

**TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



# **BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

***Đề tài: “Ứng dụng IoT vào hệ thống cảnh báo cháy”***

<b>Người hướng dẫn</b>	<b>:</b>	<b>TS. NGUYỄN XUÂN SÂM</b>
<b>Sinh viên thực hiện</b>	<b>:</b>	<b>LÊ NGUYỄN CHÁNH TÍN</b>
<b>Mã số sinh viên</b>	<b>:</b>	<b>N14DCCN112</b>
<b>Lớp</b>	<b>:</b>	<b>D14CQMT01-N</b>
<b>Khoá</b>	<b>:</b>	<b>2014</b>
<b>Hệ</b>	<b>:</b>	<b>ĐẠI HỌC</b>

**TP.HCM, tháng 7 /2018**

**TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG**  
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

---



**BÁO CÁO THỰC TẬP**  
**TỐT NGHIỆP ĐẠI HỌC**

***Đề tài: “Ứng dụng IoT vào hệ thống cảnh báo cháy”***

**Người hướng dẫn** : TS. NGUYỄN XUÂN SÂM  
**Sinh viên thực hiện** : LÊ NGUYỄN CHÁNH TÍN  
**Mã số sinh viên** : N14DCCN112  
**Lớp** : D14CQMT01-N  
**Khoá** : 2014  
**Hệ** : ĐẠI HỌC

**TP.HCM, tháng 7 /2018**

HỌC VIỆN  
CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
CƠ SỞ TẠI TP. HỒ CHÍ MINH  
**Khoa: CNTT2**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
**Độc lập – Tự do – Hạnh phúc**  
*TP. Hồ Chí Minh, ngày.....tháng.....năm 2018.....*

## PHIẾU ĐĂNG KÝ THỰC TẬP TỐT NGHIỆP

**Kính gửi:** Lãnh đạo khoa CNTT2

**1. Họ và tên sinh viên:** Lê Nguyễn Chánh Tín **Mã SV:** N14DCCN112

Lớp: D14CQMT01-N Ngành: Công nghệ thông tin Hình thức đào tạo: Đại học chính qui

**2. Nội dung thực tập chính:** Ứng dụng IoT vào hệ thống cảnh báo cháy

**3. Nơi đăng ký thực tập:**

Đơn vị chủ quản: Công ty cổ phần công nghệ Bách Hưng Khang

Đơn vị cơ sở tiếp nhận thực tập: Bộ phận Dịch vụ

Địa chỉ: Số 447B Lê Văn Việt phường Tăng Nhơn Phú A quận 9

Số ĐT: 0968511711 Số Fax:.....

**4. Đề cương thực tập:**

- **Lý thuyết:**

- + Sơ lược về công nghệ IoT, mô hình ứng dụng của hệ thống cảnh báo cháy.
- + Sơ lược về công nghệ NodeJS.
- + Quy trình hoạt động của hệ thống cảnh báo cháy.
- + Các linh kiện, thiết bị cần thiết của hệ thống

- **Thực hành:**

- + Triển khai mô hình hệ thống cảnh báo cháy.
- + Xây dựng website hiển thị thông tin của hệ thống bằng NodeJS.
- + Xây dựng ứng dụng Android hiển thị thông tin cảm biến.

**5. Giáo viên hướng dẫn:** TS. Nguyễn Xuân Sâm

**6. Thời gian thực hiện:** Từ ngày 25 tháng 6 năm 2018 đến ngày 05 tháng 8 năm 2018

TRƯỞNG BỘ MÔN

GIÁO VIÊN HƯỚNG DẪN

SINH VIÊN ĐĂNG KÝ

TS. Nguyễn Xuân Sâm

Lê Nguyễn Chánh Tín

## LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô, gia đình và bạn bè.

Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý Thầy Cô ở Khoa Công nghệ thông tin - Học viện Công nghệ Bưu chính Viễn Thông sơ sở TP HCM đã cùng với trí thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Và đặc biệt, trong học kỳ vừa qua, Khoa đã tổ chức cho chúng em được tiếp cận với môn học mà theo em là rất hữu ích đối với riêng cá nhân em. Đó là môn học “Các hệ thống phân tán”.

Em xin chân thành cảm ơn TS. Nguyễn Xuân Sâm đã tận tâm hướng dẫn chúng em qua từng buổi học trên lớp cũng như những buổi nói chuyện, thảo luận về lĩnh vực IoT. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì em nghĩ bản báo cáo này của em rất khó có thể hoàn thiện được. Một lần nữa, em xin chân thành cảm ơn thầy.

Bước đầu đi vào thực tế, tìm hiểu về lĩnh vực còn khá mới, kiến thức của em còn hạn chế và còn nhiều ngỡ ngàng. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô và các bạn học cùng lớp để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

Cuối cùng, em xin chân thành cảm ơn các anh trong bộ phận dịch vụ đã hỗ trợ, hướng dẫn em trong thời gian qua để em có thể hoàn thành tốt được đề tài thực tập của mình.

# MỤC LỤC

	Trang
<b>MỞ ĐẦU</b> .....	1
<b>CHƯƠNG 1. GIỚI THIỆU VỀ INTERNET OF THINGS (IOT)</b> .....	2
1.1. GIỚI THIỆU VỀ INTERNET OF THINGS (IOT) .....	2
1.2. KHẢ NĂNG ĐỊNH DANH .....	3
1.3. TÍNH CHẤT CỦA IOT .....	3
1.3.1. Thông minh.....	3
1.3.2. Kiến trúc dựa trên sự kiện .....	3
1.3.3. Là một hệ thống phức tạp .....	3
1.3.4. Kích thước.....	3
1.3.5. Vấn đề không gian, thời gian .....	4
1.4. ỨNG DỤNG CỦA IOT .....	4
1.5. NHỮNG TÁC NHÂN ẢNH HƯỞNG ĐẾN SỰ PHÁT TRIỂN CỦA IOT .....	5
1.5.2. Hàng rào subnetwork .....	5
1.5.3. Có quá nhiều "ngôn ngữ địa phương" .....	6
1.5.4. Tiền và chi phí .....	6
1.6. Các "hầm chứa" tập trung hay những "hòn đảo Internet"? .....	6
<b>CHƯƠNG 2. TỔNG QUAN VỀ NODEJS</b> .....	7
2.1. TỔNG QUAN VỀ NODEJS .....	7
2.2. ƯU ĐIỂM CỦA NODEJS.....	7
2.2.1. JSON APIs .....	7
2.2.2. Ứng dụng trên 1 trang .....	7
2.2.3. Shelling tools unix .....	7
2.2.4. Streaming Data (Luồng dữ liệu).....	8
2.2.5. Ứng dụng Web thời gian thực.....	8
2.3. NHƯỢC ĐIỂM CỦA NODEJS.....	8
2.3.1. Ứng dụng nặng, tốn tài nguyên.....	8
2.3.2. Tính năng hỗ trợ .....	8
2.4. ỨNG DỤNG CỦA NODEJS TRONG ĐỀ TÀI THỰC TẬP .....	8
<b>CHƯƠNG 3. HỆ THỐNG GIÁM SÁT, CẢNH BÁO CHÁY</b> .....	9
3.1. CÁC THÀNH PHẦN SỬ DỤNG TRONG HỆ THỐNG.....	9
3.1.1. Phần cứng .....	9
3.1.2. Phần mềm .....	13
3.2. NGUYÊN LÝ HOẠT ĐỘNG CỦA HỆ THỐNG.....	14

<b>3.2.1. Nguyên lý hoạt động của ESP8266 .....</b>	<b>15</b>
<b>3.2.2. Nguyên lý hoạt động của cảm biến MQ – 2 .....</b>	<b>17</b>
<b>3.2.3. Nguyên lý hoạt động của cảm biến DHT22 .....</b>	<b>18</b>
<b>3.2.4. Nguyên lý hoạt động của Realtime Database của FireBase .....</b>	<b>18</b>
<b>3.2.5. Nguyên lý hoạt động của ThingSpeak.....</b>	<b>18</b>
<b>3.2.6. Nguyên lý hoạt động của Website.....</b>	<b>18</b>
<b>3.2.7. Nguyên lý hoạt động của ứng dụng “Cảnh báo cháy” .....</b>	<b>18</b>
<b>3.3. CÁC YÊU CẦU CẦN THIẾT CỦA HỆ THỐNG .....</b>	<b>19</b>
<b>3.4. ĐÁNH GIÁ HỆ THỐNG.....</b>	<b>19</b>
<b>3.4.1. Ưu điểm.....</b>	<b>19</b>
<b>3.4.2. Khuyết điểm.....</b>	<b>19</b>
<b>KẾT LUẬN .....</b>	<b>20</b>
<b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b>	<b>21</b>

## **DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH**

Hình 1.1. Mô hình Internet of Things (IoT). ....	2
Hình 1.2. Nền tảng IoT do nhóm sinh viên Việt Nam sản xuất: iNut. ....	4
Hình 2.1. NodeJS.....	7
Hình 3.1. ESP8266 (NodeMCU).....	9
Hình 3.2. Cảm biến khí MQ-2.....	10
Hình 3.3. Cảm biến nhiệt độ, độ ẩm DHT22.....	10
Hình 3.4. Mặt trước của thiết bị.....	11
Hình 3.5. Mặt sau của thiết bị.....	11
Hình 3.6. Cạnh phải của thiết bị. ....	12
Hình 3.7. Cạnh trái của thiết bị.....	12
Hình 3.8. Website hiển thị thông số cảm biến.....	13
Hình 3.9. Biểu đồ hiển thị thông số môi trường. ....	13
Hình 3.10. Ứng dụng “Cảnh báo cháy” trên thiết bị Android. ....	14
Hình 3.11. Lưu đồ giải thuật hiển thị tình trạng môi trường của hệ thống.....	14
Hình 3.12. Sơ đồ nguyên lý hoạt động của hệ thống cảnh báo cháy. ....	15
Hình 3.13. Sơ đồ nguyên lý hoạt động của ESP8266 (1). ....	15
Hình 3.14. Các biến cần thiết để kết nối ESP8266 và FireBase.....	16
Hình 3.15. Các thức kết nối và truyền dữ liệu đến Firebase.....	16
Hình 3.16. Sơ đồ nguyên lý hoạt động của ESP8266 (2). ....	16
Hình 3.17. Các biến cần thiết để ESP8266 gửi dữ liệu về ThingSpeak. ....	17
Hình 3.18. Cách để ESP8266 kết nối và gửi dữ liệu đến ThingSpeak.. ....	17

## DANH MỤC CÁC CỤM TỪ VIẾT TẮT

API: Application Programming Interface	Giao diện lập trình ứng dụng
CSDL: Cơ sở dữ liệu	Cơ sở dữ liệu
FTP: File Transfer Protocol	Giao thức truyền tập tin
HCM: Hồ Chí Minh	Địa danh
HTTP: HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
IMAP: Internet Message Access Protocol	Giao thức lấy thư điện tử từ mail server
IoT: Internet of Things	Internet của vạn vật
IP: Internet Protocol	Giao thức Internet, địa chỉ của thiết bị trên môi trường Internet.
IPv6: Internet Protocol version 6	Giao thức Internet phiên bản 6, địa chỉ của thiết bị trên môi trường mạng Internet
NFC: Near-Field Communications	Kết nối trường gần
POP: Post Office Protocol	Giao thức lấy thư điện tử từ mail server
RFID: Radio Frequency Identification	Nhận dạng qua tần số vô tuyến
SoC: System on Chip	Hệ thống trên một vi mạch
SMTP: Simple Mail Transfer Protocol	Giao thức truyền tải thư tín đơn giản
Wi-Fi: Wireless Fidelity	Hệ thống mạng không dây sử dụng sóng vô tuyến



## MỞ ĐẦU

Trong thời gian gần đây, tình trạng hỏa hoạn xảy ra liên tiếp, có quy mô thiệt hại đa dạng trên địa bàn TP HCM. Các vụ hỏa hoạn xảy ra gần đây nhất xảy ra thường xảy ra ở các chung cư trên địa bàn. Một số khác xảy ra ở các khu vực nhà dân, xí nghiệp. Điểm chung của các địa điểm xảy ra hỏa hoạn là hệ thống cảnh báo cháy đều không được đầu tư, chú trọng, hoặc hệ thống bị tê liệt vào lúc các dấu hiệu xuất hiện hỏa hoạn.

Qua đó, để giảm thiểu các thiệt hại không đáng có vì hỏa hoạn, em đã quyết định chọn thực hiện đề tài này để giúp giảm thiểu được phần nào hỏa hoạn gây ra, cùng với đó giúp người dùng có thể chủ động hơn khi có các tình huống không may xảy ra. Nội dung nghiên cứu của đề tài hiện tại của em chỉ ở mức giám sát dữ liệu cảm biến gửi về, thông báo tình trạng môi trường từ cảm biến và từ đó đưa ra các hành động cụ thể hoặc thông báo đến người dùng để họ có thể có các phản ứng kịp thời.

## CHƯƠNG 1. GIỚI THIỆU VỀ INTERNET OF THINGS (IOT)

### 1.1. GIỚI THIỆU VỀ INTERNET OF THINGS (IOT)

IoT là thuật ngữ dùng để chỉ các đối tượng có thể được nhận biết (identifiable) cũng như chỉ sự tồn tại của chúng trong một kiến trúc mạng máy tính kết nối với nhau. Cụm từ này được đưa ra bởi Kevin Ashton vào năm 1999. Ông là một nhà khoa học đã sáng lập ra Trung tâm Auto-ID ở đại học MIT, nơi thiết lập các quy chuẩn toàn cầu cho RFID (một phương thức giao tiếp không dây dùng sóng radio) cũng như một số loại cảm biến khác. IoT sau đó cũng được dùng nhiều trong các ấn phẩm đến từ các hãng và nhà phân tích.

Vào tháng 6 năm 2009, Ashton từng cho biết rằng: "Ngày nay, máy tính và Internet hầu như hoàn toàn phụ thuộc vào con người mới có thông tin. Gần như tất cả dữ liệu trên Internet lần đầu tiên được con người gửi và tạo ra bằng cách đánh máy, nhấn nút ghi âm, chụp ảnh hoặc quét mã vạch...". Con người chính là nhân tố quyết định trong thế giới Internet hiện nay. Vấn đề là, con người rất hạn chế về thời gian và chính xác, nghĩa là con người không được tốt lắm trong việc lưu giữ dữ liệu về mọi thứ trong thế giới.



Hình 1.1. Mô hình Internet of Things (IoT).

Ví dụ đơn giản như sau: chiếc tủ lạnh thông thường của người dùng không được kết nối với thiết bị nào khác. Nếu chúng ta muốn ghi lại nhiệt độ ở từng thời điểm của tủ, chúng ta chỉ có cách ghi lại thủ công rồi nhập vào một máy tính hay thiết bị lưu trữ nào đó. Hay như bóng đèn neon ở nhà chẳng hạn, chúng ta muốn thu thập, điều chỉnh độ sáng của nó thì phải đo thủ công rồi ghi lại.

Còn nếu như máy tính có khả năng giúp con người thu thập tất cả những dữ liệu về mọi thứ xung quanh, chúng ta có thể "theo dõi và đếm mọi thứ, giúp giảm hao phí, chi phí và lỗi. Chúng ta sẽ biết chính xác khi nào các vật dụng cần phải sửa chữa, thay thế, khi nào chúng còn mới và khi nào thì chúng hết hạn sử dụng. Chưa kể đến việc chúng ta có thể kiểm soát chúng mọi lúc mọi nơi. IoT có tiềm năng thay đổi thế giới, giống như cách mà Internet đã thay đổi cuộc sống của chúng ta. Ngôi nhà thông minh với các bóng đèn thông minh, máy giặt thông minh, tủ lạnh thông minh,... có thể xem là bước đầu của IoT bởi chúng đều được liên kết với nhau và/hoặc liên kết vào Internet.

## 1.2. KHẢ NĂNG ĐỊNH DANH

Điểm quan trọng của IoT đó là các đối tượng phải có thể được nhận biết và định dạng (identifiable). Nếu mọi đối tượng, kể cả con người, được "đánh dấu" để phân biệt bản thân đối tượng đó với những thứ xung quanh thì chúng ta có thể hoàn toàn quản lý được nó thông qua máy tính. Việc đánh dấu (tagging) có thể được thực hiện thông qua nhiều công nghệ, chẳng hạn như RFID, NFC, mã vạch, mã QR, watermark kỹ thuật số... Việc kết nối thì có thể thực hiện qua Wi-Fi, mạng viễn thông băng thông rộng (3G, 4G), Bluetooth, ZigBee, hồng ngoại...

Ngoài những kỹ thuật nói trên, nếu nhìn từ thế giới web, chúng ta có thể sử dụng các địa chỉ độc nhất để xác định từng vật, chẳng hạn như địa chỉ IP. Mỗi thiết bị sẽ có một IP riêng biệt không nhầm lẫn. Sự xuất hiện của IPv6 với không gian địa chỉ cực kỳ rộng lớn sẽ giúp mọi thứ có thể dễ dàng kết nối vào Internet cũng như kết nối với nhau.

## 1.3. TÍNH CHẤT CỦA IOT

### 1.3.1. Thông minh

Sự thông minh và tự động trong điều khiển thực chất không phải là một phần trong ý tưởng về IoT. Các máy móc có thể dễ dàng nhận biết và phản hồi lại môi trường xung quanh (ambient intelligence), chúng cũng có thể tự điều khiển bản thân (autonomous control) mà không cần đến kết nối mạng. Tuy nhiên, trong thời gian gần đây người ta bắt đầu nghiên cứu kết hợp hai khái niệm IoT và autonomous control lại với nhau. Tương lai của IoT có thể là một mạng lưới các thực thể thông minh có khả năng tự tổ chức và hoạt động riêng lẻ tùy theo tình huống, môi trường, đồng thời chúng cũng có thể liên lạc với nhau để trao đổi thông tin, dữ liệu.

Việc tích hợp trí thông minh vào IoT còn có thể giúp các thiết bị, máy móc, phần mềm thu thập và phân tích các dấu vết điện tử của con người khi chúng ta tương tác với những thứ thông minh, từ đó phát hiện ra các tri thức mới liên quan tới cuộc sống, môi trường, các mối tương tác xã hội cũng như hành vi con người.

### 1.3.2. Kiến trúc dựa trên sự kiện

Các thực thể, máy móc trong IoT sẽ phản hồi dựa theo các sự kiện diễn ra trong lúc chúng hoạt động theo thời gian thực. Một số nhà nghiên cứu từng nói rằng một mạng lưới các cảm biến chính là một thành phần đơn giản của IoT.

### 1.3.3. Là một hệ thống phức tạp

Trong một thế giới mở, IoT sẽ mang tính chất phức tạp bởi nó bao gồm một lượng lớn các đường liên kết giữa những thiết bị, máy móc, dịch vụ với nhau, ngoài ra còn bởi khả năng thêm vào các nhân tố mới.

### 1.3.4. Kích thước

Một mạng lưới IoT có thể chứa đến 50 đến 100 nghìn tỉ đối tượng được kết nối và mạng lưới này có thể theo dõi sự di chuyển của từng đối tượng. Một con người sống trong thành thị có thể bị bao bọc xung quanh bởi 1000 đến 5000 đối tượng có khả năng theo dõi.

### 1.3.5. Vấn đề không gian, thời gian

Trong IoT, vị trí địa lý chính xác của một vật nào đó là rất quan trọng. Hiện nay, Internet chủ yếu được sử dụng để quản lý thông tin được xử lý bởi con người. Do đó những thông tin như địa điểm, thời gian, không gian của đối tượng không mấy quan trọng bởi người xử lý thông tin có thể quyết định các thông tin này có cần thiết hay không, và nếu cần thì họ có thể bổ sung thêm. Trong khi đó, IoT về lý thuyết sẽ thu thập rất nhiều dữ liệu, trong đó có thể có dữ liệu thừa về địa điểm, và việc xử lý dữ liệu đó được xem như không hiệu quả. Ngoài ra, việc xử lý một khối lượng lớn dữ liệu trong thời gian ngắn đủ để đáp ứng cho hoạt động của các đối tượng cũng là một thách thức hiện nay.

### 1.4. ỨNG DỤNG CỦA IOT

IoT có ứng dụng rộng vô cùng, có thể kể ra một số thứ như sau:

- Quản lý chất thải
- Quản lý và lập kế hoạch quản lý đô thị
- Quản lý môi trường
- Phản hồi trong các tình huống khẩn cấp
- Mua sắm thông minh
- Quản lý các thiết bị cá nhân
- Đồng hồ đo thông minh
- Tự động hóa ngôi nhà, thành thị.

Một trong những vấn đề với IoT đó là thời gian để tạo ra một ứng dụng IoT nhanh chóng. Để khắc phục, hiện nay nhiều hãng, công ty, tổ chức trên thế giới đang nghiên cứu các nền tảng giúp xây dựng nhanh ứng dụng dành cho IoT. Đại học British Columbia ở Canada hiện đang tập trung vào một bộ toolkit cho phép phát triển phần mềm IoT chỉ bằng các công nghệ/tiêu chuẩn Web cũng như giao thức phổ biến. Công ty như ioBridge thì cung cấp giải pháp kết nối và điều khiển hầu như bất kì thiết bị nào có khả năng kết nối Internet, kể cả đèn bàn, quạt máy...



Hình 2.2. Nền tảng IoT do nhóm sinh viên Việt Nam sản xuất: iNut.

Việt Nam cũng đã có một số nền tảng, bộ toolkit để cho người dùng có thể sử dụng, rút ngắn thời gian lập trình, tạo ra sản phẩm nhanh nhất có thể. Có thể kể đến như nền tảng iNut của nhóm bạn sinh viên trường Đại học Khoa học Tự nhiên HCM – Ngô Huỳnh Ngọc Khánh.

## 1.5. NHỮNG TÁC NHÂN ẢNH HƯỞNG ĐẾN SỰ PHÁT TRIỂN CỦA IOT

### 1.5.1. Chưa có một ngôn ngữ chung

Ở mức cơ bản nhất, Internet là một mạng dùng để nối thiết bị này với thiết bị khác. Nếu chỉ riêng có kết nối không thôi thì không có gì đảm bảo rằng các thiết bị biết cách nói chuyện nói nhau. Ví dụ, người dùng có thể đi từ Việt Nam đến Mỹ, nhưng không đảm bảo rằng người dùng có thể nói chuyện tới với người Mỹ.

Để các thiết bị có thể giao tiếp với nhau, chúng sẽ cần một hoặc nhiều giao thức (protocols), có thể xem là một thứ ngôn ngữ chuyên biệt để giải quyết một tác vụ nào đó. Chắc chắn người dùng đã ít nhiều sử dụng một trong những giao thức phổ biến nhất thế giới, đó là HyperText Transfer Protocol (HTTP) để tải web. Ngoài ra chúng ta còn có SMTP, POP, IMAP dành cho email, FTP dùng để trao đổi file, và một số giao thức khác.

Những giao thức như thế này hoạt động ổn định bởi các máy chủ web, mail và FTP thường không phải nói với nhau nhiều, khi cần, một phần mềm phiên dịch đơn giản sẽ đứng ra làm trung gian để hai bên giao tiếp. Còn với các thiết bị IoT, chúng phải đảm đương rất nhiều thứ, phải nói chuyện với nhiều loại máy móc thiết bị khác nhau. Đáng tiếc rằng hiện người ta chưa có nhiều sự đồng thuận về các giao thức để IoT trao đổi dữ liệu. Hiện tại các thiết bị này vẫn phải mượn các giao thức để có thể giao tiếp với nhau.

### 1.5.2. Hàng rào subnetwork

Như đã nói ở trên, thay vì giao tiếp trực tiếp với nhau, các thiết bị IoT hiện nay chủ yếu kết nối đến một máy chủ trung tâm do hãng sản xuất một nhà phát triển nào đó quản lý. Cách này cũng vẫn ổn thôi, những thiết bị vẫn hoàn toàn nói được với nhau thông qua chức năng phiên dịch của máy chủ rồi. Thế nhưng mọi chuyện không đơn giản như thế, cứ mỗi một mạng lưới như thế tạo thành một subnetwork riêng, và buồn thay các máy móc nằm trong subnetwork này không thể giao tiếp tốt với subnetwork khác.

Lấy ví dụ như xe ô tô chẳng hạn. Một chiếc Ford Focus có thể giao tiếp cực kì tốt đến các dịch vụ và trung tâm dữ liệu của Ford khi gửi dữ liệu lên mạng. Nếu một bộ phận nào đó cần thay thế, hệ thống trên xe sẽ thông báo về Ford, từ đó hãng tiếp tục thông báo đến người dùng. Nhưng trong trường hợp chúng ta muốn tạo ra một hệ thống cảnh báo kẹt xe thì mọi chuyện rắc rối hơn nhiều bởi xe Ford được thiết lập chỉ để nói chuyện với server của Ford, không phải với server của Honda, Audi, Mercedes hay BMW. Lý do cho việc giao tiếp thất bại? Chúng ta thiếu đi một ngôn ngữ chung. Và để thiết lập cho các hệ thống này nói chuyện được với nhau thì rất tốn kém, đắt tiền.

Một số trong những vấn đề nói trên chỉ đơn giản là vấn đề về kiến trúc mạng, về kết nối mà các thiết bị sẽ liên lạc với nhau (Wifi, Bluetooth, NFC,...). Những thứ này thì tương đối dễ khắc phục với công nghệ không dây ngày nay. Còn với các vấn đề về giao thức thì phức tạp hơn rất nhiều, nó chính là vật vãn lớn và trực tiếp trên còn đường phát triển của IoT.

### 1.5.3. Có quá nhiều "ngôn ngữ địa phương"

Bây giờ giả sử như các nhà sản xuất xe ô tô nhận thấy rằng họ cần một giao thức chung để xe của nhiều hãng có thể trao đổi dữ liệu cho nhau và họ đã phát triển thành công giao thức đó. Thế nhưng vấn đề vẫn chưa được giải quyết. Nếu các trạm thu phí đường bộ, các trạm bơm xăng muốn giao tiếp với xe thì sao? Mỗi một loại thiết bị lại sử dụng một "ngôn ngữ địa phương" riêng thì mục đích của IoT vẫn chưa đạt được đến mức tối đa. Đồng ý rằng chúng ta vẫn có thể có một trạm kiểm soát trung tâm, thế nhưng các thiết bị vẫn chưa thật sự nói được với nhau.

### 1.5.4. Tiền và chi phí

Cách duy nhất để các thiết bị IoT có thể thật sự giao tiếp đó là khi có một động lực kinh tế để mạnh khiến các nhà sản xuất đồng ý chia sẻ quyền điều khiển cũng như dữ liệu mà các thiết bị của họ thu thập được. Hiện tại, các động lực này không nhiều. Có thể xét đến ví dụ sau: một công ty thu gom rác muốn kiểm tra xem các thùng rác có đầy hay chưa. Khi đó, họ phải gặp nhà sản xuất thùng rác, đảm bảo rằng họ có thể truy cập vào hệ thống quản lý của từng thùng một. Điều đó khiến chi phí bị đội lên, và công ty thu gom rác có thể đơn giản chọn giải pháp cho một người chạy xe kiểm tra từng thùng một.

## 1.6. Các "hầm chứa" tập trung hay những "hòn đảo Internet"?

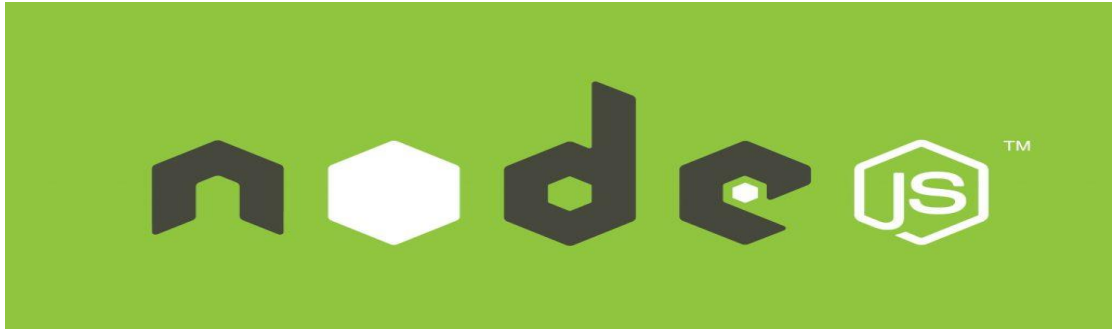
Nếu xu hướng hiện nay tiếp tục, dữ liệu được các thiết bị gửi và nhận sẽ nằm trong các "hầm chứa" mang tính chất tập trung (centralized silo). Các công ty, nhà sản xuất có thể kết nối đến các hầm này để thu thập dữ liệu, từ đó tạo ra các bộ giao thức của riêng mình. Tuy nhiên, nhược điểm của mô hình này đó là dữ liệu sẽ trở nên khó chia sẻ hơn bởi người ta cứ phải tạo ra các đường giao tiếp mới giữa các hầm chứa. Dữ liệu sẽ phải di chuyển xa hơn và làm chậm tốc độ kết nối. Chưa kể đến các nguy cơ bảo mật và nguy cơ tiềm ẩn về quyền riêng tư của người dùng nữa.

Trái ngược với hướng đi trên, nếu như các nhà sản xuất có thể thống nhất được các bộ giao tiếp chung thì sẽ tạo ra các "hòn đảo Internet" (Internet of Islands). Thiết bị trong một căn phòng có thể giao tiếp với nhau, giao tiếp với các máy móc khác trong nhà và thậm chí là cả... nhà hàng xóm. Dữ liệu sẽ được phân bố trong một khu vực hẹp hơn nên đảm bảo các vấn đề bảo mật, đồng thời tăng tốc độ hoạt động. Dữ liệu cũng nhờ đó mà linh hoạt hơn, các thiết bị có thể phản hồi nhanh hơn. Bên cạnh đó, một khi các thiết bị có thể nói chuyện tốt với nhau, một hệ thống tự động hóa có thể bắt đầu học hỏi những gì đang diễn ra ở thế giới xung quanh, từ đó đưa ra hành động đúng ý muốn của người dùng.

## CHƯƠNG 2. TỔNG QUAN VỀ NODEJS

### 2.1. TỔNG QUAN VỀ NODEJS

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các trang diễn đàn và đặc biệt là trang mạng xã hội có phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.



Hình 3.1. NodeJS

Khi nói đến NodeJS thì phải nghĩ tới vấn đề Realtime. Realtime ở đây chính là xử lý giao tiếp từ client tới máy chủ theo thời gian thực. Giống như khi người dùng truy cập Facebook thì mỗi khi bình luận hay nhất nút “Like” một bài viết nào đó thì ngay lập tức người đăng bài viết và những người đã bình luận trước đó sẽ nhận được thông báo là đã có một ai đó bình luận. Đây có thể nói là điểm đáng chú ý nhất của NodeJS khi so sánh với các ngôn ngữ khác.

### 2.2. ƯU ĐIỂM CỦA NODEJS

#### 2.2.1. JSON APIs

Bởi lẽ REST/JSON APIs gọn nhẹ là điều khiến NodeJS tỏa sáng. Với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Web làm bằng JSON. NodeJS tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ cao như PHP.

#### 2.2.2. Ứng dụng trên 1 trang

NodeJS rất phù hợp lập trình nên các ứng dụng web với định dạng 1 trang tương tự như Gmail. Với khả năng xử lý nhiều Request đồng thời và thời gian phản hồi nhanh. Các ứng dụng người dùng định viết không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn của người dùng.

#### 2.2.3. Shelling tools unix

NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu suất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.

#### **2.2.4. Streaming Data (Luồng dữ liệu)**

Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả sử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.

#### **2.2.5. Ứng dụng Web thời gian thực**

Giả sử người dùng xây dựng 1 ứng dụng chat, feed ... Facebook, Twitter là điển hình cho Web thời gian thực. NodeJS làm khá tốt điều đó.

### **2.3. NHƯỢC ĐIỂM CỦA NODEJS**

#### **2.3.1. Ứng dụng nặng, tốn tài nguyên**

Nếu người dùng cần xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption... hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS (Lý do: NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút). Trường hợp này người dùng hãy viết 1 Addon C++ để tích hợp với NodeJS để tăng hiệu suất tối đa.

#### **2.3.2. Tính năng hỗ trợ**

NodeJS, PHP, Ruby, Python, .NET đều là các ngôn ngữ phục vụ cho việc lập trình các website. Tùy theo góc nhìn và các tính năng mà có thể thấy NodeJS tốt hơn hay các ngôn ngữ khác tốt hơn. Điểm mạnh của NodeJS là việc hỗ trợ tính năng thời gian thực, thời gian phản hồi các request của phía client nhanh hơn so với hầu hết các ngôn ngữ khác. Nhưng vì đây là một ngôn ngữ có tuổi đời kém hơn so với các ngôn ngữ hiện có, nên việc hỗ trợ còn chưa được đầy đủ bằng.

### **2.4. ỨNG DỤNG CỦA NODEJS TRONG ĐỀ TÀI THỰC TẬP**

Trong đề tài thực tập lần này, NodeJS được ứng dụng để sử dụng hiển thị thông tin mà cảm biến truyền về, từ đó người quản trị hệ thống có thể biết được các thông số môi trường tại nơi đặt cảm biến, có các phản ứng kịp thời nếu tình huống xấu xảy ra. Từ đó giảm thiểu các rủi ro về tài sản và con người cho người sử dụng, đặc biệt là tính mạng con người.



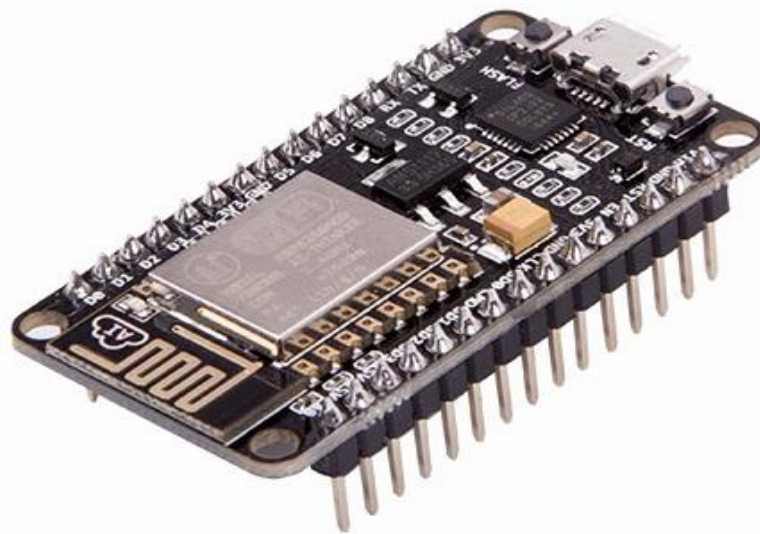
## CHƯƠNG 3. HỆ THỐNG GIÁM SÁT, CẢNH BÁO CHÁY

### 3.1. CÁC THÀNH PHẦN SỬ DỤNG TRONG HỆ THỐNG

#### 3.1.1. Phần cứng

##### a) ESP8266 (NodeMCU)

ESP8266 là một mạch vi điều khiển có thể giúp chúng ta điều khiển các thiết bị điện tử. Điều đặc biệt của nó, đó là sự kết hợp của module Wifi tích hợp sẵn bên trong con vi điều khiển chính. Hiện nay, ESP8266 rất được giới nghiên cứu tự động hóa Việt Nam ưa chuộng vì giá thành cực kỳ rẻ (chỉ bằng một con Arduino Nano), nhưng lại được tích hợp sẵn Wifi, bộ nhớ flash 8Mb.



Hình 4.1. ESP8266 (NodeMCU)

Thông số kỹ thuật:

- IC chính: ESP8266 Wifi SoC.
- Phiên bản firmware: NodeMCU Lua
- Chip nạp và giao tiếp UART: CP2102.
- 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: 25 x 50 mm

##### b) Cảm biến khí MQ-2

Cảm biến khí MQ-2 sử dụng để phát hiện khí gas trong môi trường. Cảm biến có độ nhạy cao khả năng phản hồi nhanh, độ nhạy có thể điều chỉnh được bằng biến trở, có thể phát hiện khí gas, metan, butan, LPG, khói.



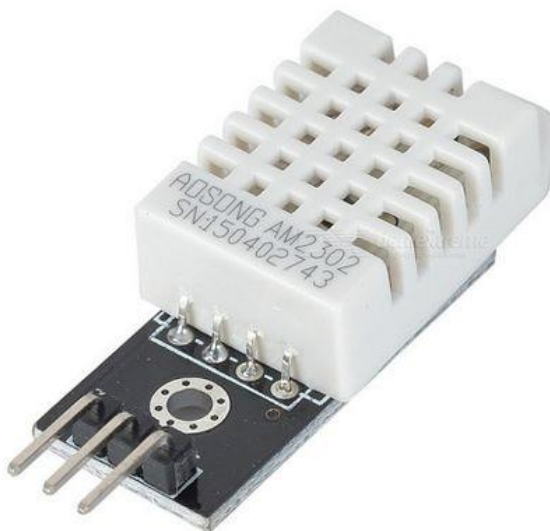
Hình 5.2. Cảm biến khí MQ-2

Thông Số Kỹ thuật:

- Nguồn hoạt động: 5V
- Loại dữ liệu: Analog
- Phạm vi phát hiện rộng
- Tốc độ phản hồi nhanh và độ nhạy cao
- Mạch đơn giản
- Ổn định khi sử dụng trong thời gian dài

**c) Cảm biến nhiệt độ, độ ẩm DHT22**

Đây là cảm biến thông dụng tích hợp vừa đo được nhiệt độ và độ ẩm, độ chính xác khá cao. Giao tiếp với vi điều khiển qua chuẩn giao tiếp 1 dây. DHT22 có độ chính xác cao và khoảng đo hoạt động rộng. Module truyền dữ liệu thông qua giao tiếp 1 dây nên dễ dàng kết nối và lấy dữ liệu. Module được thiết kế hoạt động ở mức điện áp 5V.

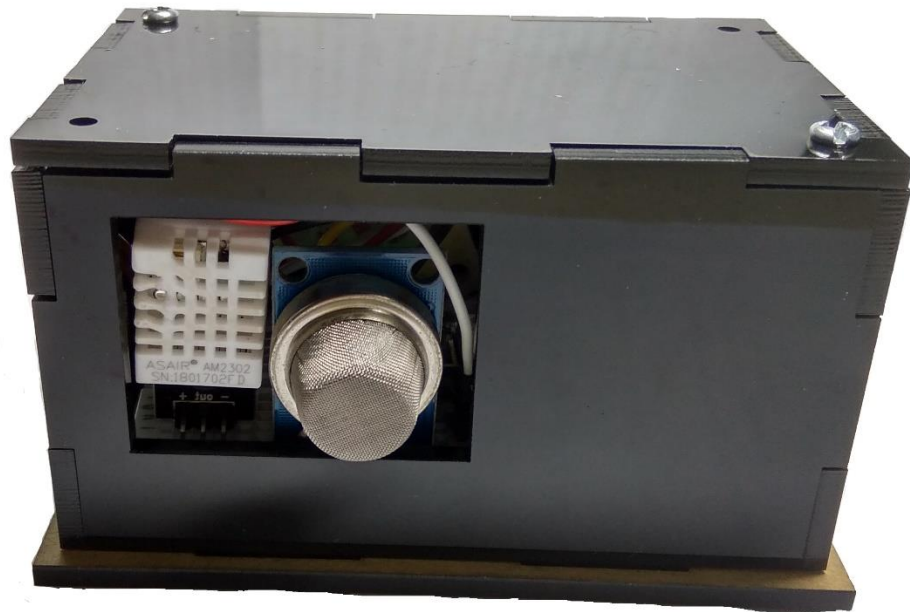


Hình 6.3. Cảm biến nhiệt độ, độ ẩm DHT22

Thông số kỹ thuật:

- Điện áp hoạt động: 5V.
- Khoảng đo độ ẩm: 0% - 100% RH sai số 2% RH.
- Khoảng đo nhiệt độ: -40 ~ -80 độ C sai số 0.5% độ C.
- Tần số lấy mẫu tối đa 0.5Hz (2 giây / lần).
- Kích thước: 28mm x 12mm x 10mm.

Hình ảnh về thiết bị phân cứng:



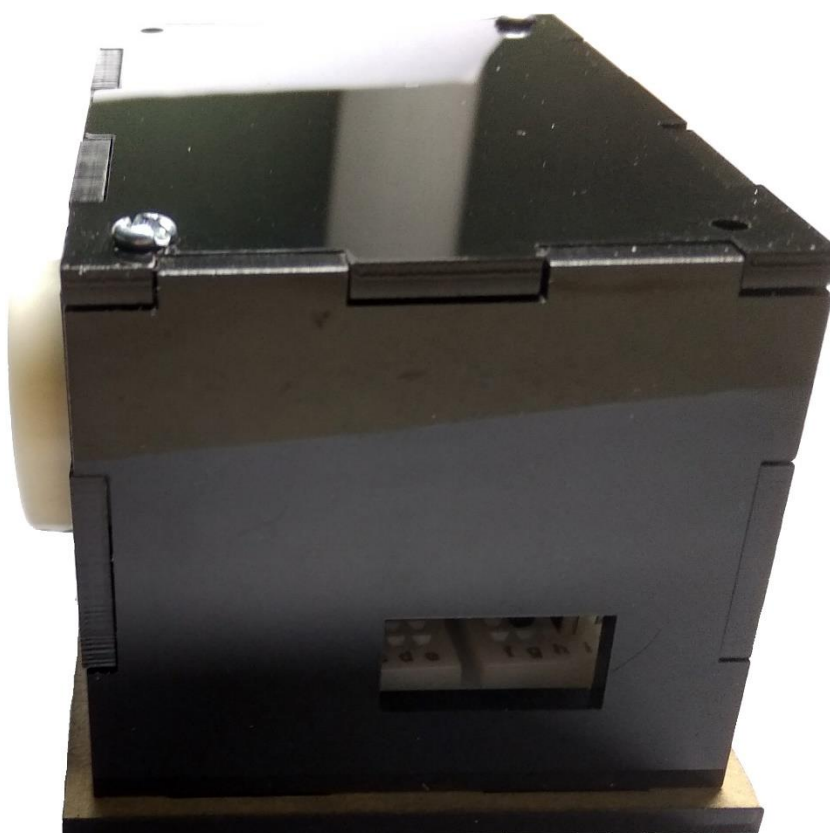
Hình 7.4. Mặt trước của thiết bị.



Hình 8.5. Mặt sau của thiết bị.



*Hình 9.6. Cạnh phải của thiết bị.*



*Hình 10.7. Cạnh trái của thiết bị*

### 3.1.2. Phần mềm

#### a) Website hiển thị thông số từ cảm biến (NodeJS)

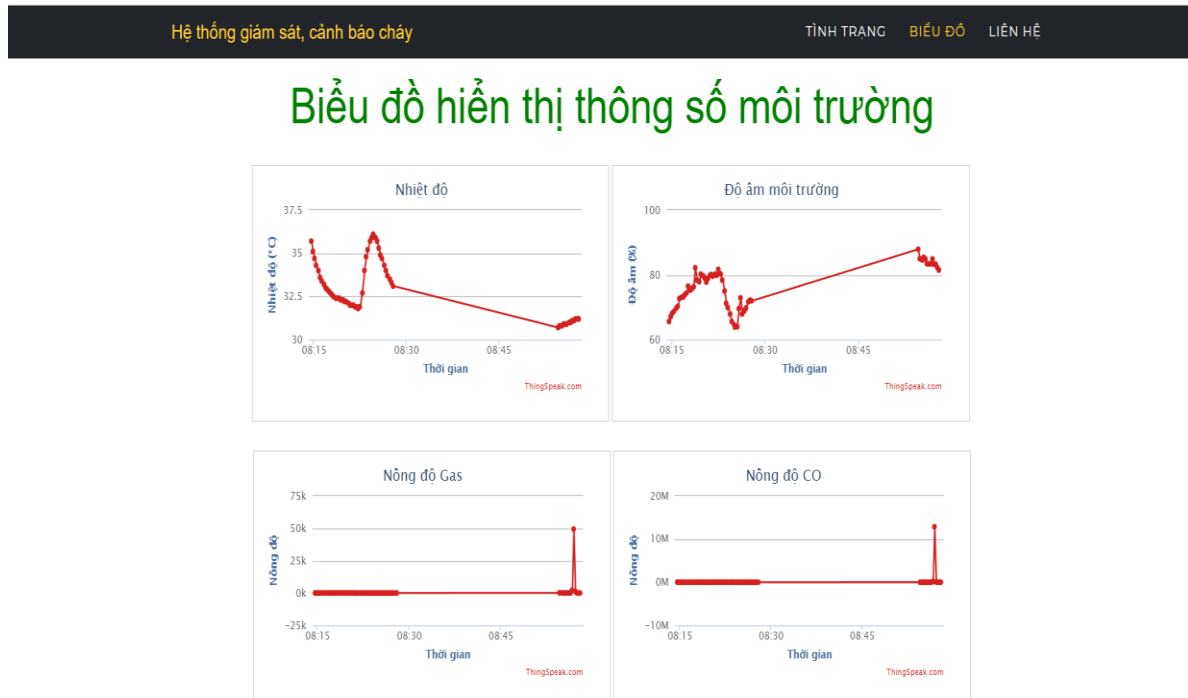
Để hiển thị thông số môi trường từ cảm biến, ngoài việc hiển thị thông số trên thiết bị Android thì còn có xem thông số trực tiếp từ website.



Hình 11.8. Website hiển thị thông số cảm biến

Dựa vào các thông số môi trường mà phần tình trạng sẽ được hiển thị. Ví dụ, đối với nhiệt độ môi trường dưới 27 độ C (không tính các thông số môi trường khác) thì tình trạng môi trường sẽ hiển thị là “Bình thường”. Hoặc nếu nhiệt độ môi trường vượt quá ngưỡng cho phép (nhiệt độ vượt ngưỡng 50 °C) thì tình trạng sẽ là “Cháy”.

Ngoài các thông số môi trường, trang web còn được tích hợp các biểu đồ thể hiện các thông số theo thời gian, nhờ vào công cụ ThingSpeak. Từ đó, người quản trị hệ thống có thể quan sát một cách trực quan sự thay đổi của các yếu tố môi trường.



Hình 12.9. Biểu đồ hiển thị thông số môi trường.



**b) Ứng dụng trên điện thoại Android để hiển thị thông số từ cảm biến**



Hình 13.10. Ứng dụng “Cảnh báo cháy” trên thiết bị Android.

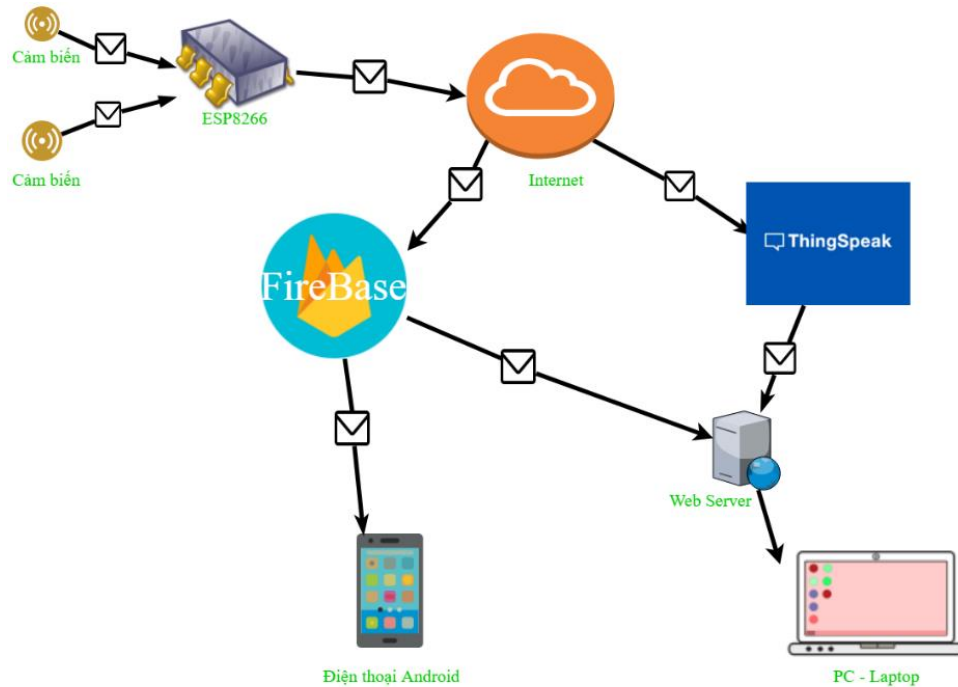
Tương tự như website, ứng dụng “Cảnh báo cháy” trên thiết bị Android cũng có công dụng hiển thị thông số môi trường tại nơi đặt cảm biến và tình trạng môi trường sẽ hiển thị dựa theo thông số môi trường nhận được với giải thuật xác định tình trạng tương tự như trên trang web.

**3.2. NGUYÊN LÝ HOẠT ĐỘNG CỦA HỆ THỐNG**



Hình 14.11. Lưu đồ giải thuật hiển thị tình trạng môi trường của hệ thống.

Các thông số môi trường sẽ được thu thập từ các cảm biến sau đó được tập hợp tại ESP8266. Các thông số này bao gồm: nhiệt độ, độ ẩm, nồng độ khí gas, nồng độ khói, nồng độ khí CO / CO<sub>2</sub>. Sau đó dữ liệu này sẽ được truyền đến cơ sở dữ liệu Firebase và ThingSpeak thông qua môi trường Internet.



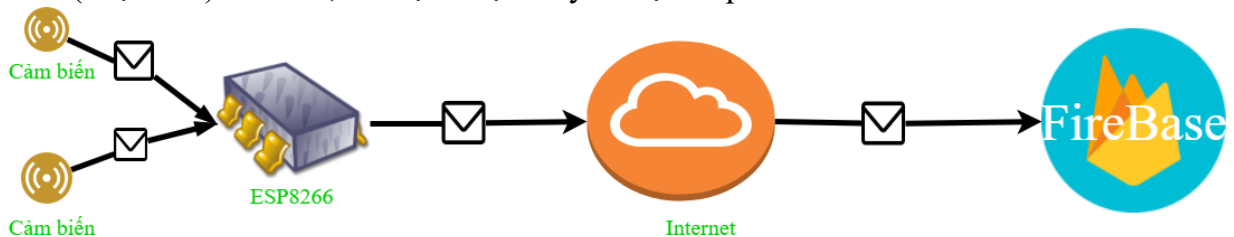
Hình 15.12. Sơ đồ nguyên lý hoạt động của hệ thống cảnh báo cháy.

Hai hệ thống máy chủ này sẽ sử dụng các khóa cần thiết được gửi trong dữ liệu để xác định cơ sở dữ liệu của người dùng để thêm dữ liệu. Quá trình thêm dữ liệu sẽ được diễn ra nhanh chóng, và người dùng có thể quan sát được quá trình thêm dữ liệu trên cơ sở dữ liệu của mình trên Firebase và ThingSpeak thông qua giao diện người dùng một cách trực quan.

Dữ liệu được lưu trữ trên Firebase sẽ được gửi đến Website và ứng dụng “Cảnh báo cháy” để hiển thị trên trang web “canhbaochay.herokuapp.com” và thiết bị Android đã cài đặt ứng dụng “Cảnh báo cháy”. Website ngoài việc nhận dữ liệu từ Firebase còn nhận dữ liệu theo dạng biểu đồ từ ThingSpeak để hiển thị lên trang web. Từ đó người dùng có thể quan sát được tình trạng thay đổi của môi trường trong thời gian nhất định.

### 3.2.1. Nguyên lý hoạt động của ESP8266

Hai cảm biến nhiệt, độ ẩm (DHT22) và cảm biến khí (MQ-2) có nhiệm vụ thu thập các thông số môi trường, bao gồm nhiệt độ, độ ẩm, nồng độ khói, nồng độ gas, nồng độ CO<sub>2</sub> (hoặc CO). Sau đó, dữ liệu được truyền trực tiếp đến ESP8266.



Hình 16.13. Sơ đồ nguyên lý hoạt động của ESP8266 (1).

ESP8266 khi hoạt động sẽ đọc các dữ liệu liên tục được gửi từ các cảm biến. Sau đó, nhờ vào kết nối Internet, ESP8266 sẽ gửi dữ liệu mà các cảm biến thu thập được đến Firebase dựa theo biến đường dẫn “Firebase\_Host” và mã xác thực “FIREBASE\_AUTH”. Hai biến này là yếu tố cần thiết để có thể truyền dữ liệu đến Firebase, từ đó các ứng dụng liên quan sẽ nhận được dữ liệu để có thể hiển thị. Dữ liệu được gửi từ ESP8266 đến Firebase sẽ bao gồm hai biến nêu trên và các thông số thu thập được từ cảm biến.

```
#define FIREBASE_HOST "canhbaochay.firebaseio.com"
#define FIREBASE_AUTH "g505zsy04C10h0X0j9QZhXIw400zRp70ojsYr33G"
```

Hình 17.14. Các biến cần thiết để kết nối ESP8266 và Firebase.

Sau đó, ESP8266 sẽ tạo lập kết nối đến máy chủ Firebase, gửi dữ liệu với các biến là các node bên trong CSDL của Firebase, người lập trình chỉ cần gọi hàm và đúng với tên nút lưu trữ giá trị bên trong CSDL là có thể ghi dữ liệu lên CSDL của Firebase.

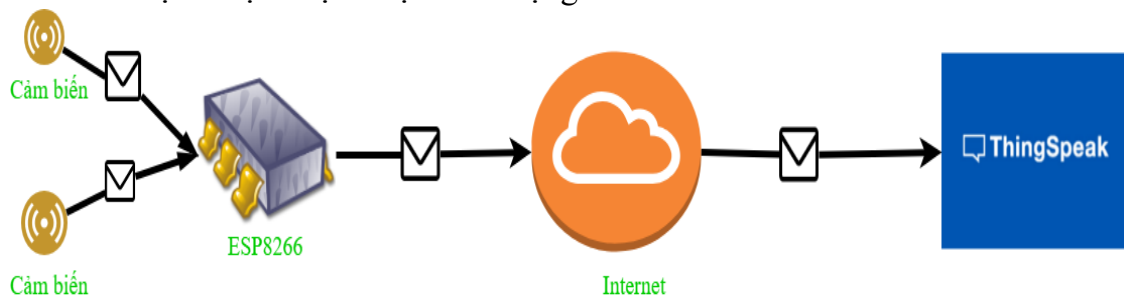
```
//Thiết lập kết nối với máy chủ Firebase
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

//Sau đó, gửi dữ liệu về Firebase
Firebase.setFloat("nhietdo", nhietdo);
Firebase.setFloat("doam", doam);
Firebase.setFloat("gas", gas);
Firebase.setFloat("khoi", khoi);
Firebase.setFloat("CO", co);

//Kiểm tra lỗi
if (Firebase.failed()) {
  Serial.print("setting /number failed:");
  Serial.println(Firebase.error());
  return;
}
```

Hình 18.15. Các thức kết nối và truyền dữ liệu đến Firebase

Ngoài việc gửi dữ liệu đến Firebase, ESP8266 sẽ gửi dữ liệu thu thập được đến ThingSpeak. Đây là một công cụ khá hữu ích đối với IoT. ThingSpeak hỗ trợ khá nhiều công cụ đi kèm giúp người dùng có thể phân tích, đánh giá hệ thống, các công cụ báo cáo, phản ứng đi kèm. Và trong đồ án này, ThingSpeak đóng vai trò là công cụ nhận dữ liệu và hiển thị dữ liệu nhận được dưới dạng biểu đồ.



Hình 19.16. Sơ đồ nguyên lý hoạt động của ESP8266 (2).



Để ThingSpeak có thể nhận được dữ liệu từ ESP8266 thông qua môi trường Internet, trong mã nguồn được nạp vào ESP8266 phải khai báo các biến như “channelID”, “writeAPIKey”, “server”. Trong đó, biến “channelID” giúp ESP8266 có thể gửi đến chính xác channel được sử dụng trên ThingSpeak.

```
const int channelID = 478199;
String writeAPIKey = "EB73VMZLI139YU5S";
const char* server = "api.thingspeak.com";
```

Hình 20.17. Các biến cần thiết để ESP8266 gửi dữ liệu về ThingSpeak.

Tương tự như khi gửi dữ liệu đến FireBase, các gói tin được gửi về ThingSpeak sẽ bao gồm các thông số môi trường thu thập được từ các cảm biến và các biến cần thiết của ThingSpeak để dữ liệu được gửi đến đích chính xác nhất.

Để thiết lập kết nối đến máy chủ của ThingSpeak, ESP8266 sẽ sử dụng một loạt các hàm kết nối đến máy chủ ThingSpeak, trong đó gọi các biến cần thiết nêu trên, giao thức gửi nhận dữ liệu theo kiểu GET hoặc POST, độ dài của dữ liệu sẽ gửi đi và một số thành phần khác.

```
if (client.connect(server, 80)) {
    String body = "field1=" + String(nhietdo, 1) + "&field2=" + String(doam, 1) + "&field3=" + String(gas, 1) +
    "&field4=" + String(co, 1);
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + writeAPIKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(body.length());
    client.print("\n\n");
    client.print(body);
    client.print("\n\n");
}
client.stop();
```

Hình 21.18. Cách để ESP8266 kết nối và gửi dữ liệu đến ThingSpeak..

### 3.2.2. Nguyên lý hoạt động của cảm biến MQ – 2

Sau khi được cấp nguồn điện, cảm biến khí MQ – 2 sẽ liên tục đọc các giá trị không khí trong môi trường. Các thông số sẽ được truyền từ các chân DOUT và AOUT của cảm biến về vi điều khiển. Ví dụ:

- Đối với trường hợp khí gas, tín hiệu DOUT:
  - Nếu có khí gas: tín hiệu ở mức thấp.
  - Nếu không có khí gas: tín hiệu ở mức cao.
- Đối với trường hợp khói:
  - Nếu có khói: điện áp ra cao.

- Nếu không có khói: điện áp ra thấp.

Đầu ra tín hiệu có thể là tín hiệu Analog được đọc bằng chân Analog của ESP8266 hoặc cũng có thể tín hiệu Digital được đọc bằng chân Digital của ESP8266.

### 3.2.3. Nguyên lý hoạt động của cảm biến DHT22

Ngưỡng điện áp hoạt động của cảm biến từ 3.3V đến 5V (một số lên tới 6V), nhiệt độ từ -40°C tới +80°C (một số datasheet là 125 °C) với độ chính xác  $\pm 0.5$  °C cho nhiệt độ và  $\pm 2\%$  cho độ ẩm. Chu kỳ lấy dữ liệu trung bình là 2s một lần.

### 3.2.4. Nguyên lý hoạt động của Realtime Database của Firebase

Firebase đóng vai trò là cơ sở dữ liệu để các ứng dụng liên kết và truy vấn dữ liệu đến. Khi dữ liệu bên trong Firebase thay đổi thì các ứng dụng liên kết đến nó cũng sẽ được đồng bộ, cập nhật dữ liệu theo. Cơ sở dữ liệu thời gian thực Firebase lưu trữ dữ liệu database dưới dạng JSON và thực hiện đồng bộ database tới tất cả các client theo thời gian thực.

Cụ thể hơn là người dùng có thể xây dựng được client đa nền tảng (cross-platform client) và tất cả các client này sẽ cùng sử dụng chung 1 database đến từ Firebase và có thể tự động cập nhật mỗi khi dữ liệu trong database được thêm mới hoặc sửa đổi.

Ví dụ, khi vi điều khiển ESP8266 gửi các thông số môi trường về cho Firebase, các thông số này sẽ được ghi lại trên cơ sở dữ liệu theo thời gian thực. Sau đó, Firebase sẽ cập nhật các thông số mới đến các ứng dụng liên kết.

### 3.2.5. Nguyên lý hoạt động của ThingSpeak

Sau khi kết nối giữa ESP8266 và ThingSpeak được thiết lập, phía ESP8266 khi gửi dữ liệu về ThingSpeak, sẽ gửi chỉ số kênh của người dùng để gửi dữ liệu, độ dài chuỗi sẽ gửi và các thông số của môi trường. Dựa vào chỉ số kênh, field của dữ liệu nhận được mà ThingSpeak sẽ ghi lại vào trong dữ liệu của mình, sau đó hiển thị các thông số này qua giao diện biểu đồ qua mỗi phiên kết nối.

### 3.2.6. Nguyên lý hoạt động của Website

Trang web dùng để hiển thị các thông số môi trường do ESP8266 thu thập được. Dữ liệu dùng để hiển thị sẽ được lấy từ 2 nguồn: Firebase và ThingSpeak. Từ các dữ liệu thông số môi trường nhận được, trang web sẽ xuất ra tình trạng môi trường hiện tại lên trên giao diện để người dùng có thể biết được tình trạng môi trường hiện tại, từ đó có các phản ứng cần thiết.

Trong đó, các thông số hiển thị trên trang web sẽ được lấy từ cơ sở dữ liệu của Firebase thông qua API có sẵn của Firebase. Sau đó sử dụng JQuery, dựa vào thông số môi trường để hiển thị tình trạng môi trường.

Các biểu đồ hiển thị của các thông số môi trường sẽ được lấy từ ThingSpeak dựa theo các thẻ iframe liên kết đến ThingSpeak để hiển thị các biểu đồ thông số môi trường theo thời gian mà ThingSpeak nhận được.

### 3.2.7. Nguyên lý hoạt động của ứng dụng “Cảnh báo cháy”

Dựa vào các API của Firebase, ứng dụng “Cảnh báo cháy” sẽ hiển thị các thông số môi trường từ cảm biến đã gửi đến Firebase. Khi có bất cứ thay đổi nào về các thông số môi trường thì ứng dụng cũng sẽ được cập nhật theo. Dựa vào các thông số môi trường nhận được, ứng dụng sẽ hiển thị tình trạng môi trường cho người dùng biết. Nếu tình trạng môi trường xấu đi, ứng dụng sẽ phát ra âm thanh cảnh báo đến người dùng.

### 3.3. CÁC YÊU CẦU CẦN THIẾT CỦA HỆ THỐNG

- Điểm mấu chốt của các hệ thống IoT hiện nay đó là kết nối Internet. Các hệ thống IoT sẽ không gọi là IoT nếu như không có kết nối Internet. Đây là điểm đầu tiên cần phải được chú ý đến.
- Hệ thống máy chủ web cần phải được cài đặt NodeJS để có thể chạy được ứng dụng web NodeJS.
- Thiết bị Android cần được kết nối Internet khi sử dụng giám sát môi trường thông qua Internet.
- Nguồn điện cấp cho thiết bị cảm biến cũng là một yêu cầu cần thiết để hệ thống hoạt động được ổn định.

### 3.4. ĐÁNH GIÁ HỆ THỐNG

#### 3.4.1. Ưu điểm

- Giao diện trực quan, dễ quan sát.
- Chi phí triển khai thấp.
- Thời gian phản hồi tương đối cao.
- Kết hợp nhiều nền tảng.
- Dễ sử dụng.

#### 3.4.2. Khuyết điểm

- Tồn tại độ trễ trong quá trình hiển thị các thông số, nhất là biểu đồ của ThingSpeak.
- Thiếu tính bảo mật.
- Chưa tận dụng hết nền tảng NodeJS (Socket.io).
- Tính thẩm mỹ chưa cao.

## KẾT LUẬN

Thông qua quá trình tìm hiểu và triển khai dưới sự hỗ trợ các anh đồng nghiệp trong công ty, cũng như những sự giúp đỡ, động viên từ bạn bè, người thân, em đã gần như hoàn thiện được đề tài thực tập “Hệ thống giám sát, cảnh báo cháy” của mình trong thời gian vừa qua. Qua đó, tiếp nhận được một lượng kiến thức mới về các công nghệ mới và sẽ là xu thế công nghệ trong tương lai mà trước đó vẫn chưa được tiếp cận được, đơn cử như NodeJS, IoT.

Hiểu được tầm quan trọng và mức độ nguy hiểm mà các vụ hỏa hoạn có thể gây ra đối với người và của. Được tìm hiểu và thu nhận được một lượng kiến thức bổ ích về các công nghệ, nền tảng mới sẽ phát triển trong tương lai. Trong đó NodeJS với tính năng real-time của mình sẽ hỗ trợ tốt trong việc phản hồi các truy vấn, giám sát của người dùng đối với các hệ thống IoT. Cùng với đó, IoT sẽ là một bước tiến lớn trong cuộc cách mạng công nghệ 4.0. Công nghệ hiện đại sẽ từng bước tiến gần hơn với thực tế đời sống, phục vụ và đáp ứng các nhu cầu thiết yếu trong đời sống của con người. Nâng cao giá trị cuộc sống và giúp cuộc sống trở nên đơn giản, tiện lợi hơn bao giờ hết nhờ vào các sản phẩm công nghệ IoT nói chung.

Riêng về các thiết bị đo thông số môi trường, cảnh báo đến người dùng sẽ giúp cho con người có thể chủ động phòng chống, phản ứng kịp thời trước các hiểm họa, thiên tai, tai nạn hơn so với thời kì chưa có các máy móc, công nghệ hỗ trợ. Nói riêng về thiết bị của “Hệ thống giám sát, cảnh báo cháy”, nó sẽ giúp cho người dùng có thể phản ứng kịp thời trước những rủi ro sắp xảy ra: phát hiện các địa điểm nguy hiểm, giám sát được tình trạng môi trường tại những nơi đặt cảm biến. Từ đó có thể phòng ngừa và giảm thiểu rủi ro về người và của, phản ứng kịp thời trước những rủi ro không đáng có trong các hộ gia đình. Ngoài ra, thiết bị này có thể sử dụng cho các chiến sĩ thuộc lực lượng phòng cháy, chữa cháy. Chúng sẽ là công cụ để người chỉ huy, giám sát viên có thể giám sát, biết được tình hình của các chiến sĩ thuộc lực lượng, đơn vị của mình đang ở trong môi trường như thế nào, từ đó đưa ra các phương án hành động phù hợp, đảm bảo an toàn tính mạng cho chiến sĩ cũng như người bị nạn.

Tuy nhiên, vì vấn đề về thời gian và chi phí, hệ thống giám sát, cảnh báo cháy vẫn còn đang trong quá trình hoàn thiện. Hiện tại hệ thống đã phần nào đáp ứng được công dụng cơ bản của một hệ thống giám sát và cảnh báo khi có sự cố xảy ra. Ngoài ra, tính bảo mật của hệ thống vẫn còn chưa cao vì dữ liệu truyền đi vẫn chưa được mã hóa. Kích thước của sản phẩm vẫn chưa được tối ưu theo phương án tốt nhất.

Về định hướng phát triển đề tài, hệ thống cần được đầu tư kỹ hơn về yếu tố kỹ thuật cũng như yếu tố thẩm mỹ. Về yếu tố kỹ thuật, hệ thống cần phải mã hóa dữ liệu để tăng tính bảo mật, an toàn thông tin trong quá trình truyền tải dữ liệu lên cơ sở dữ liệu và trong quá trình từ cơ sở dữ liệu trả về cho các thiết bị kết nối đến. Tăng thời gian phản hồi các thông số trên trang web để tăng tính real-time bằng cách tận dụng Socket.io của NodeJS, cũng như sử dụng các công cụ, nền tảng của NodeJS để vẽ các biểu đồ thể hiện sự thay đổi các thông số môi trường.

## DANH MỤC TÀI LIỆU THAM KHẢO

### Tiếng Việt:

1. ThS Chữ Hoài Nam, ThS Lê Minh Tuấn, Giám sát dữ liệu mạng cảm biến thông qua máy chủ dịch vụ THINGSPEAK — TẠP CHÍ CNTT&TT, Số 536 (726), trang 45-49, kỳ 1 (1/2017).

### Tiếng Anh:

1. Jiankai.li, Grove - Gas Sensor (MQ2) User Manual, 22/09/2015, Seeed Studio.
2. John C. Shovic, Raspberry Pi IoT Projects, 2016, Springer Science+Business Media New York, USA.
3. Marco Schwartz, Internet of Things with ESP8266, 07/2016, Birmingham B3 2PB, UK.

### Danh mục các Website tham khảo:

1. ATHL Solutions: <http://www.athlsolutions.com/web/ho-tro/kien-thuc-co-ban/internet-of-things-la-gi-tim-hieu-ve-internet-of-things>
2. Components101: <https://components101.com/mq2-gas-sensor>
3. HocARM: <https://hocarm.org/energia-bai-8-dht22-tiva-cam-bien-do-am-dat/>
4. Losant: <https://www.losant.com/blog/getting-started-with-the-esp8266-and-dht22-sensor>
5. TechMaster: <https://techmaster.vn/posts/33428/nodejs-la-gi-va-tai-sao-toi-nen-hoc-lap-trinh-nodejs>
6. Tinh tế: <https://tinhte.vn/threads/internet-of-things-la-gi-va-no-co-anh-huong-nhu-the-nao-trong-cuoc-song-cua-chung-ta.2129417/>
7. Waveshare: [https://www.waveshare.com/wiki/DHT22\\_Temperature-Humidity\\_Sensor](https://www.waveshare.com/wiki/DHT22_Temperature-Humidity_Sensor)