với mỗi 30 phút một lần đo.
- Ngày đo trong tuần.
- Level of Service (LOS): Thang đo đánh giá tình trạng lưu thông trên đường.

_id	segment_i	date	weekday	period	LOS	s_node_id	e_node_id	length	street_id	max_veloc	street_level	street_name	street_type	long_snode	lat_snode	long_enode	lat_enode
0	26	#####	4	period_0_A		3.66E+08	3.66E+08	116	32575820			4 Nguyễn...n	tertiary	106.7687	10.84151	106.7693	10.84242
1	33	8/2/2020	6	period_23_C		3.66E+08	3.79E+09	26	32575862			3 Ấ...ng s	secondary	106.762	10.87865	106.7621	10.87881
2	33	8/3/2020	0	period_0_D		3.66E+08	3.79E+09	26	32575862			3 Ấ...ng s	secondary	106.762	10.87865	106.7621	10.87881
3	67	3/9/2021	1	period_9_B		3.66E+08	5.76E+09	7	32575862			3 Ấ...ng s	secondary	106.7684	10.88082	106.7685	10.88077
4	67	#####	1	period_9_B		3.66E+08	5.76E+09	7	32575862			3 Ấ...ng s	secondary	106.7684	10.88082	106.7685	10.88077
5	70	8/2/2020	6	period_23_D		5.82E+09	4.61E+09	8	32575864	40		3 ChẤCu VẤf	secondary	106.659	10.75194	106.659	10.75202
6	70	8/3/2020	0	period_0_E		5.82E+09	4.61E+09	8	32575864	40		3 ChẤCu VẤf	secondary	106.659	10.75194	106.659	10.75202
7	71	8/2/2020	6	period_23_E		3.66E+08	5.74E+09	43	32575869			4 LẤ VẤfn T	unclassified	106.7657	10.77659	106.766	10.77691
8	71	8/3/2020	0	period_0_C		3.66E+08	5.74E+09	43	32575869			4 LẤ VẤfn T	unclassified	106.7657	10.77659	106.766	10.77691
9	161	#####	4	period_0_A		3.2E+09	3.2E+09	21	32575935			3 TẤCn PhẤ primary_l	li	106.7236	10.73362	106.7238	10.7336
10	173	#####	0	period_4_A		2.3E+09	4.64E+09	9	32575943			4 LẤ Minh >	tertiary	106.6516	10.78623	106.6517	10.78625

Hình 20: Data của tập tình trạng giao thông

**Data của tập thời tiết (7000 dòng dữ liệu):**



- Nhiệt độ.
- Độ ẩm.
- Lượng mưa.
- Độ che phủ của mây.
- Và các thông số khác liên quan.

Thời gian đo: 24 lần đo trong một ngày, tương ứng với 1 giờ 1 lần đo mỗi ngày.

time	temperature	relative_humid	apparent_temperature	rain (mm)	weather_cloud_cover (%)
2020-07-0	25.6	96	31.3	0.8	53
2020-07-0	25.5	96	31.4	1.2	55
2020-07-0	25.3	96	30.9	1.1	55
2020-07-0	25.1	97	30.6	1.3	61
2020-07-0	25	97	30.5	0.5	53
2020-07-0	25	97	30.5	1.1	55
2020-07-0	25.2	96	30.7	1	55
2020-07-0	25.8	93	31.5	0.8	53
2020-07-0	26.6	91	31.6	0.1	51

Hình 21: Data của weather

### 7.1.2 Merge dữ liệu để hoàn thành dữ liệu cho tập dữ liệu

Kết hợp dữ liệu: Merge giữa tập dữ liệu về tình trạng giao thông (train) và dữ liệu về thời tiết (weather), tiến hành tiền xử lý dữ liệu để có một cột chung là dateTime, đồng bộ giữa các lần đo từ cả hai tập dữ liệu. Do số lần đo đang là khác nhau (48 lần đo 1 ngày trong tập giao thông và 24 lần đo 1 ngày trong tập thời tiết), nên cần bước tiền xử lý phù hợp. Có 2 hướng chính:

- Augmentation (tăng cường dữ liệu):

Thêm vào tập dữ liệu về thời tiết các lần đo vào giờ lẻ bằng các phương pháp như trung bình các giá trị lân cận.

=> Hướng này tạo ra tập dữ liệu lớn hơn nhưng không phù hợp với kích thước ban đầu của tập dữ liệu về tình trạng giao thông (train), do chiếm tới một nửa tập này - 16000 dòng.

- Loại bỏ dữ liệu đo giờ lẻ của tập train:

Loại bỏ các lần đo vào giờ lẻ của tập dữ liệu về tình trạng giao thông.

=> Cách này phù hợp hơn do sau khi loại bỏ giờ lẻ, tập dữ liệu về giao thông vẫn còn 16000 dòng, đủ để train một model hoàn chỉnh.

Kết quả là một tập dữ liệu với đầy đủ feature và kích thước tương đương giữa tập train và tập weather:

_id	segment_id	date	weekday	LOS	s_node_id	e_node_id	length	street_id	street_level	street_type	long_node	lat_node	time_x	month	temperature	relative_humidity	apparent_rain	weather_c	cloud_cov	season
2	33	8/3/2020	0	D	3.66E+08	3.79E+09	26	32575862	3	secondary	106.762	10.87865	0:00	8	25.5	93	29.9	0.4	51	31 rain
6	70	8/3/2020	0	E	5.82E+09	4.61E+09	8	32575864	3	secondary	106.659	10.75194	0:00	8	25.5	93	29.9	0.4	51	31 rain
8	71	8/3/2020	0	C	3.66E+08	5.74E+09	43	32575869	4	unclassified	106.7657	10.77659	0:00	8	25.5	93	29.9	0.4	51	31 rain
87	432	8/3/2020	0	C	4.98E+09	3.66E+08	42	32576127	4	tertiary	106.6646	10.75103	0:00	8	25.5	93	29.9	0.4	51	31 rain
89	433	8/3/2020	0	C	4.98E+09	4.59E+09	50	32576127	4	tertiary	106.6646	10.75103	0:00	8	25.5	93	29.9	0.4	51	31 rain
91	434	8/3/2020	0	D	4.59E+09	4.98E+09	50	32576127	4	tertiary	106.6645	10.75148	0:00	8	25.5	93	29.9	0.4	51	31 rain
93	435	8/3/2020	0	D	4.59E+09	3.66E+08	89	32576127	4	tertiary	106.6645	10.75148	0:00	8	25.5	93	29.9	0.4	51	31 rain
95	436	8/3/2020	0	C	3.66E+08	4.59E+09	89	32576127	4	tertiary	106.6644	10.75227	0:00	8	25.5	93	29.9	0.4	51	31 rain
97	437	8/3/2020	0	D	3.66E+08	5.82E+09	69	32576127	4	tertiary	106.6644	10.75227	0:00	8	25.5	93	29.9	0.4	51	31 rain

Hình 22: Data cuối cùng để train model

Tiền xử lý dữ liệu

```
# Load your data
data = pd.read_csv('merged_output.csv')
# Identify columns with object dtype
object_cols = data.select_dtypes(include=['object']).columns
# Create a LabelEncoder instance and encode object columns
label_encoder = LabelEncoder()
label_mapping = {}

print(data.tail(10).to_string()+"\n")

for col in object_cols:
    data[col] = label_encoder.fit_transform(data[col])
    label_mapping[col] = dict(zip(label_encoder.classes_, label_encoder.transform(label_encoder.classes_)))

# Save the label mappings
joblib.dump(label_mapping, 'label_mapping.joblib')

# Lựa chọn các cột số cần chuẩn hóa
numeric_columns = ["weekday", "length", "street_level", "long_node", "lat_node", "temperature_2m", "relative_humidity_2m", "cloud_cover", "season"]
data_to_normalize = data[numeric_columns]

# Chuẩn hóa các cột số
scaler = MinMaxScaler()
data[numeric_columns] = scaler.fit_transform(data_to_normalize)

# Hiển thị dữ liệu sau khi chuẩn hóa
print(data.tail(10).to_string())
```

_id	segment_id	date	weekday	LOS	s_node_id	e_node_id	length	street_id	street_level	street_type	long_node	lat_node	time_x	month	temperature_2m	
16597	32991	83729	1/22/2021	4	F	5762890465	5762890475	31	619030684	4	tertiary	106.805389	10.875074	14:00	1	31.3
16598	32974	83727	1/24/2021	6	F	5762890475	5763376688	104	619030684	4	tertiary	106.805656	10.874978	3:00	1	22.9

## 7.2 Mã giả xử lý từng quy trình

### 7.2.1 Đọc dữ liệu từ các sensor

Khai báo thư viện và biến:

- Sử dụng thư viện ArduinoJson để xử lý dữ liệu JSON.
- Sử dụng thư viện SoftwareSerial để tạo cổng nối tiếp ảo.
- Sử dụng thư viện MQ135 và DHT để đọc giá trị từ cảm biến khí CO2 và cảm biến nhiệt độ, độ ẩm.
- Khai báo các chân và biến cần thiết.

```

#include <ArduinoJson.h>
#include <SoftwareSerial.h>
#include <MQ135.h>
#include <DHT.h>

#define rainAnalogPin A0
#define rainDigitalPin 2
#define LIGHTSENSORPIN A1
#define PIN_MQ135 A2 // MQ135 Analog Input Pin
#define DHTPIN 4 // DHT Digital Input Pin
#define DHTTYPE DHT11 // DHT11 or DHT22, depends on your sensor

MQ135 mq135_sensor(PIN_MQ135);
DHT dht(DHTPIN, DHTTYPE);
StaticJsonDocument<300> doc;
#define RX_PIN 10 // Chân RX của cổng nối tiếp mềm
#define TX_PIN 11 // Chân TX của cổng nối tiếp mềm
SoftwareSerial mySerial(RX_PIN, TX_PIN); // Tạo một đối tượng cổng nối tiếp mềm
float temperature,temperatureF, humidity;

```

Hàm setup:

- Khởi tạo chế độ đầu vào cho các chân cảm biến.
- Bắt đầu giao tiếp với cảm biến DHT.
- Khởi tạo cổng nối tiếp và cổng nối tiếp ảo.

```

void setup() {
  pinMode(rainAnalogPin, INPUT);
  pinMode(rainDigitalPin, INPUT);
  pinMode(LIGHTSENSORPIN, INPUT);
  dht.begin();
  Serial.begin(9600);
  mySerial.begin(9600);
}

```

Hàm loop:

- Đọc giá trị từ cảm biến mưa (rainAnalogPin và rainDigitalPin) và hiển thị trên Serial Monitor.

```
//-----RAIN-----
int rainAnalogValue = analogRead(rainAnalogPin);
Serial.print("Rain analog: ");
Serial.print(rainAnalogValue);
int rainDigitalValue = digitalRead(rainDigitalPin);
Serial.print(", Rain digital: ");
Serial.println(rainDigitalValue);
Serial.print("Weather: ");
if (rainDigitalValue == LOW)
{
  Serial.println("Rain Detected");
} else {
  Serial.println("No Rain Detected");
}
```

- Đọc giá trị từ cảm biến ánh sáng và tính toán tỷ lệ ánh sáng, sau đó hiển thị kết quả.

```
//-----TEMT6000-----
float readLight = analogRead(LIGHTSENSORPIN); //Read light level
float square_ratio = readLight / 1023.0; //Get percent of maximum value (1023)
square_ratio = pow(square_ratio, 2.0);
Serial.print("Light analog: ");
Serial.print(readLight);
Serial.print(", Percent light: ");
Serial.println(square_ratio);
```

- Đọc giá trị nhiệt độ và độ ẩm từ cảm biến DHT11, hiển thị kết quả và kiểm tra lỗi.

```
//-----DHT11-----
humidity = dht.readHumidity();
temperature = dht.readTemperature(false);
temperatureF = dht.readTemperature(true);
Serial.print("Temperature (C): (");
Serial.print(temperature);
Serial.print(" C, ");
Serial.print(temperatureF);
Serial.print(" F), Humidity: ");
Serial.print(humidity);
Serial.println("%");
if (isnan(humidity) || isnan(temperature) || isnan(temperatureF)) {
  Serial.println(F("Failed to read from DHT sensor!"));
  return;
}
```

- Đọc giá trị từ cảm biến khí CO2 (MQ135) và hiển thị các thông số như RZero, PPM, Resistance, và các giá trị đã được chỉnh sửa.

```
//-----MQ135-----
float rzero = mq135_sensor.getRZero();
float correctedRZero = mq135_sensor.getCorrectedRZero(temperature, humidity);
float resistance = mq135_sensor.getResistance();
float ppm = mq135_sensor.getPPM();
float correctedPPM = mq135_sensor.getCorrectedPPM(temperature, humidity);

Serial.print("MQ135 RZero: ");
Serial.print(rzero);
Serial.print(", Corrected RZero: ");
Serial.print(correctedRZero);
Serial.print(", Resistance: ");
Serial.print(resistance);
Serial.print(", PPM: ");
Serial.print(ppm);
Serial.print("ppm");
Serial.print(", Corrected PPM: ");
Serial.print(correctedPPM);
Serial.println("ppm");
```

- Gán giá trị vào đối tượng JsonDocument.

```
// Gán giá trị nhiệt độ và độ ẩm vào đối tượng JsonDocument
doc["temperature"] = temperature;
doc["humidity"] = humidity;
doc["rainDigitalValue"] = rainDigitalValue;
doc["readLight"] = readLight;
doc["correctedPPM"] = correctedPPM;
serializeJson(doc, mySerial);
mySerial.println();
serializeJson(doc, Serial);
Serial.println();
doc.clear();
```

- Gửi dữ liệu JSON qua cổng nối tiếp ảo và Serial.
- Delay 15 giây trước khi đo lại.

### 7.2.2 Arduino sử dụng ESP8266 (ESP-01) để đọc dữ liệu từ cảm biến

Khai báo thư viện và biến:

- Sử dụng thư viện ArduinoJson để xử lý dữ liệu JSON.
- Sử dụng thư viện SoftwareSerial để tạo cổng nối tiếp ảo.
- Sử dụng thư viện MQ135 và DHT để đọc giá trị từ cảm biến khí CO2 và cảm biến nhiệt độ, độ ẩm.
- Khai báo các chân và biến cần thiết.



```
#include <ArduinoJson.h>
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>

String apiKey = "G7A7420ZT1ZEB730"; // Enter your Write API key from ThingSpeak
const char *ssid = "5k/1gb-laptop"; // replace with your wifi ssid and wpa2 key
const char *pass = "12345679";
const char *server = "api.thingspeak.com";

WiFiClient client;
StaticJsonDocument<300> doc;

#define RX_PIN 4 // Chân RX của EspSoftwareSerial
#define TX_PIN 5 // Chân TX của EspSoftwareSerial
SoftwareSerial mySerial(RX_PIN, TX_PIN); // RX, TX
```

Hàm loop:

- Bắt đầu mỗi chu kỳ lặp bằng việc đặt đèn LED built-in ở trạng thái HIGH.

```
digitalWrite(LED_BUILTIN, HIGH);
if (mySerial.available()) {
    String line = mySerial.readStringUntil('\n');
    DeserializationError error = deserializeJson(doc, line);
    if (error) {
        Serial.print("deserializeJson() failed: ");
        Serial.println(error.c_str());
        return;
    }
}
```

- Nếu có dữ liệu trên cổng nối tiếp ảo, đọc dữ liệu và giải mã JSON.

```
float temperature = doc["temperature"];
float humidity = doc["humidity"];
float rainDigitalValue = doc["rainDigitalValue"];
float readLight = doc["readLight"];
float correctedPPM = doc["correctedPPM"];
```

- Đọc giá trị từ JSON và gửi chúng lên ThingSpeak thông qua HTTP POST request.
- Hiển thị giá trị đọc được trên Serial Monitor và đèn LED built-in sẽ nhấp nháy.
- Delay 2 giây để tránh gửi dữ liệu quá nhanh và giảm lượng request tới ThingSpeak.
- Kết thúc vòng lặp, đèn LED sẽ chuyển về trạng thái HIGH và chờ cho chu kỳ lặp tiếp theo.

### 7.2.3 Train mô hình bằng python

#### Xây dựng mô hình và huấn luyện mô hình

```
# Chuẩn bị dữ liệu
x = data[["weekday", "length", "street_level", "long_node", "lat_node", "temperature_2m", "relative_humidity_2m", "cloud_cover", "season"]]
y = data[["rain", "weather_code", "LOS"]] # Đây là cột chứa nhiệt độ sau 30 phút

# X = data[[26.7, 53, , "Air Quality"]]
# 26.7 53 1 64 0.13
# y = data["Rain"] # Đây là cột chứa nhiệt độ sau 30 phút
test_size=len(X)-50

# Chia dữ liệu
X_train, X_test = X[:test_size], X[test_size:]
y_train, y_test = y[:test_size], y[test_size:]

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

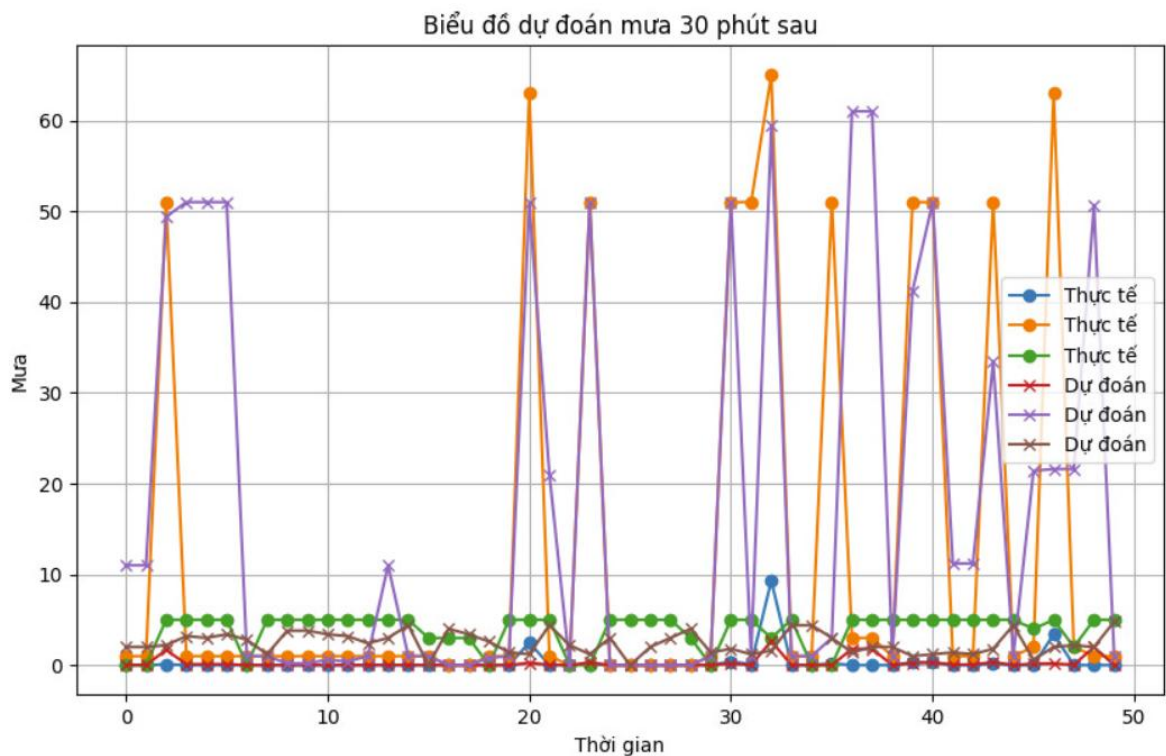
# Tạo và huấn luyện mô hình
# Mô hình Random Forest Regression
model = RandomForestRegressor(n_estimators=5, random_state=42)
model.fit(X_train, y_train)
joblib.dump(model, 'rf_model.joblib')

y_pred = model.predict(X_test)

# Đánh giá mô hình
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Hiển thị kết quả
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

Mean Squared Error: 172.54000533333337  
R-squared: -0.21080462284826199



Hình 23: Biểu đồ dự đoán mưa sau 30 phút tới

## 7.2.4 Đẩy dữ liệu kết quả lên Firebase

Vòng lặp cập nhật dữ liệu

```
# Thiết lập kết nối tới Firebase
if not firebase_admin.apps:
    cred = credentials.Certificate("serviceAccountKey.json")
    firebase_admin.initialize_app(cred, {
        'databaseURL': 'https://weather-forecast-13899-default-rtdb.asia-southeast1.firebaseio.com/'
    })
ref = db.reference()
# Dự đoán nhiệt độ 30 phút sau trên tập kiểm tra
while True:

    # Các thông tin cần thiết
    channel_id = '2332757'
    # URL của Thingspeak
    fields=[]
    url = f'https://api.thingspeak.com/channels/{channel_id}/feeds.json?results=5'
    # Gửi dữ liệu lên Thingspeak
    create_at=""
    response = requests.get(url)
    if response.status_code == 200:
        res = response.json()
        # Xử lý dữ liệu ở đây
        # print(res['feeds'][0])
        fields = []
        feed=res['feeds'][0]
        for key in feed:
            if key=='created_at':
                create_at=feed[key]
            if key.startswith('field'):
                fields.append(feed[key])
    else:
        continue
    # Sử dụng hàm
    new_hour, new_minute, new_second = add_hours(create_at)
    # Tạo dữ liệu mới từ dòng cuối cùng của DataFrame
    if (new_hour>7 and new_hour<18):
```

The screenshot shows the Firebase Realtime Database interface. The left sidebar contains navigation options like Project Overview, Realtime Database, and Analytics. The main area displays the 'weather\_data' node with a tree view of its fields: hum: 84, light: 2, los: 0, ppm: 0.93, rain: 0, rain\_pred: 0, temp: 28, and weather\_code: 0. The database URL is https://weather-forecast-13899-default-rtdb.asia-southeast1.firebaseio.com/ and the location is Singapore (asia-southeast1).

Hình 24: Dữ liệu lên Firebase

## 7.2.5 Get dữ liệu qua Android Studio



```

for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
    WeatherData weatherData = snapshot.getValue(WeatherData.class);

    if (weatherData != null) {
        // Access data using getters from the model class
        double rainPred = weatherData.getRain_pred();
        int weatherCode = weatherData.getWeather_code();
        double los = weatherData.getLos();
        double temp = weatherData.getTemp();
        double hum = weatherData.getHum();
        int rain = weatherData.getRain();
        double light = weatherData.getLight();
        double ppm = weatherData.getPpm();
    }
}

public WeatherData(Double hum, Double light, Double los, Double ppm, int rain, Double rain_pred, Double temp, int weather_code) {
    this.hum = hum;
    this.light = light;
    this.los = los;
    this.ppm = ppm;
    this.rain = rain;
    this.rain_pred = rain_pred;
    this.temp = temp;
    this.weather_code = weather_code;
}

```

Hình 25: Lấy dữ liệu ở android app

Nhiệt độ (°C)	Đánh giá
$\geq 37$	Rất nóng
$27 \leq X < 37$	Nóng
$22 \leq X < 27$	Dễ chịu
$12 \leq X < 22$	Lạnh
$\leq 12$	Rất lạnh

Bảng 2: Đánh giá thời tiết qua nhiệt độ

Độ ẩm (%)	Đánh giá
$\geq 70$	Rất ẩm
$60 \leq X < 70$	Tương đối ẩm

$50 \leq X < 60$	Ổn định
$40 \leq X < 50$	Khá khô
$\leq 40$	Rất khô

Bảng 3: Đánh giá thời tiết qua độ ẩm

Chất lượng không khí (ppm)	Đánh giá
$\geq 1000$	Rất xấu
$500 \leq X < 1000$	Xấu
$200 \leq X < 500$	Tạm ổn
$100 \leq X < 200$	Tốt
$\leq 100$	Rất tốt

Bảng 4: Đánh giá thời tiết qua chất lượng không khí

Lượng mưa (mm/12h)	Đánh giá
$X \leq 0$	Khô ráo
$0 < X < 3$	Mưa nhỏ
$3 \leq X < 25$	Mưa vừa
$25 \leq X < 50$	Mưa to
$X \geq 50$	Mưa rất to

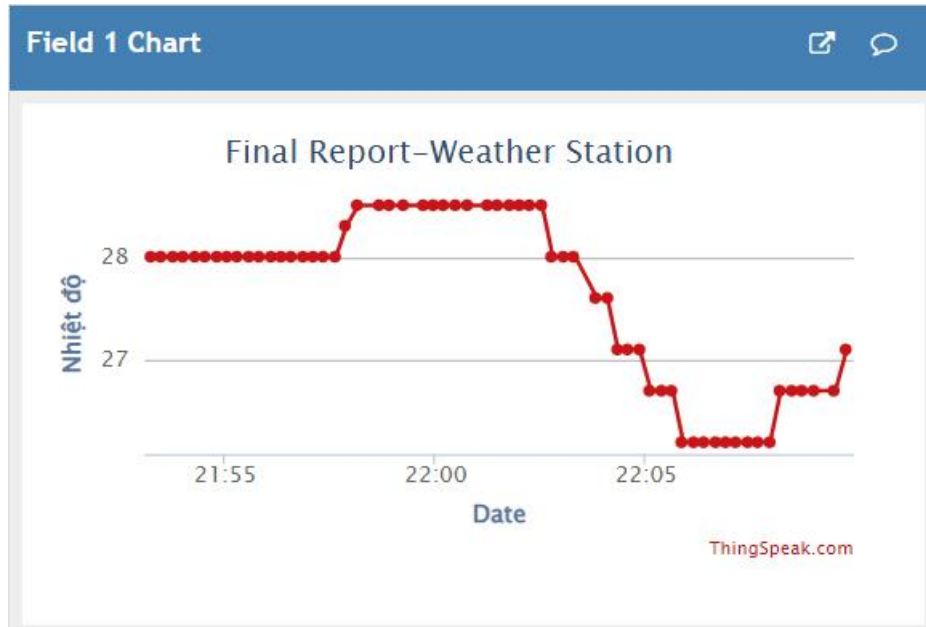
Bảng 5: : Đánh giá thời tiết qua lượng mưa

Tình hình giao thông (Level of Service)	Đánh giá
0	Tình trạng giao thông rất tốt.
1	Tình trạng giao thông tốt, có một ít tắc nghẽn nhẹ.
2	Tình trạng giao thông ổn định, tốc độ giảm nhẹ.
3	Tình trạng giao thông tắc nghẽn.
4	Tình trạng giao thông kém, tắc nghẽn nhiều.

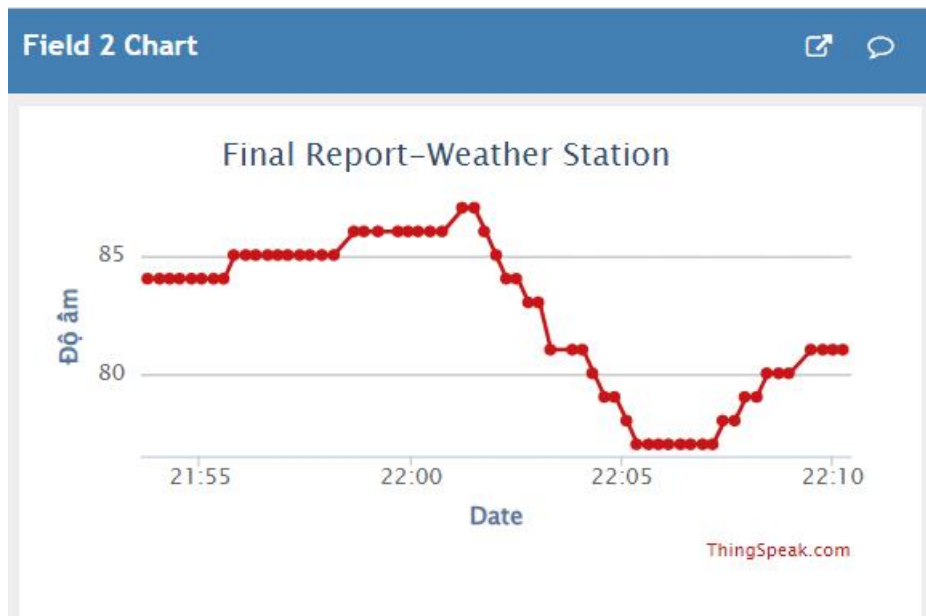
5	Tình trạng giao thông rất xấu. Tắc nghẽn
---	--

Bảng 6: Đánh giá tình hình kẹt xe tại khu vực

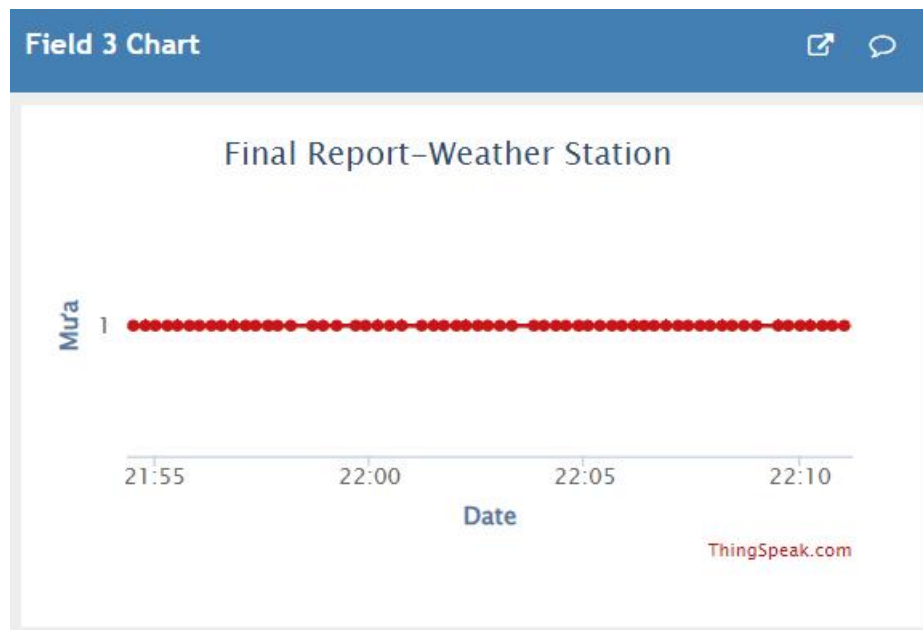
### 7.3 Dữ liệu từ sensor được đưa lên Thingspeak



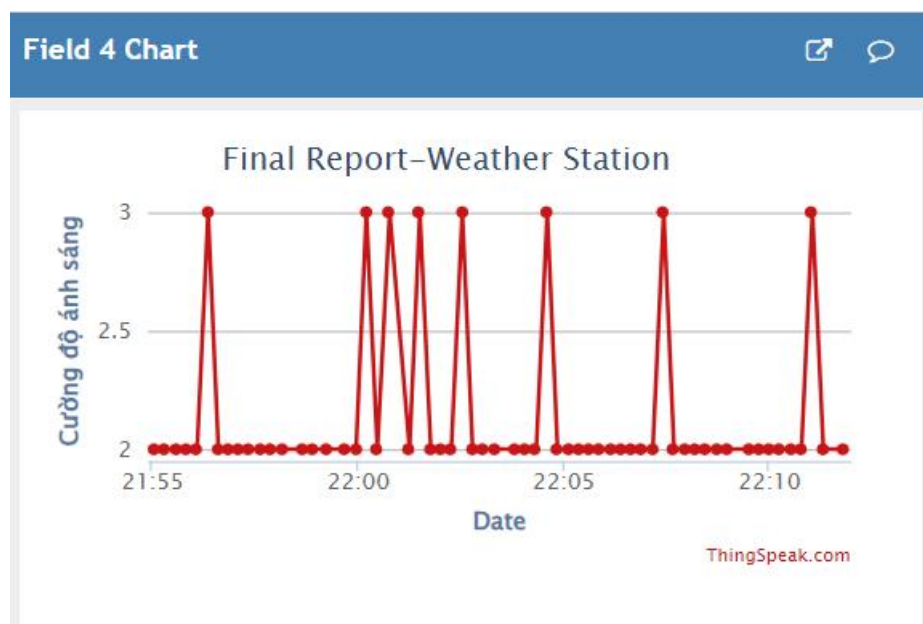
Hình 26: Nhiệt độ



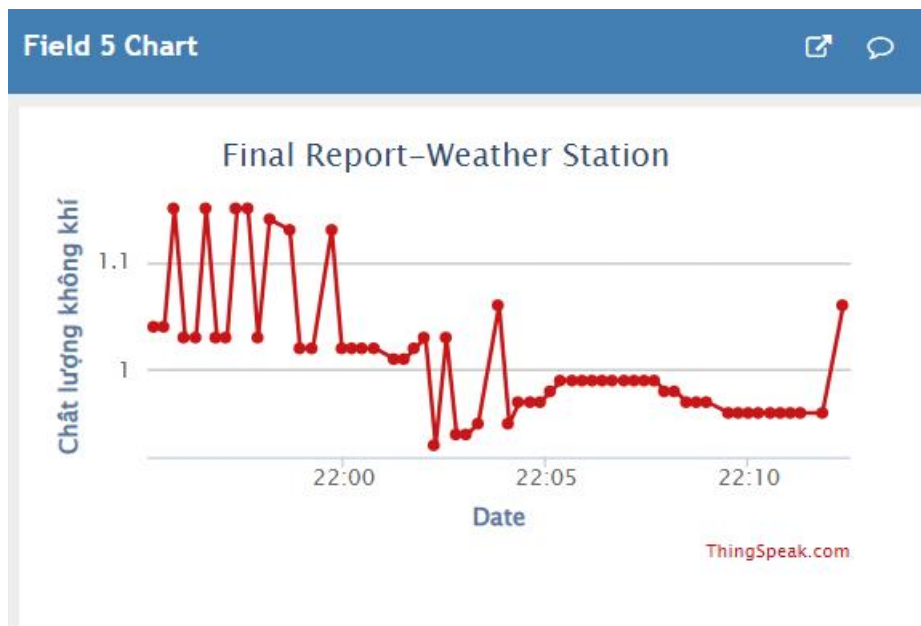
Hình 27: Độ ẩm



Hình 28: Mưa



Hình 29: Cường độ ánh sáng



Hình 30: Chất lượng không khí

## 7.4 Kết quả demo của ứng dụng

### 7.3.1 Trang chủ của ứng dụng



Hình 31: Trang chủ của ứng dụng

### 7.3.2 Trang thống kê của hệ thống



Hình 32: Biểu đồ hiển thị các thông số trong ứng dụng



Hình 33: Biểu đồ hiển thị các thông số trong ứng dụng

## TÀI LIỆU THAM KHẢO

### Tiếng việt

[1] Lavender, Machine Learning - Thử làm Nhà Thiên Văn Dự Báo Thời Tiết, 30/1/2018, [https://viblo.asia/p/machine-learning-thu-lam-nha-thien-van-du-bao-thoi-tiet-djeZ1xYmKWz#\\_tong-ket-4](https://viblo.asia/p/machine-learning-thu-lam-nha-thien-van-du-bao-thoi-tiet-djeZ1xYmKWz#_tong-ket-4), Ngày truy cập 27/10/2023.

### Tiếng Anh

[2] "Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry" by Maciej Kranz, 2016.

[3] "Internet of Things: A Hands-On Approach" by Arshdeep Bahga, Vijay Madisetti, 2014.

[4] "Designing the Internet of Things" by Adrian McEwen, Hakim Cassimally, 2014.

[5] "The Internet of Things (The MIT Press Essential Knowledge series)" by Samuel Greengard, 2015.

[6] "Hands-On IoT Solutions with Blockchain: Build effective blockchain IoT projects using Ethereum, Hyperledger, and Stellar" by Nikhil Buduma, 2019.

[7] Neverofftheinternet, ThingSpeak Arduino Weather Station, 4/4/2020, <https://www.hackster.io/neverofftheinternet/thingspeak-arduino-weather-station-70b4bb>, accessed on 26/10/2023.

[8] NCD, ThingSpeak Weather APP using ESP8266 And SHT31, <https://ncd.io/blog/thingspeak-weather-app-using-esp8266/>, accessed on 27/10/2023.

[9] Shruti Dadhich, Vibhakar Pathak, Rohit Mittal, Ruchi Doshi, Machine learning for weather forecasting, [https://books.google.com.vn/books?hl=vi&lr=&id=Lo89EAAQBAJ&oi=fnd&pg=PA161&dq=weather+forecasting+iot+machine+learning&ots=iv0OG53Vuq&sig=02QftyKsDSFWcJxPzLbSuFS-0zI&redir\\_esc=y#v=onepage&q=weather%20forecasting%20iot%20machine%20learning&f=false](https://books.google.com.vn/books?hl=vi&lr=&id=Lo89EAAQBAJ&oi=fnd&pg=PA161&dq=weather+forecasting+iot+machine+learning&ots=iv0OG53Vuq&sig=02QftyKsDSFWcJxPzLbSuFS-0zI&redir_esc=y#v=onepage&q=weather%20forecasting%20iot%20machine%20learning&f=false), Accessed on 26/10/2023.