

# HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

## KHOA CÔNG NGHỆ THÔNG TIN



### BÁO CÁO BÀI TẬP LỚN

**Môn học: IOT và ứng dụng**

**Đề tài: Ứng dụng công nghệ Internet of Things trong Hệ thống cảm biến nhiệt độ và độ ẩm**

**Giảng viên: Kim Ngọc Bách**

**Nhóm 01:**

**Thành viên:**

<b>Nguyễn Thị Lan</b>	<b>B21DCCN818</b>
<b>Lê Đình Phúc</b>	<b>B21DCCN593</b>
<b>Trịnh Lê Đức</b>	<b>B21DCCN257</b>
<b>Nguyễn Quang Anh</b>	<b>B21DCCN005</b>

**Hà Nội – 2024**

## NHIỆM VỤ ĐỒ ÁN

Đề tài **Ứng dụng công nghệ Internet of Things trong Hệ thống cảm biến nhiệt độ và độ ẩm** tập trung vào các nội dung chính sau:

Giới thiệu về công nghệ Internet of Things (IoT): Kiến trúc của IoT, Các công nghệ truyền thông trong IoT.

- Tìm hiểu về cấu trúc chương trình C cho ESP32 và Arduino IDE.
- Tìm hiểu về quá trình xây dựng ứng dụng IoT trên iot-playground.
- Tìm hiểu về Cảm biến nhiệt độ, độ ẩm DHT11, Module Wifi ESP32.

Ứng dụng công nghệ IoT trong xây dựng giao diện giám sát trên nền Web theo thời gian thực, kèm theo cảnh báo nhiệt độ, độ ẩm khi vượt ngưỡng.

## LỜI CẢM ƠN

Trước tiên chúng em xin gửi lời cảm ơn chân thành sâu sắc tới các thầy cô giáo trong trường Học viện Công nghệ Bưu chính viễn thông và các thầy cô giáo trong Bộ môn Khoa Công nghệ thông tin đã tận tình giảng dạy, truyền đạt cho em những kiến thức, kinh nghiệm quý báu trong suốt thời gian qua. Đặc biệt em xin gửi lời cảm ơn đến **thầy giáo Kim Ngọc Bách** đã tận tình giúp đỡ, trực tiếp chỉ bảo, hướng dẫn em trong suốt quá trình làm báo cáo. Trong thời gian làm việc với thầy, cô, chúng em không ngừng tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiên cứu khoa học nghiêm túc, hiệu quả, đây là những điều rất cần thiết cho chúng em trong quá trình học và công tác sau này. Sau cùng xin gửi lời cảm ơn chân thành tới gia đình, bạn bè đã động viên, đóng góp ý kiến và giúp đỡ trong quá trình học tập, nghiên cứu và hoàn thành báo cáo.

## MỤC LỤC

<b>LỜI NÓI ĐẦU</b> .....	8
<b>CHƯƠNG 1. GIỚI THIỆU DỰ ÁN</b> .....	9
1. Tình hình nghiên cứu thuộc lĩnh vực đề tài.....	9
1.1. Trên thế giới.....	9
1.2. Tình hình trong nước .....	10
2. Tính cấp thiết của đề tài.....	11
3. Phạm vi nghiên cứu của đề tài.....	12
4. Mục tiêu của đề tài.....	12
5. Ứng dụng của đề tài.....	12
<b>CHƯƠNG 2. ÔN TẬP LÝ THUYẾT</b> .....	13
<b>GIỚI THIỆU VỀ CÔNG NGHỆ INTERNET OF THINGS</b> .....	13
1. Tổng quan về Internet of Things.....	13
1.1 Giới thiệu về mô hình Internet of Things (IoT).....	13
1.1.1. IoT là gì? .....	13
1.1.2. Xu hướng và tính chất của IoT .....	14
1.1.3. Những thách thức ảnh hưởng đến sự phát triển của mô hình IoT.....	16
1.2. Các công nghệ thành phần.....	17
1.2.1. Hệ thống nhúng (Embedded Systems) .....	17
1.2.2. Điện thoại di động (Mobile Telephony) .....	18
1.2.3. Đo lường từ xa và truyền thông MTM.....	19
1.2.4. Mạng cảm biến không dây và mạng cảm biến rộng.....	20
1.2.5. Máy tính di động (Mobile Computing).....	21
1.2.6. Mạng máy tính (Computer Networking).....	21
1.3. Cấu trúc mạng mở rộng.....	22
1.4. Các mô hình ứng dụng của IoT.....	24
1.4.1. Lưới điện thông minh .....	24
1.4.2. Nhà thông minh .....	24
1.4.3. Hệ thống giao thông thông minh .....	25
1.5. Phần cứng và phần mềm của một node mạng trong mô hình ứng dụng IoT.....	25
1.5.1. Phần cứng.....	25
1.5.2. Phần mềm.....	26
2. Kiến trúc tham chiếu của IoT.....	26
2.1. Tổng quan .....	26

2.2. Phân loại thiết bị IoT và phương thức kết nối Internet .....	27
2.3. Các yêu cầu của kiến trúc tham chiếu cho IoT .....	28
2.4. Mô hình tham chiếu của IoT.....	29
2.4.1. Lớp thiết bị (Devices) .....	29
2.4.2. Lớp truyền thông (Communications) .....	29
2.4.3. Lớp hợp nhất/Bus (Aggregation/ Bus).....	30
2.4.4. Lớp xử lý sự kiện và phân tích (Event Processing and Analytics).....	30
2.4.5. Lớp truyền thông ngoài (External Communication).....	30
2.4.6. Lớp quản lý thiết bị (Device Management).....	31
2.4.7. Lớp quản lý định danh và truy nhập (Identity and Access Management).....	31
3. Giới thiệu về điện toán đám mây.....	32
3.1. Giới thiệu chung .....	32
3.2. Các đặc điểm .....	32
3.2.1. Tự sửa chữa .....	32
3.2.2. Nhiều người sử dụng.....	33
3.2.3. Khả năng mở rộng tuyến tính.....	33
3.2.4. Hưởng dịch vụ.....	33
3.2.5. Điều khiển SLA (Service level agreement).....	33
3.2.6. Khả năng ảo hóa .....	33
3.2.7. Linh hoạt .....	33
4. Các công nghệ truyền thông trong IoT.....	34
4.1. Radio Frequency Identification (RFID) .....	34
4.2. Bluetooth .....	34
4.3. Zigbee .....	36
4.4. Wifi .....	37
4.5. Lora LoraWAN .....	39
5. Các Giao Thức Truyền Tin.....	40
5.1 HTTP .....	40
5.2 WebSocket .....	43
5.3 MQTT .....	45
6. Giao tiếp ngoại vi .....	49
6.1 Giao tiếp I2C .....	49
<b>CHƯƠNG 3. PHÂN TÍCH YÊU CẦU BÀI TOÁN .....</b>	<b>51</b>
1. Requirement .....	51
1.1 Mô tả bằng ngôn ngữ tự nhiên.....	51

1.1.1 Giới thiệu mục đích hệ thống .....	51
1.1.2 Phạm vi hệ thống .....	51
1.1.3 Mô tả chi tiết hoạt động nghiệp vụ .....	51
1.1.4 Thông tin các đối tượng cần xử lý, quản lý .....	52
1.1.5 Quan hệ giữa các đối tượng: .....	54
1.2. Mô tả bằng UML .....	55
1.2.1 Mô hình tổng quát .....	55
1.2.2. Mô hình chi tiết .....	56
2. Giải pháp thiết kế .....	61
2.1 Sơ đồ khối .....	61
2.2. Nguyên lý hoạt động của hệ thống .....	62
<b>CHƯƠNG 4. PHÂN TÍCH THIẾT KẾ .....</b>	<b>63</b>
1. Mô tả hệ thống: .....	63
2. API và các hàm sẽ implement .....	63
2.1 API: .....	63
2.2 Hàm: .....	64
3. Thực thi thiết kế .....	64
3.1 Thiết kế phần cứng .....	64
3.1.1 ESP32 .....	64
3.1.2. DHT11 .....	66
3.1.3. Quạt .....	69
3.1.4. Đèn LED .....	71
3.1.5. Loa Buzzer .....	72
3.1.6. Màn hình LCD I2C .....	74
3.2 Thiết kế phần mềm .....	76
3.3 Lưu đồ thuật toán .....	76
<b>CHƯƠNG 5. ĐÁNH GIÁ KẾT QUẢ DỰ ÁN VÀ KẾT LUẬN .....</b>	<b>77</b>
1. Bảng phân công công việc .....	77
2. Đánh giá kết quả dự án .....	77
3. Kết luận .....	77
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>79</b>
<b>PHỤ LỤC .....</b>	<b>80</b>

## DANH MỤC HÌNH ẢNH

<i>Hình 2. 1: Mô hình IoT .....</i>	<i>15</i>
<i>Hình 2. 2: Mô hình các công nghệ thành phần của IoT .....</i>	<i>19</i>
<i>Hình 2. 3: Mô hình hệ thống nhúng giám sát điều khiển từ xa.....</i>	<i>19</i>
<i>Hình 2. 4: Mô hình hệ thống mạng thông tin di động tế bào .....</i>	<i>21</i>
<i>Hình 2. 5: Mô hình đo lưu lượng nước lũ từ xa .....</i>	<i>22</i>
<i>Hình 2. 6: Mô hình mạng cảm biến chuyển tiếp thông tin đến trạm gốc .....</i>	<i>22</i>
<i>Hình 2. 7: Mô hình Mobile Computing .....</i>	<i>23</i>
<i>Hình 2. 8: Mô hình Computing Networking .....</i>	<i>24</i>
<i>Hình 2. 9: Mô hình mạng mở rộng.....</i>	<i>24</i>
<i>Hình 2. 10: Mô hình TCP/IP .....</i>	<i>25</i>
<i>Hình 2. 11: Kiến trúc phần cứng của đối tượng thông minh được trang bị các loại thiết bị truyền thông .....</i>	<i>28</i>
<i>Hình 2. 12: Cấu phần của IoT .....</i>	<i>29</i>
<i>Hình 2. 13: Hai mô hình kết nối của thiết bị IoT.....</i>	<i>30</i>
<i>Hình 2. 14: Mô hình tham chiếu của IoT.....</i>	<i>31</i>
<i>Hình 2. 15: Mô hình của điện toán đám mây.....</i>	<i>34</i>
<i>Hình 2. 16: Cơ chế hoạt động của thẻ RFID .....</i>	<i>36</i>
<i>Hình 2. 17: Kiến Trúc Mạng Bluetooth Piconet.....</i>	<i>37</i>
<i>Hình 2. 18: Kiến Trúc Mạng Bluetooth ScatterNet .....</i>	<i>37</i>
<i>Hình 2. 19: Mô hình mạng Zigbee.....</i>	<i>38</i>
<i>Hình 2. 20: Một số công nghệ truy cập có thể được sử dụng để truyền dữ liệu không dây và phạm vi truyền dự kiến của chúng so với băng thông .....</i>	<i>42</i>
<i>Hình 2. 21: Cấu trúc cơ bản của 1 ứng dụng web .....</i>	<i>42</i>
<i>Hình 2. 22: Sơ đồ hoạt động của HTTP.....</i>	<i>43</i>
<i>Hình 2. 23: Một số HTTP Request thường dùng .....</i>	<i>44</i>
<i>Hình 2. 24: Quy trình hoạt động của Websocket .....</i>	<i>45</i>
<i>Hình 2. 25: Sơ đồ kiến trúc WebSocket dựa trên internet .....</i>	<i>46</i>
<i>Hình 2. 26: Sơ đồ kiến trúc WebSocket dựa trên mạng nội bộ .....</i>	<i>47</i>
<i>Hình 2. 27: Mô hình Publish/Subscribe của MQTT.....</i>	<i>49</i>
<i>Hình 2. 28: Cơ chế hoạt động của MQTT.....</i>	<i>50</i>
<i>Hình 2. 29: Bus vật lý I2C .....</i>	<i>51</i>

<i>Hình 2. 30: Bus vật lý I2C .....</i>	<i>52</i>
<i>Hình 3. 1: Mô hình tổng quát.....</i>	<i>57</i>
<i>Hình 3. 2: Mô hình Quản lý các trạm.....</i>	<i>59</i>
<i>Hình 3. 3: Mô hình Quản lý thành viên .....</i>	<i>61</i>
<i>Hình 4. 1: Module Wi-Fi ESP32.....</i>	<i>66</i>
<i>Hình 4. 2: Hình DHT11 .....</i>	<i>68</i>
<i>Hình 4. 3: Kết nối DHT11 với ESP32.....</i>	<i>70</i>
<i>Hình 4. 4: Động cơ quạt mini.....</i>	<i>71</i>
<i>Hình 4. 5: Ảnh LED .....</i>	<i>72</i>
<i>Hình 4. 6: Kết nối LED với ESP .....</i>	<i>72</i>
<i>Hình 4. 7: Nguyên lý hoạt động của LED .....</i>	<i>73</i>
<i>Hình 4. 8: Piezo Buzzer .....</i>	<i>74</i>
<i>Hình 4. 9: Active Buzzer hoạt động .....</i>	<i>74</i>
<i>Hình 4. 10: Passive Buzzer hoạt động .....</i>	<i>75</i>
<i>Hình 4. 11: Kết nối ESP với Buzzer.....</i>	<i>75</i>
<i>Hình 4. 12: Màn hình LCD I2C.....</i>	<i>76</i>
<i>Hình 4. 13: Màn hình LCD I2C.....</i>	<i>77</i>
<i>Hình 4. 14: Kết nối màn hình LCD I2C với ESP.....</i>	<i>77</i>



## LỜI NÓI ĐẦU

Ngày nay khoa học công nghệ ngày càng phát triển, con người ngày càng chế tạo ra được nhiều các vật dụng thông minh hơn, hiện đại hơn. Từ đó có thể tối ưu hóa các nhu cầu của con người một cách dễ dàng hơn. Mỗi giai đoạn phát triển của lịch sử thế giới đều gắn liền với những cuộc cách mạng về khoa học kỹ thuật. Và ngày nay, cuộc cách mạng Internet of Things đã tạo nên những thay đổi đáng kể cho cuộc sống con người ở hiện tại và trong tương lai.

Internet of Things được ứng dụng vào rất nhiều mặt của cuộc sống. Ứng dụng trong công nghiệp, nông nghiệp, giáo dục, y tế... Đặc biệt là ứng dụng trong việc giám sát sự thay đổi của môi trường, và việc thu thập và đánh giá nhiệt độ, độ ẩm môi trường là một phần trong đó. Việc giám sát được nhiệt độ, độ ẩm môi trường thông qua internet là một điều mang tính ứng dụng cao. Nhiệt độ, độ ẩm là một trong những đại lượng vật lý được quan tâm nhiều do nó có vai trò quyết định trong nhiều tính chất của vật chất. Vì thế đề tài: ***Ứng dụng công nghệ Internet of Things trong Hệ thống cảm biến nhiệt độ và độ ẩm*** đảm bảo tính cấp thiết, quan trọng mà xã hội phát triển đặt ra. Đề án gồm 2 phần chính với 5 chương cụ thể như sau:

Phần thứ nhất: Cơ sở lý thuyết, tập trung giới thiệu về Internet of Things, tầm quan trọng, tính tương lai phát triển, kiến trúc và ứng dụng của IoT, các công nghệ truyền thông có thể sử dụng trong IoT. Phần này gồm 2 chương:

### ***Chương 1. Giới thiệu dự án.***

### ***Chương 2. Ôn tập lý thuyết: Giới thiệu về Công nghệ Internet of Things.***

Phần thứ hai: Tập trung vào thiết kế ứng dụng công nghệ IoT trong xây dựng giao diện giám sát trên nền Web theo thời gian thực, kèm theo cảnh báo nhiệt độ, độ ẩm khi vượt ngưỡng. Phần này gồm 2 chương:

### ***Chương 3. Phân tích yêu cầu đề tài.***

### ***Chương 4. Phân tích thiết kế.***

Phần thứ ba: Tập trung vào đánh giá kết quả thực hiện dự án và đưa ra kết luận cho đề tài. Phần này gồm chương:

### ***Chương 5. Đánh giá kết quả dự án và kết luận.***

## CHƯƠNG 1. GIỚI THIỆU DỰ ÁN

### 1. Tình hình nghiên cứu thuộc lĩnh vực đề tài

Mỗi giai đoạn phát triển của lịch sử thế giới đều gắn liền với những cuộc cách mạng về khoa học kỹ thuật. Và ngày nay, cuộc cách mạng Internet of Things đã tạo nên những thay đổi đáng kể cho cuộc sống con người ở hiện tại và trong tương lai.

Với sự phát triển của Internet, smartphone và đặc biệt là các thiết bị cảm biến, Internet of Things (IoT) đang trở thành xu hướng mới của thế giới. IoT được định nghĩa là những vật dụng có khả năng kết nối Internet. Bật vào nhà, mở khóa cửa, đèn sẽ tự động sáng chỗ bạn đứng, điều hòa sẽ tự động điều chỉnh nhiệt độ, nhạc sẽ tự động bật để chào đón bạn... những điều chỉ có trong phim khoa học viễn tưởng, đang dần trở thành hiện thực với công nghệ IoT.

Nhiệt độ, độ ẩm là một trong những đại lượng vật lý được quan tâm nhiều do nó có vai trò quyết định trong nhiều tính chất của vật chất. Một trong những đặc điểm của nhiệt độ là làm thay đổi liên tục các đại lượng chịu sự ảnh hưởng của nó, ví dụ như áp suất, thể tích của một chất khí. Nó cũng là yếu tố ảnh hưởng rất lớn đến chính con người chúng ta và tất cả sự sống trên trái đất. Ngày nay khi nền công nghiệp phát triển mạnh, sản phẩm tạo ra ngày càng phong phú thì nhu cầu lưu trữ hàng hóa trong các nhà kho càng lớn. Việc kiểm soát nhiệt độ, độ ẩm khi bảo quản, lưu trữ sản phẩm công – nông nghiệp trong các nhà kho là rất quan trọng. Cũng giống như vậy, trong các phòng thí nghiệm trong bệnh viện, trong các nhà kính trồng cây cảnh, khu sản xuất rau sạch... việc đảm bảo nhiệt độ, độ ẩm cũng là yếu tố quan trọng quyết định đến chất lượng sản phẩm...

#### 1.1. Trên thế giới

Theo Ericsson Mobility Report 2024, số lượng thiết bị kết nối IoT trên toàn cầu tiếp tục gia tăng mạnh mẽ. Dự báo cho thấy số lượng kết nối IoT di động sẽ đạt khoảng 4 tỷ vào cuối năm 2024, tăng từ 3,4 tỷ vào cuối năm 2023. Các kết nối IoT diện rộng và ngắn hạn cũng đang trên đà phát triển nhanh chóng, với dự kiến tổng cộng 15,7 tỷ kết nối IoT vào năm 2023 và có thể đạt 38,8 tỷ vào năm 2029. Đặc biệt, các công nghệ IoT diện rộng, như NB-IoT và Cat-M, tiếp tục mở rộng, nhờ sự nâng cấp hạ tầng mạng hỗ trợ IoT trên toàn cầu.

IoT đang diễn ra một cách mạnh mẽ. 50% doanh nghiệp đã bắt đầu triển khai những dự án về IoT. IoT mang lại một cơ hội doanh thu cho rất nhiều ngành và những giải pháp đó bắt đầu

thương mại hóa với tốc độ rất nhanh. Ngành dịch vụ tiện ích, giao thông, tòa nhà thông minh và các ngành bán lẻ là những ngành đi đầu trong việc ứng dụng IoT.

Trên thế giới đã và đang nghiên cứu một số hệ thống, thiết bị đo nhiệt độ trong tất cả các lĩnh vực có thể kể đến như:

Nhà chế tạo chip vi điều khiển hàng đầu thế giới – Atmel đã cho ra mắt dòng chip đo nhiệt độ có tên gọi AT30TS750, truyền thông theo giao thức I<sup>2</sup>C với bộ nhớ EEPROM tích hợp.

Alfredo Milani Comparetti đã cho ra đời phần mềm Speedfan theo dõi điện áp, tốc độ quạt và nhiệt độ trong máy tính với màn hình phần cứng chip.

Tại Úc các nhà vật lý học thuộc Đại học Adelaide tuyên bố đã chế tạo thành công nhiệt kế chính xác nhất thế giới.

Hãng Cypress Microsystem đã cho ra đời công nghệ PSOC (Programmable System On Chip) để có thể phát triển các thiết bị đo nhiệt độ dựa trên công nghệ này.

## ***1.2. Tình hình trong nước***

Trước việc IoT phát triển mạnh mẽ trên thế giới, Việt Nam cũng là một nước đón đầu xu thế mới.

Tại Việt Nam, IoT đã được ứng dụng từ lâu dưới các hình thức tự động hóa như hệ thống điều khiển đèn giao thông, hệ thống tưới tiêu tự động ..., tuy nhiên chỉ đến năm 2015 thì khái niệm Internet of Things (IoT) mới được nhắc đến nhiều thông qua các hội thảo, hội nghị về xu hướng công nghệ của Cisco, Intel, Hội Tin học TP.HCM và một số công ty trong nước như MobiFone, DTT, Sao Bắc Đẩu.

Xu hướng phát triển Internet of Things (IoT) tại Việt Nam năm 2024 đang tiếp tục bùng nổ, với tốc độ tăng trưởng mạnh mẽ nhờ sự thúc đẩy từ nhiều ngành công nghiệp và các chính sách hỗ trợ của chính phủ. Thị trường IoT tại Việt Nam được dự báo sẽ đạt giá trị **3,7 tỷ USD vào năm 2023** và tiếp tục tăng trưởng với tốc độ CAGR khoảng **16,2%** đến năm 2029. Sự phát triển này được thúc đẩy bởi nhu cầu ngày càng cao về số hóa trong các ngành công nghiệp và việc triển khai các dự án thành phố thông minh

Nhằm phục vụ cho yêu cầu sản xuất trong tất cả các ngành kinh tế, nông nghiệp, và công nghiệp... Đã có nhiều báo cáo, tài liệu nghiên cứu về vai trò và các giải pháp kiểm soát nhiệt độ như:

Nghiên cứu thiết kế và chế tạo hệ thống đo giám sát nhiệt, ẩm cho các kho nông lâm sản có môi trường khắc nghiệt của Phạm Minh Tuấn, Phạm Thượng Cát, Trần Đức Minh, *Viện Công nghệ Thông Tin, Viện Khoa học và Công nghệ Việt Nam*.

Nghiên cứu ảnh hưởng của nhiệt độ đến tính chất cơ lý của mặt đường bê tông nhựa của TS. Nguyễn Thống Nhất và ThS. Trần Văn Thiện, mục đích của đề tài nhằm tạo được mặt đường bê tông nhựa có chất lượng, có cường độ ổn định và tuổi thọ cao.

Đề tài nghiên cứu khoa học độc lập cấp nhà nước “Ảnh hưởng của Enso đến các cực trị nhiệt độ và lượng mưa ở Việt Nam và khả năng dự báo” của Nguyễn Đức Ngữ, *Trung tâm Khoa học công nghệ Khí tượng Thủy văn và Môi trường* đã làm sáng tỏ cơ chế tác động của ENSO và đánh giá khả năng dự báo mùa đối với sự xuất hiện các cực trị nhiệt độ và lượng mưa trên cơ sở các thông tin, nhận thức về ENSO.

## **2. Tính cấp thiết của đề tài**

Hiện nay trong đời sống cũng như xã hội việc ứng dụng công nghệ Internet of Things là rất cần thiết, con người có thể điều khiển mọi thứ xung quanh mình nhanh và hiệu quả hơn. Vì vậy chúng ta tìm hiểu về nó là một điều thật sự đúng đắn.

Hiện nay trong đời sống và tất cả các lĩnh vực khác về kinh tế, quốc phòng, công nghiệp, nông nghiệp,... việc đo và cảnh báo nhiệt độ, độ ẩm đóng vai trò hết sức quan trọng vì nó liên quan đến chất lượng sản phẩm. Ví dụ như cảnh báo quá nhiệt trong lò luyện kim, cán thép, hệ thống đo nhiệt độ và độ ẩm môi trường đất để phục vụ tưới tiêu.... Do đó việc nghiên cứu và thiết kế các hệ thống đo và cảnh báo nhiệt độ, độ ẩm ngày càng đòi hỏi độ chính xác cao đáp ứng nhu cầu của thị trường. Cùng với sự phát triển nhanh chóng của công nghệ và hệ thống thông tin thì việc đo và cảnh báo an toàn về nhiệt độ, độ ẩm sẽ được đảm bảo chính xác và từ đó kịp thời hạn chế các hậu quả do nhiệt độ, độ ẩm gây ra ở mức thấp nhất. Các hệ thống đo và cảnh báo nhiệt độ, độ ẩm sẽ giúp cho con người chủ động hơn trong việc cảnh báo an toàn cho các hệ thống dây chuyền, các kho bảo quản sản phẩm, cũng như cảnh báo cho chính con người chúng ta. Do vậy

việc thực hiện đề tài: ***Ứng dụng công nghệ Internet of Things trong Hệ thống cảm biến nhiệt độ và độ ẩm*** đảm bảo tính cấp thiết, quan trọng trong sự phát triển của thế giới hiện đại.

### **3. Phạm vi nghiên cứu của đề tài**

Đề tài **Ứng dụng công nghệ Internet of Things trong Hệ thống cảm biến nhiệt độ và độ ẩm** có thể được ứng dụng thực tiễn trong các ngôi nhà thông minh, ứng dụng trong nông nghiệp ở các nhà kho sự đảm bảo về nhiệt độ độ ẩm(kho chứa lương thực, kho để đồ,...). Ứng dụng trong công nghiệp để đảm bảo tính chính xác trong sản xuất. Sản phẩm cũng có thể được ứng dụng trong giám sát, quan trắc, thời tiết môi trường. Dựa vào sự thay đổi của nhiệt độ độ ẩm môi trường theo thời gian để có những dự báo cũng như biện pháp phù hợp với khu vực cần khảo sát thực nghiệm.

### **4. Mục tiêu của đề tài**

Đề tài ứng dụng công nghệ Internet of Thing trong thu thập và đánh giá nhiệt độ độ ẩm môi trường có mục tiêu cần đạt được là:

- Nghiên cứu và tìm hiểu lý thuyết chung về công nghệ Internet of Things
- Ứng dụng công nghệ Internet of Things vào việc đo nhiệt độ và độ ẩm trong nhà, ứng dụng trong các nhà kho lưu trữ yêu cầu các mức nhiệt độ và độ ẩm phù hợp.

### **5. Ứng dụng của đề tài**

- **Đo nhiệt độ và độ ẩm trong phòng và kho:** Hệ thống này có thể được sử dụng để giám sát điều kiện môi trường trong phòng và kho, đảm bảo môi trường sống và lưu trữ thoải mái.
- **Dữ liệu nhiệt độ và độ ẩm sẽ được hiển thị trên màn hình LCD và truyền tới web app.**
- **Cảnh báo khi vượt ngưỡng:** Khi nhiệt độ vượt quá 30°C hoặc độ ẩm vượt quá 70%, hệ thống sẽ bật quạt, báo hiệu bằng đèn LED và loa buzzer. Bạn có thể điều chỉnh ngưỡng nhiệt độ và độ ẩm theo nhu cầu cụ thể của mình.
- **Điều khiển từ xa:** Từ web app, người dùng có thể điều khiển quạt, đèn LED và loa buzzer.

## CHƯƠNG 2. ÔN TẬP LÝ THUYẾT

## GIỚI THIỆU VỀ CÔNG NGHỆ INTERNET OF THINGS

## 1. Tổng quan về Internet of Things

### 1.1 Giới thiệu về mô hình Internet of Things (IoT)

### 1.1.1. IoT là gì?

IoT (Internet of Things) là một mạng lưới gồm các đối tượng có khả năng kết nối Internet và tác động qua lại giữa các dịch vụ web.

IoT không chỉ là các máy "giao tiếp" với nhau mà còn nhiều thứ khác nữa, bao gồm khả năng thay đổi hoàn toàn thế giới, cả trong cuộc sống và cách chúng ta cảm nhận trong thực tế.



### Hình 2. 1: Mô hình 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 tính kết nối. 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 "hiện nay máy tính và Internet gần như phụ thuộc hoàn toàn vào con người để chuyển tải dữ liệu. Gần như tất cả trong số 50 Petabyte dữ liệu đang có trên Internet (vào thời điểm đó) đều được ghi lại hoặc tạo ra bởi con người chúng ta, thông qua các cách thức như gõ chữ, nhấn nút, chụp ảnh, quét mã vạch... Con người chính là nhân tố quyết định trong thế giới Internet hiện nay. Thế nhưng con người lại có nhiều nhược điểm: chúng ta chỉ có thời gian hạn chế, khả năng tập trung và độ chính xác cũng ở mức thấp so với máy móc. Điều đó có nghĩa là chúng ta không giỏi trong việc thu thập thông tin về thế giới xung quanh, và đây là một vấn đề lớn.

Một quan điểm khác lại cho rằng, IoT gồm các đối tượng thông minh có thể được điều khiển và tương tác với những đối tượng có thể đáp ứng tương tác từ xa, hay có thể làm việc độc lập nhằm cung cấp các dịch vụ và giải pháp mà không cần sự can thiệp của con người.

Hiện nay, IoT đang trải qua giai đoạn phát triển "bộc phát" và điều này xảy ra nhờ vào một số nhân tố, trong đó gồm IPv6, 4G, chi phí, tính sẵn có của công nghệ. Trong những năm tiếp theo, bạn sẽ thấy ngày càng có nhiều thiết bị trên thị trường. Những thách thức đang diễn ra là quản lý dữ liệu và chuyển sang IPv6 (IPv6 đã sẵn sàng và chạy với địa chỉ đã được cấp phát. IPv4 đã cạn kiệt và 2011 chỉ còn lại những địa chỉ cuối cùng).

### ***1.1.2. Xu hướng và tính chất của IoT***

#### ***1.1.2.1. Sự 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.

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.

#### ***1.1.2.2. 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 sensor chính là một thành phần đơn giản của IoT.

#### ***1.1.2.3. 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.

#### ***1.1.2.5. 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.

#### ***1.1.2.6. Luồng năng lượng mới***

Hiện nay, IoT đang trải qua giai đoạn phát triển "bộc phát" và điều này xảy ra nhờ vào một số nhân tố, trong đó gồm IPv6, 4G, chi phí, tính sẵn có của công nghệ.

ARM đã "nhanh chân" trong việc nhận ra rằng, ổ đĩa có xu hướng sử dụng các bộ vi điều khiển 32-bit là giải pháp cho những người có ý định thực hiện một số quyết định của riêng họ theo một cách tự động. Gary tin rằng, khả năng của các bộ vi điều khiển này ngày càng tăng, điều này có nghĩa là người dùng có thể làm những điều mà trước đây là bất khả.

Axel Pawlik, Giám đốc quản lý của RIPE NCC lý giải tại sao IPv6 cần thiết cho tương lai của IoT, với IPv6 chúng ta sẽ có lượng địa chỉ phong phú và điều này sẽ mở ra khả năng gán địa chỉ cho mỗi thiết bị (gadget) và chip. Các giải pháp sẽ dễ dàng và đơn giản hơn, rõ ràng hơn, có thể phục hồi đến từng mục địa chỉ riêng, và phạm vi phát triển vô cùng to lớn.



### ***1.1.3. Những thách thức ảnh hưởng đến sự phát triển của mô hình IoT***

#### ***1.1.3.1. Chưa có sự chuẩn hóa***

Sự chuẩn hóa ở đây được hiểu như là một ngôn ngữ giao tiếp 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 giao tiếp với nhau (ví dụ như bạn có thể đi từ Việt Nam đến Mỹ, nhưng không đảm bảo rằng bạn 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 hẳn bạn 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, ...vv. Những giao thức như thế này hoạt động ổn 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 hiểu nhau. 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.

#### ***1.1.3.2. 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, những thiết bị vẫn hoàn toàn nói được với nhau thông qua chức năng phiên dịch của máy chủ. Thế nhưng mọi chuyện không đơn giản như thế, cứ mỗi một mạng lưới như thế tạo thành một subnetwork riêng, và vấn đề là các máy móc nằm trong subnetwork này không thể giao tiếp tốt với subnetwork khác.

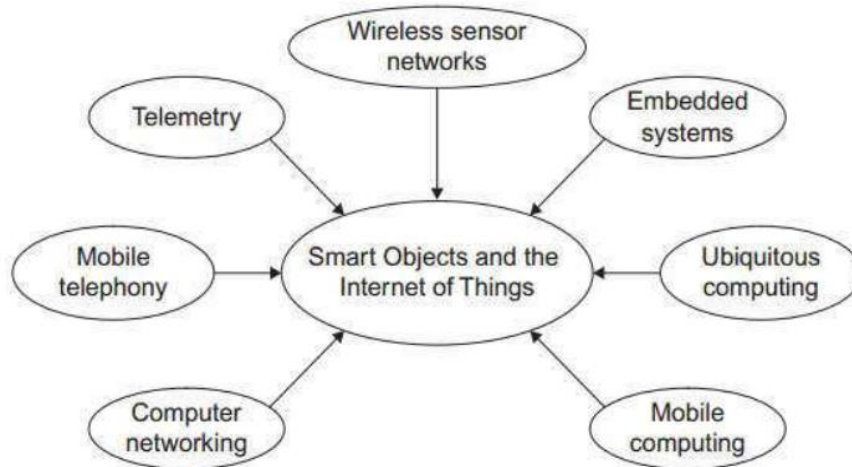
Một số trong những vấn đề nói trên chỉ đơn giản là vấn đề về kiến trúc mạng, về kết nối mà các thiết bị sẽ liên lạc với nhau (Wifi, Bluetooth, NFC,...). Những thứ này thì tương đối dễ khắc phục với công nghệ không dây ngày nay. Còn với các vấn đề về giao thức thì phức tạp hơn rất nhiều, nó chính là vật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

#### ***1.1.3.3. Chi phí phát triển mạng***

Cách duy nhất để các thiết bị IoT có thể thật sự giao tiếp đó là khi có một động lực kinh tế đầy mạnh khiến các nhà sản xuất đồng ý chia sẻ quyền điều khiển cũng như dữ liệu mà các

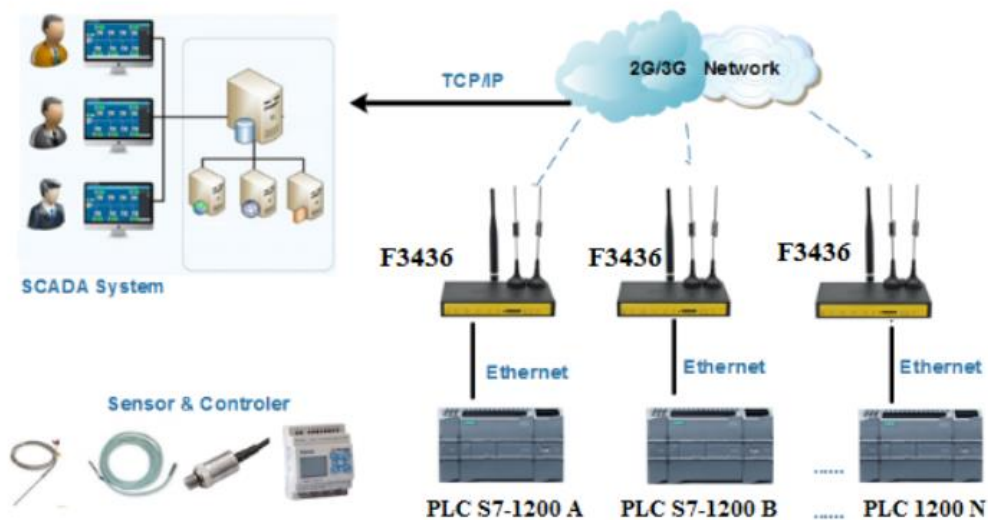
thiết bị của họ thu thập được, hiện tại các động lực này không nhiều. Ví dụ: một công ty thu gom rác muốn kiểm tra xem các thùng rác có đầy hay chưa. Khi đó, họ phải gặp nhà sản xuất thùng rác, đảm bảo rằng họ có thể truy cập vào hệ thống quản lý của từng thùng một. Điều đó khiến chi phí bị đội lên, và công ty thu gom rác có thể đơn giản chọn giải pháp cho một người chạy xe kiểm tra từng thùng một.

## 1.2. Các công nghệ thành phần



**Hình 2. 2: Mô hình các công nghệ thành phần của IoT**

### 1.2.1. Hệ thống nhúng (Embedded Systems)



**Hình 2. 3: Mô hình hệ thống nhúng giám sát điều khiển từ xa**

Hệ thống nhúng (Embedded system) là một thuật ngữ để chỉ một hệ thống có khả năng tự trị được nhúng vào trong một môi trường hay một hệ thống trung tâm. Đó là các hệ thống tích hợp cả phần cứng và phần mềm phục vụ các bài toán chuyên dụng trong nhiều lĩnh vực công nghiệp, tự động hóa điều khiển, quan trắc và truyền tin.

Các hệ thống nhúng được thiết kế để thực hiện một số nhiệm vụ chuyên dụng chứ không phải đóng vai trò là các hệ thống máy tính đa chức năng. Một hệ thống nhúng thường không phải là một khối riêng biệt mà là một hệ thống phức tạp nằm trong thiết bị mà nó điều khiển. Phần mềm được viết cho các hệ thống nhúng được gọi là firmware và được lưu trữ trong các chip bộ nhớ ROM hoặc bộ nhớ flash. Phần mềm thường chạy với số tài nguyên phần cứng hạn chế: không có bàn phím, màn hình hoặc có nhưng với kích thước nhỏ, dung lượng thấp.

***Kiến trúc phần mềm của hệ thống nhúng:***

Vòng lặp kiểm soát đơn giản.

Hệ thống ngắt điều khiển.

Đa nhiệm tương tác.

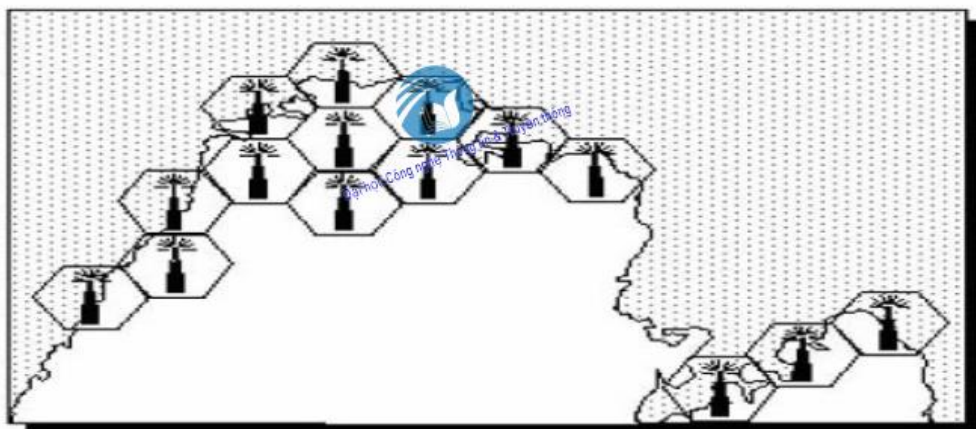
Đa nhiệm ưu tiên.

Vi nhân (Microkernel) và nhân ngoại (Exokernel).

Nhân khối (Monolithic kernels).

***1.2.2. Điện thoại di động (Mobile Telephony)***

Mobile Telephony (Điện thoại di động) được trưởng thành từ ngành công nghiệp điện thoại với những hứa hẹn trong mạng điện thoại truy nhập rộng khắp.



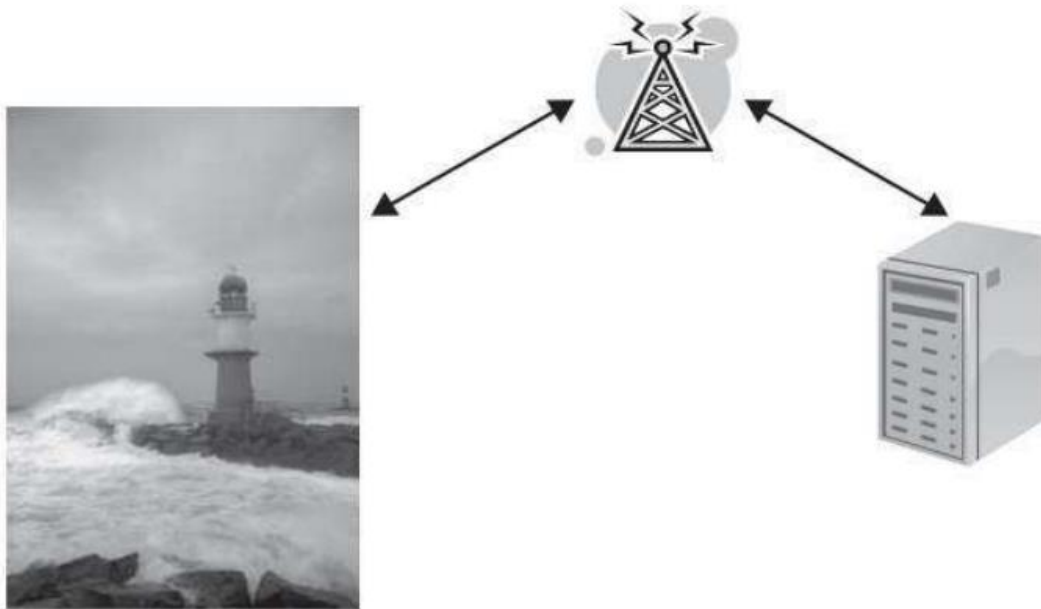
**Hình 2. 4: Mô hình hệ thống mạng thông tin di động tế bào**

Điện thoại di động thường được gọi là điện thoại tế bào bởi vì trong điện thoại di động hoạt động theo cấu trúc của mạng không dây. Các mạng được chia thành các tế bào nơi mỗi điện thoại được kết nối với một tế bào xác thực tại bất kỳ thời điểm nào. Một tế bào bao gồm một khu vực vật lý có kích thước được xác định bởi các nhà điều hành mạng. Vì mỗi tế bào thường xử lý một số lượng hạn chế các cuộc gọi điện thoại đồng thời, các nhà khai thác mạng có kế hoạch mạng lưới của họ để các tế bào nhỏ hơn và nhiều hơn ở những nơi khai thác mong đợi nhiều người thực hiện cuộc gọi điện thoại. Mỗi tế bào được điều hành bởi một tháp di động trên đó một trạm thu phát không dây được gắn kết.

Với chiếc điện thoại thông minh hiện đại, truy cập Internet không còn giới hạn với máy tính, nó đã thực sự phổ biến. Với một vài nút bấm nhanh, email, tin nhắn nhanh, và World Wide Web( mạng lưới toàn cầu) sẽ có sẵn ngay lập tức.

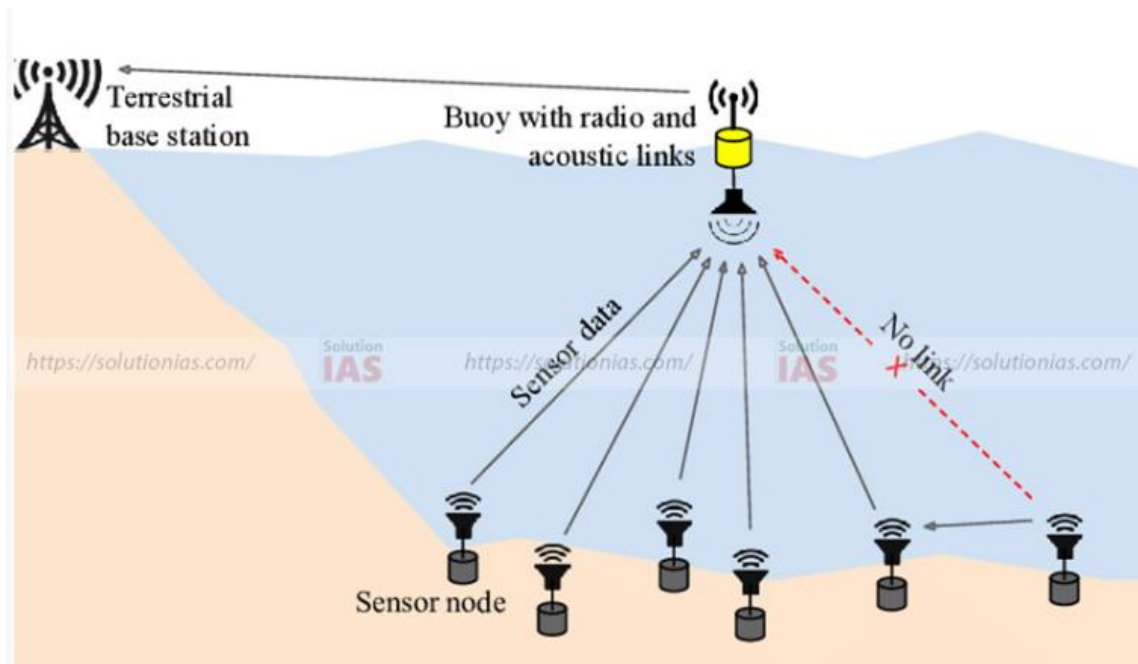
### **1.2.3. Đo lường từ xa và truyền thông MTM**

Telemetry là một từ ghép Hy Lạp giữa tele (từ xa) và metron (đo đạc). Telemetry là việc thực hiện các phép đo từ xa. Truyền thông machine-to-machine là sự tổng quan của telemetry với hàm ý là sự giao tiếp hoạt động của truyền thông tự điều khiển là trung tâm của khái niệm telemetry.



**Hình 2. 5: Mô hình đo lưu lượng nước lũ từ xa**

#### **1.2.4. Mạng cảm biến không dây và mạng cảm biến rộng**



**Hình 2. 6: Mô hình mạng cảm biến chuyển tiếp thông tin đến trạm gốc**

Mạng cảm biến không dây đã phát triển từ ý tưởng rằng các cảm biến không dây nhỏ có thể được sử dụng để thu thập thông tin từ các môi trường vật lý trong một số lượng lớn các trường hợp khác nhau, từ theo dõi cháy rừng và quan sát động vật để quản lý nông nghiệp và

giám sát công nghiệp. Mỗi một cảm biến không dây truyền thông tin tới một trạm gốc. Cảm biến giúp đỡ lẫn nhau để chuyển tiếp thông tin đến các trạm cơ sở.

#### ***1.2.5. Máy tính di động (Mobile Computing)***



***Hình 2. 7: Mô hình Mobile Computing***

Máy tính di động là lĩnh vực truyền thông không dây và kết chuyển xung quanh (hoán vị vòng quanh) máy tính, chẳng hạn như máy tính xách tay. Trong một số cách thức lĩnh vực máy tính di động tách ra từ việc được khởi tạo ở khắp nơi trong khu vực máy tính. Tương tự như vậy, sự tập trung ban đầu về mạng không dây đã dẫn tới nghiên cứu cơ chế truyền thông không dây. Làm việc trên các cơ chế này bắt đầu vào giữa những năm 1980 và đã dẫn đến các tiêu chuẩn trên toàn mạng cục bộ không dây (WiFi) mà bắt đầu hình thành vào cuối năm 1990.

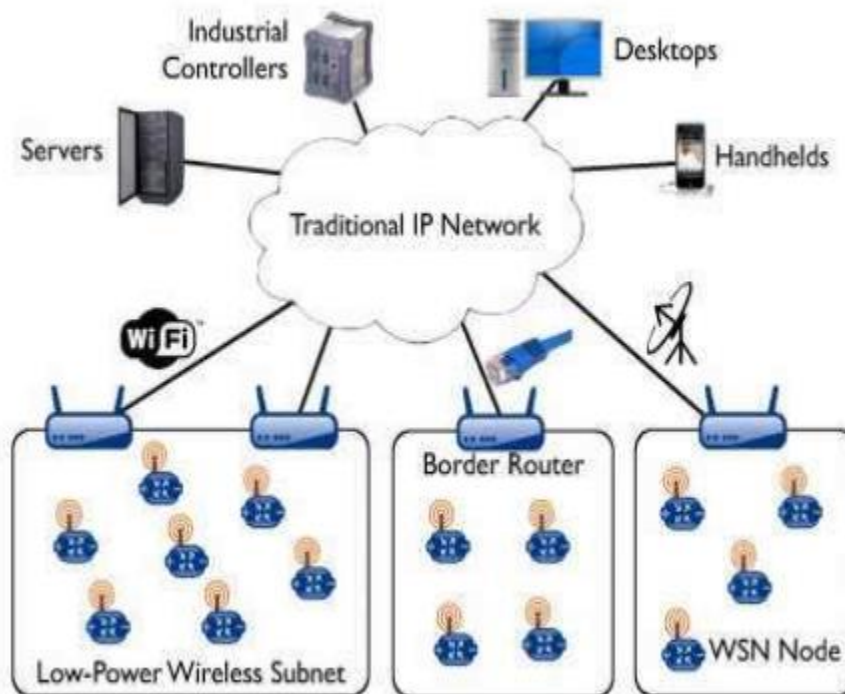
#### ***1.2.6. Mạng máy tính (Computer Networking)***

Mạng máy tính là giới thiệu về việc kết nối các máy tính để cho phép chúng giao tiếp với nhau. Các mạng này ban đầu đều có dây, nhưng với sự ra đời của máy tính di động, các mạng không dây có sẵn.



*Hình 2. 8: Mô hình Computing Networking*

### *1.3. Cấu trúc mạng mở rộng*

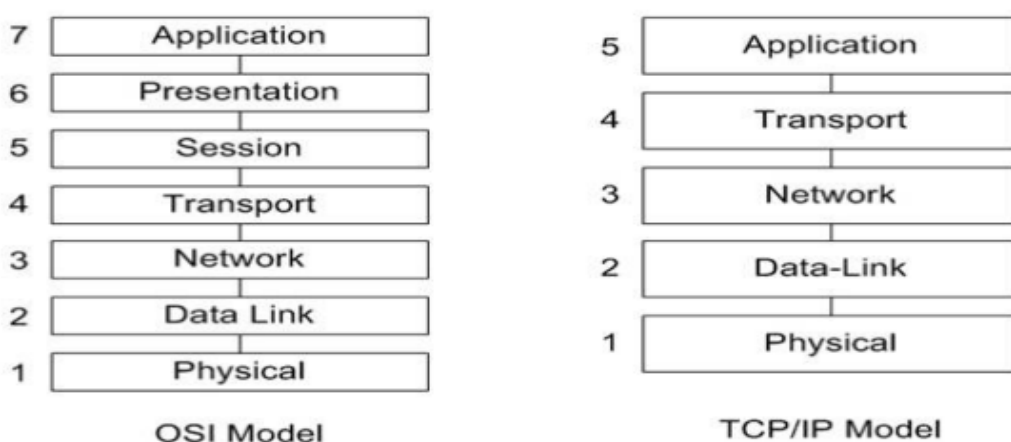


*Hình 2. 9: Mô hình mạng mở rộng*



Kiến trúc REST (Representational State Transfer) chính là lõi chính của web hiện tại và sử dụng URIs (Universal Resource Identifiers) cho việc bao bọc và nhận dạng dịch vụ. Chúng tập trung vào việc tạo ra các cặp dịch vụ lỏng lẻo cho các ứng dụng để từ đó chúng có thể dễ dàng được chia sẻ và sử dụng lại. Kiến trúc REST bao gồm các client và các server nhờ đó các client khởi tạo các yêu cầu đến các server, các server này sẽ xử lý các yêu cầu và trả về đáp ứng thích hợp. Trong triển khai Web, những thành phần chính rộng lớn của kiến trúc REST ngày nay sử dụng giao thức truyền siêu văn bản HTTP như là giao thức ứng dụng, nó sẽ thao tác tài nguyên bằng các phương thức riêng của nó GET, PUT, POST ... tài nguyên được tách đôi ra từ biểu diễn của chúng và từ đó được biểu diễn tùy ý trong các định dạng khác nhau như là JSON hoặc XML. Tuy nhiên HTTP không được thiết kế với bộ nhớ hạn chế của các đối tượng thông minh. Kích thước gói và bộ nhớ được yêu cầu dành cho TCP và các tiêu đề văn bản được phù hợp cho các hạn chế của các node mạng cảm biến.

IoT là một kiến mạng tổng hợp của nhiều thành phần mạng khác nhau trong một cấu trúc chung, hệ thống gồm rất nhiều các thành phần khác nhau và tuân theo kiến trúc IP dựa trên giao thức TCP/IP. Bộ ứng dụng giao thức IP đã được nâng cao để hỗ trợ truyền thông đa điểm, chất lượng dịch vụ (QoS), kỹ thuật lưu lượng, và các dịch vụ thời gian thực với kiến trúc được duy trì trong một thời gian dài. Kiến trúc phân lớp là một trong những cơ sở của thiết kế và nó tạo thành một kiến trúc cực kỳ linh hoạt. Kiến trúc phân lớp được biết đến với mô hình năm lớp hiện nay của giao thức TCP/IP và mô hình OSI bảy lớp.



**Hình 2. 10: Mô hình TCP/IP**



## **1.4. Các mô hình ứng dụng của IoT**

### **1.4.1. Lưới điện thông minh**

Lưới điện thông minh (hệ thống điện thông minh) là hệ thống điện có sử dụng các công nghệ thông tin và truyền thông để tối ưu việc truyền dẫn, phân phối điện năng giữa nhà sản xuất và hộ tiêu thụ, hợp nhất cơ sở hạ tầng điện với cơ sở hạ tầng thông tin liên lạc. Có thể coi hệ thống điện thông minh gồm có 2 lớp: lớp 1 là hệ thống điện thông thường và bên trên nó là lớp 2 hệ thống thông tin, truyền thông, đo lường.

Lưới điện thông minh phát triển trên 4 khâu:

- + Phát điện: Smart Generation.
- + Truyền tải: Smart Transmission.
- + Phân phối: Smart Distribution.
- + Tiêu thụ: Smart Power Consumers

### **1.4.2. Nhà thông minh**

Nhà thông minh (Smart Home) là một ngôi nhà được trang bị hệ thống tự động tiên tiến dành cho điều khiển đèn chiếu sáng, nhiệt độ, truyền thông đa phương tiện, an ninh, rèm, cửa và nhiều tính năng khác nhằm mục đích làm cho cuộc sống ngày càng tiện nghi, an toàn và góp phần sử dụng hợp lý các nguồn tài nguyên. Các thành phần cơ bản của một ngôi nhà thông minh bao gồm:

- + Một máy tính điều khiển trung tâm (Home server).
- + Các thiết bị gia dụng đầu cuối.
- + Hệ thống các phần mềm điều khiển ngôi nhà.
- + Các thiết bị điều khiển.

Các chức năng chính của một ngôi nhà thông minh bao gồm:

- + Điều khiển hệ thống chiếu sáng.
- + Hệ thống an ninh, báo động, báo cháy.
- + Điều khiển điều hòa, máy lạnh.

- + Điều khiển thiết bị điện nói chung.
- + Hệ thống âm thanh đa vùng.
- + Camera, chuông hình.
- + Hệ thống bảo vệ nguồn điện.
- + Các tiện ích và ứng dụng khác.

### ***1.4.3. Hệ thống giao thông thông minh***

Hệ thống giao thông thông minh là những ứng dụng tiên tiến, nhằm mục đích để cung cấp các dịch vụ mới liên quan đến phương thức vận tải và quản lý giao thông cho phép người dùng có thông tin tốt hơn khi tham gia mạng lưới giao thông. Nói cách khác hệ thống giao thông thông minh là hệ thống giao thông mà ở đó công nghệ thông tin và truyền thông được áp dụng trong lĩnh vực vận tải đường bộ và các hình thức vận tải khác bao gồm các cơ sở hạ tầng, phương tiện giao thông và người dùng. Hệ thống giao thông thông minh bao gồm các chức năng cơ bản sau:

Các trang thiết bị đo lường, thu thập thông tin như: cảm biến, đồng hồ, camera, điện thoại thông minh, các thiết bị sinh trắc học giúp hệ thống có thể đo lường, nhận biết phương tiện giao thông và các hoạt động, hành vi giao thông (phạm lỗi, gây tai nạn, tắc nghẽn,...).

Trung tâm xử lý thông tin giao thông sẽ tự động phát hiện, xử lý thông tin được thu thập, xác định mẫu và mối quan hệ giữa các mẫu thu được để đưa ra phương án xử lý kịp thời, tương tác theo thời gian thực tới phương tiện, người tham gia giao thông.

Trung tâm giám sát, vận hành hệ thống. Hệ thống biển báo, thông báo thông tin giao thông tới phương tiện và người tham gia giao thông. Hệ thống thanh toán được tích hợp.

## ***1.5. Phần cứng và phần mềm của một node mạng trong mô hình ứng dụng IoT***

### ***1.5.1. Phần cứng***

Cấu trúc phần cứng cơ bản của một nút mạng trong mô hình ứng dụng IoT bao gồm thiết bị truyền thông, một bộ vi xử lý, các cảm biến/thiết bị truyền động và nguồn cung cấp. Các cảm biến/thiết bị truyền động cho phép các nút mạng có thể tương tác được với thế giới vật lý. Bộ vi xử lý cho phép các nút mạng có thể chuyển đổi dữ liệu nhận được từ các cảm biến... Thiết bị truyền thông cho phép các mạng có thể truyền thông được với nhau. Thông qua thiết bị truyền

thông, các nút mạng có thể gửi dữ liệu cảm biến đọc được ra thế giới bên ngoài và nhận dữ liệu từ các nút mạng khác.

***Hình 2. 11: Kiến trúc phần cứng của đối tượng thông minh được trang bị các loại thiết bị truyền thông***

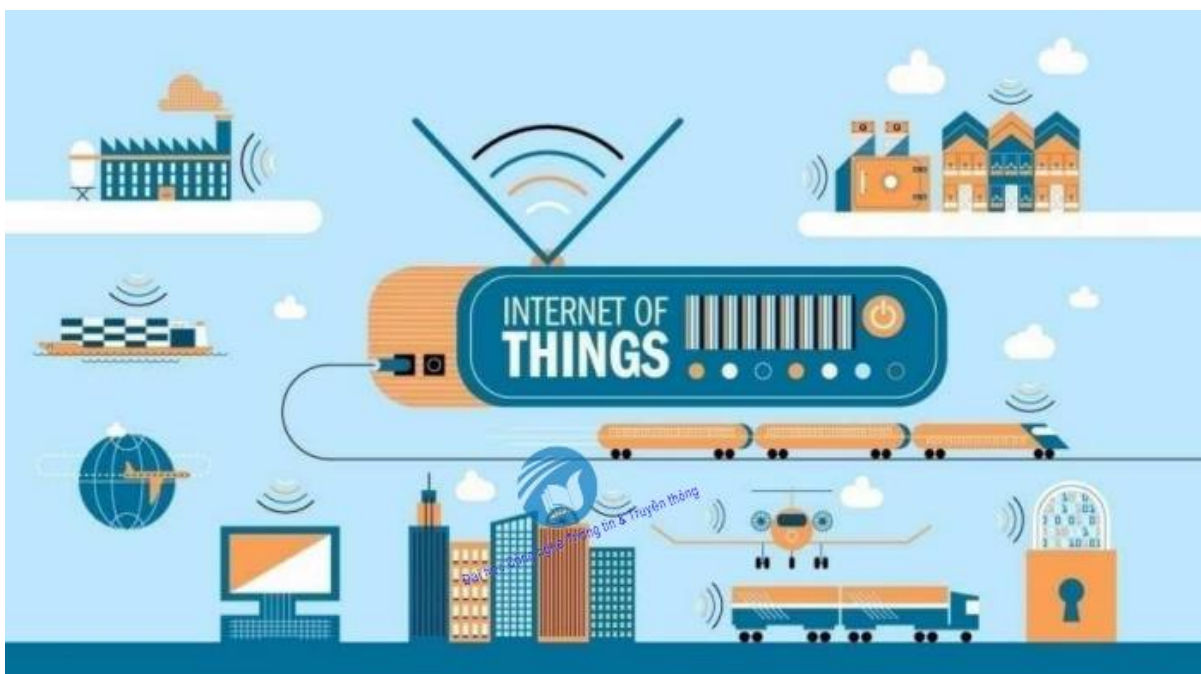
### ***1.5.2. Phần mềm***

Mọi hoạt động của một nút mạng được xác định bởi phần mềm chạy trên bộ vi điều khiển. Các chương trình phần mềm thường được viết tương tự như các chương trình phần mềm cho các máy tính. Các chương trình được viết bằng một ngôn ngữ lập trình, chẳng hạn như C và được biên dịch với một trình biên dịch mã máy cho vi điều khiển. Các mã máy được ghi vào ROM của bộ vi điều khiển khi các node cảm biến được sản xuất. Khi các nút mạng được bật nguồn thì bộ vi điều khiển sẽ chạy các phần mềm. Chức năng cơ bản của hệ điều hành bao gồm việc trừu tượng hóa tài nguyên cho các thiết bị phần cứng khác nhau, quản lý ngắt và lập lịch các nhiệm vụ, điều khiển đồng thời và hỗ trợ mạng. Dựa trên các dịch vụ được cung cấp bởi hệ điều hành, người lập trình ứng dụng có thể thuận tiện sử dụng các giao diện lập trình ứng dụng mức cao (APIs) độc lập với phần cứng lớp dưới.

## **2. Kiến trúc tham chiếu của IoT**

### ***2.1. Tổng quan***

Kiến trúc IoT được đại diện cơ bản bởi 4 phần: Vạn vật (Things), trạm kết nối (Gateways), hạ tầng mạng và điện toán đám mây (Network and Cloud) và các lớp tạo và cung cấp dịch vụ (Services-creation and Solutions Layers).



**Hình 2. 12: Cấu phần của IoT**

## **2.2. Phân loại thiết bị IoT và phương thức kết nối Internet**

Những thiết bị IoT có kích thước nhỏ nhất dùng bộ điều khiển 8 bit nhúng, kiểu cả hệ thống trên chip SoC (System on Chip) và thường không có hệ điều hành. Mức cao hơn là hệ thống dựa trên chip ARM và Athero có kiến trúc 32 bit rút gọn (limited). Những thiết bị này thường là các bộ router nhỏ dành cho gia đình và những biến thể khác. Hệ thống thường chạy trên nền tảng nhúng Linux rút gọn hoặc hệ điều hành nhúng dành riêng. Một số trường hợp không sử dụng hệ điều hành như Arduino Zero hoặc Arduino Yun.

Hệ thống IOT lớn nhất là loại sử dụng nền tảng 32 hoặc 64 bit đầy đủ. Những hệ thống như Raspberry Pi hay BeagleBone có thể chạy hệ điều hành Linux đầy đủ hoặc Android. Nhiều trường hợp chính là điện thoại di động hoặc dựa trên công nghệ điện thoại di động. Những thiết bị này có thể đóng vai trò gateway hoặc cầu (bridge) cho các thiết bị nhỏ hơn.

Có một số giải pháp kết nối giữa thiết bị với Internet hoặc gateway:

Kết nối Ethernet hoặc Wi-Fi trực tiếp qua giao thức TCP hoặc UDP.

Bluetooth công suất thấp.

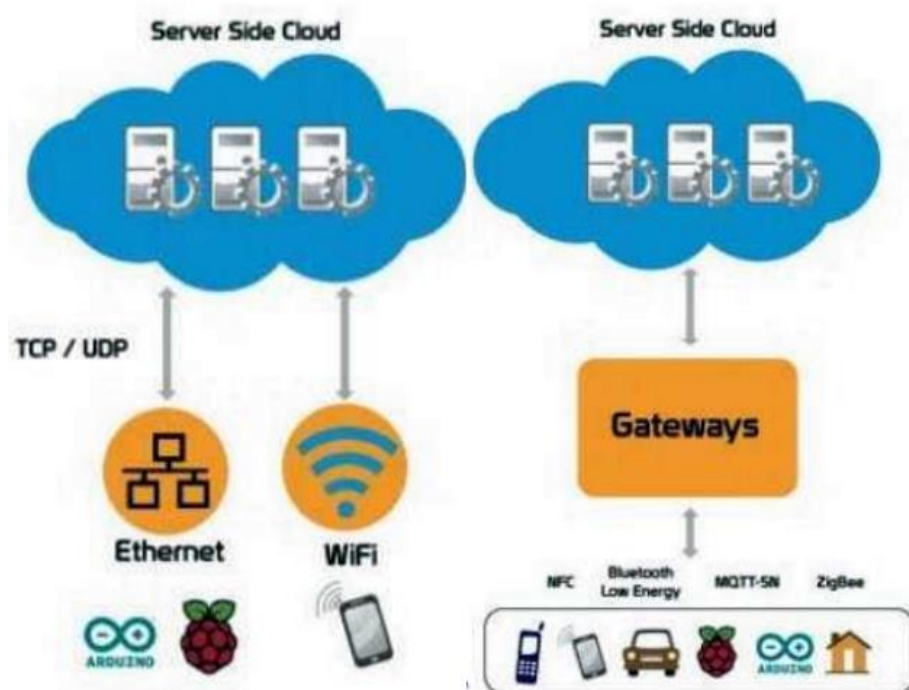
Kết nối trường gần (NFC).

Zigbee hoặc các mạng vô tuyến khác. 38

SRF và kết nối vô tuyến điểm-điểm.

UART hoặc kênh nối tiếp (serial lines).

SPI hoặc kênh I2C (wired buses).



**Hình 2. 13: Hai mô hình kết nối của thiết bị IoT**

### **2.3. Các yêu cầu của kiến trúc tham chiếu cho IoT**

Có thể tóm tắt các yêu cầu của kiến trúc tham chiếu cho IoT trong 5 nhóm:

Kết nối và giao tiếp.

Quản lý thiết bị

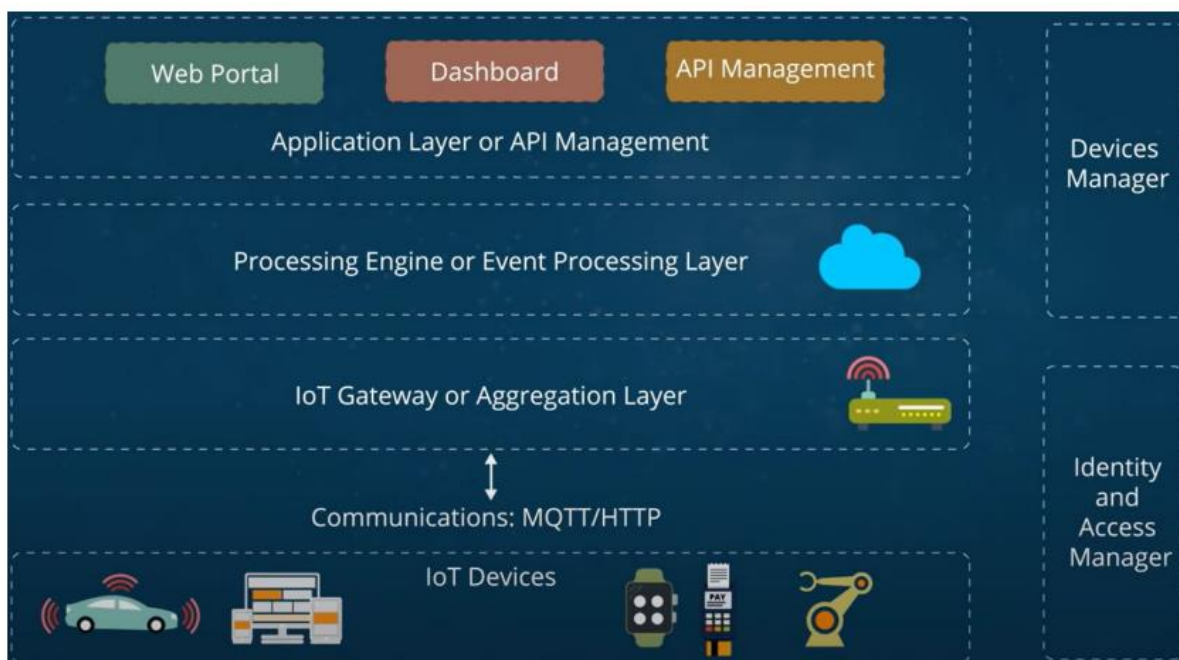
Thu thập, phân tích và khởi động dữ liệu.

Tính khả mở (scalability).

An toàn bảo mật

## 2.4. Mô hình tham chiếu của IoT

Mô hình tham chiếu IoT bao gồm 5 lớp xếp chồng. Mỗi lớp có một chức năng riêng, có thể minh họa bằng những công nghệ cụ thể. Có 2 lớp theo chiều dọc là quản lý thiết bị và quản lý định danh và truy nhập.



**Hình 2. 14: Mô hình tham chiếu của IoT**

### 2.4.1. Lớp thiết bị (Devices)

Các thiết bị IoT phải có giao thức truyền thông trực tiếp (Arduino, Raspberry Pi, Intel Galileo qua Ethernet hoặc Wi-Fi) hoặc gián tiếp kết nối được với Internet (ZigBee, Bluetooth hoặc Bluetooth công suất thấp qua điện thoại di động,...).

Mỗi thiết bị cần có định danh thuộc một trong các loại: định danh duy nhất (UUID) ghi sẵn trong phần cứng (thường là một phần của SoC hoặc chip thứ cấp), UUID gửi qua hệ thống vô tuyến phụ (ví dụ: định danh Bluetooth, địa chỉ Wi-Fi MAC), token OAuth2 Refresh/Bearer (có thể là bổ sung cho các loại khác), định danh lưu trong bộ nhớ chỉ đọc như EEPROM.

### 2.4.2. Lớp truyền thông (Communications)

Lớp truyền thông hỗ trợ kết nối các thiết bị. Nhiều giao thức có thể sử dụng trong lớp này như HTTP/HTTPS, MQTT 3.1/3.1.1, CoAP (Constrained Application Protocol). Trong đó, HTTP là giao thức lâu đời và phổ biến nhất nên có nhiều thư viện hỗ trợ. Vì đó là giao thức dựa

trên ký tự đơn giản nên nhiều thiết bị nhỏ như bộ điều khiển 8 bit đều có thể hỗ trợ HTTP. Các thiết bị 32 bit lớn hơn có thể sử dụng các thư viện HTTP client đầy đủ.

#### **2.4.3. Lớp hợp nhất/Bus (*Aggregation/ Bus*)**

Đây là lớp quan trọng để hợp nhất và chuyển đổi các thông điệp (message broker hay middleware) truyền thông với 3 chức năng sau:

Hỗ trợ máy chủ HTTP và/hoặc chức năng chuyển đổi MQTT để giao tiếp với thiết bị.

Hợp nhất nội dung truyền từ các thiết bị khác nhau và định tuyến truyền thông tới một thiết bị cụ thể (có thể qua gateway).

Bắc cầu và chuyển đổi giữa 2 giao thức khác nhau, ví dụ chuyển đổi API dựa trên HTTP ở lớp trên vào thông điệp MQTT đến thiết bị.

Cuối cùng, lớp hợp nhất/Bus cần thực hiện 2 nhiệm vụ an toàn bảo mật là máy chủ tài nguyên OAuth2 (thẩm định Bearer token và các truy nhập tài nguyên liên quan) và điểm tăng cường chính sách (PEP) đối với truy nhập dựa trên chính sách. Trong mô hình, lớp Bus yêu cầu lớp quản lý truy nhập và định danh thẩm định các yêu cầu truy nhập. Lớp quản lý truy nhập và định danh đóng vai trò như điểm quyết định chính sách (PDP) trong quá trình này. Sau đó, lớp Bus thực hiện theo kết quả do PDP mang đến, nghĩa là cho phép hoặc không cho phép truy nhập tài nguyên

#### **2.4.4. Lớp xử lý sự kiện và phân tích (*Event Processing and Analytics*)**

Lớp này xử lý các sự kiện từ lớp Bus chuyển lên. Yêu cầu chủ yếu ở đây là khả năng lưu trữ dữ liệu vào cơ sở dữ liệu. Mô hình truyền thống sẽ viết một ứng dụng phía máy chủ (ví dụ, JAX-RS). Tuy nhiên, có một số cách tiếp cận khác linh hoạt hơn

#### **2.4.5. Lớp truyền thông ngoài (*External Communication*)**

Lớp nào tạo ra giao diện giúp quản lý các thiết bị IoT như: web/portal, dashboard (bảng hiển thị tổng hợp) hoặc hệ thống quản lý API. Với web/portal, kiến trúc cần hỗ trợ các công nghệ web phía máy chủ như Java Servlets/JSP, PHP, Python, Ruby ...

Web server dựa trên Java phổ biến nhất là Apache Tomcat.

Dashboard là hệ thống tái sử dụng tập trung vào việc trình bày đồ thị mô tả dữ liệu đến từ các thiết bị và lớp xử lý sự kiện.

Lớp quản lý API có 3 chức năng: cung cấp portal tập trung vào hỗ trợ lập trình viên tác nghiệp (chứ không phải là người sử dụng như portal thông thường) và quản lý các phiên bản của API được xuất bản, đóng vai trò gateway quản lý truy nhập vào các API, kiểm tra việc điều khiển truy nhập (đối với yêu cầu từ bên ngoài), điều tiết sử dụng dựa trên chính sách, định tuyến và cân bằng tải. Cuối cùng là chức năng gateway đẩy dữ liệu vào lớp phân tích để lưu trữ và xử lý, giúp hiểu được các API đã được sử dụng như thế nào.

#### ***2.4.6. Lớp quản lý thiết bị (Device Management)***

Trong lớp quản lý thiết bị, hệ thống phía máy chủ DM (Device Manager) giao tiếp với các thiết bị thông qua các giao thức khác nhau và điều khiển phần mềm của từng thiết bị hoặc một nhóm thiết bị (có thể khóa hoặc xóa dữ liệu trên thiết bị khi cần), quản lý định danh các thiết bị và ánh xạ vào chủ nhân các thiết bị đó. DM phải phối hợp với lớp quản lý định danh và truy nhập để quản lý việc điều khiển truy nhập vào thiết bị (những người có quyền truy nhập vào thiết bị ngoài chủ nhân, quyền điều khiển của chủ nhân thiết bị so với người quản trị...).

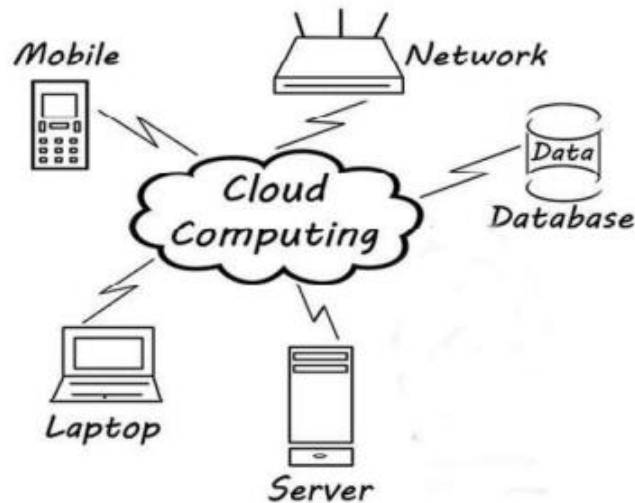
#### ***2.4.7. Lớp quản lý định danh và truy nhập (Identity and Access Management)***

Lớp này cần cung cấp các dịch vụ: Phát hành, thẩm định OAuth2 token; Các dịch vụ định danh khác, gồm SAML2 SSO và OpenID Connect; XACML PDP; Danh bạ cho người dùng (ví dụ: LDAP); Quản lý chính sách điều khiển truy nhập (PCP)



### 3. Giới thiệu về điện toán đám mây

#### 3.1. Giới thiệu chung



**Hình 2. 15: Mô hình của điện toán đám mây**

Điện toán đám mây (cloud computing ) hay còn gọi là điện toán máy chủ ảo là một mô hình điện toán có khả năng co giãn (scalable) linh động và các tài nguyên thường được ảo hóa được cung cấp như một dịch vụ trên mạng Internet. Điện toán đám mây là một kiểu tính toán mà ở đó các công việc công nghệ thông tin được cung cấp như một dịch vụ trên internet đến nhiều khách hàng bên ngoài và khách hàng được tính tiền theo sự sử dụng dịch vụ của họ.

Nói một cách đơn giản nhất “ứng dụng điện toán đám mây” chính là: Một mô hình điện toán mới. Các tài nguyên về hạ tầng (phần cứng, thiết bị lưu trữ, phần mềm hệ thống) và các ứng dụng được cung cấp theo mô hình X-as-a Services dựa theo mô hình trả tiền theo mức độ sử dụng.

#### 3.2. Các đặc điểm

##### 3.2.1. Tự sửa chữa

Bất kỳ ứng dụng hoặc dịch vụ nào đang chạy trong một môi trường điện toán đám mây có một tính chất tự sửa chữa. Trong trường hợp ứng dụng thất bại, luôn luôn có một dự phòng tức thời của ứng dụng sẵn sàng để cho công việc không bị gián đoạn.

### **3.2.2. Nhiều người sử dụng**

Với điện toán đám mây, bất kỳ ứng dụng nào cũng hỗ trợ đa người dùng. Đó là khái niệm dùng để chỉ nhiều người sử dụng đám mây trong cùng thời gian.

### **3.2.3. Khả năng mở rộng tuyến tính**

Dịch vụ điện toán đám mây có khả năng mở rộng tuyến tính. Hệ thống có khả năng phân chia các luồng công việc thành phần nhỏ và phục vụ nó qua cơ sở hạ tầng. Một ý tưởng chính xác của khả năng mở rộng tuyến tính có thể được lấy từ thực tế là nếu một máy chủ có thể xử lý 1000 giao dịch trong một giây, thì hai máy chủ có thể xử lý 2.000 giao dịch trong một giây.

### **3.2.4. Hưởng dịch vụ**

Hệ thống Điện toán đám mây là tất cả các dịch vụ theo định hướng. Những dịch vụ như vậy được tạo ra từ những dịch vụ rời rạc khác. Rất nhiều dịch vụ như vậy là sự kết hợp của nhiều dịch vụ độc lập khác với nhau để tạo dịch vụ này. Điều này cho phép việc tái sử dụng các dịch vụ khác nhau sẵn có và đang được tạo ra. Bằng việc sử dụng các dịch vụ đã được tạo ra trước đó, những dịch vụ khác có thể được tạo ra

### **3.2.5. Điều khiển SLA (Service level agreement)**

Các dịch vụ điện toán đám mây là hướng SLA, như việc khi hệ thống có kinh nghiệm đạt đỉnh của tải, nó sẽ tự động điều chỉnh chính nó để tuân thủ các thỏa thuận ở cấp độ dịch vụ. Các dịch vụ sẽ tạo ra thêm những thực thể của ứng dụng trên nhiều server để cho việc tải có thể dễ dàng quản lý.

### **3.2.6. Khả năng ảo hóa**

Các ứng dụng trong điện toán đám mây hoàn toàn tách rời khỏi phần cứng nằm bên dưới. Môi trường điện toán đám mây là một môi trường ảo hóa đầy đủ.

### **3.2.7. Linh hoạt**

Một tính năng khác của các dịch vụ điện toán đám mây là chúng linh hoạt. Chúng có thể được dùng để phục vụ rất nhiều loại công việc có khối lượng khác nhau từ tải nhỏ của một ứng dụng nhỏ cho đến tải rất nặng của một ứng dụng thương mại.

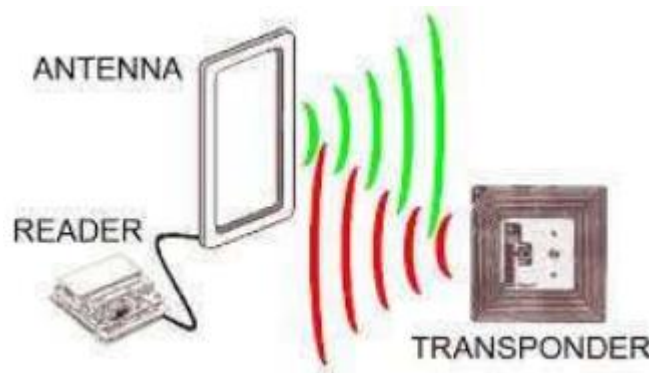
## 4. Các công nghệ truyền thông trong IoT

### 4.1. Radio Frequency Identification (RFID)

Radio Frequency Identification (RFID) là công nghệ nhận dạng đối tượng bằng sóng vô tuyến. Hai thiết bị này hoạt động thu phát sóng điện từ cùng tần số với nhau. Các tần số thường được sử dụng trong hệ thống RFID là 125Khz hoặc 900Mhz.

Một thiết bị hay một hệ thống RFID được cấu tạo bởi hai thành phần chính là thiết bị đọc (reader) và thiết bị phát mã RFID có gắn chip hay còn gọi là tag.

Nguyên lý hoạt động: Thiết bị RFID reader phát ra sóng điện từ ở một tần số nhất định, khi thiết bị RFID tag trong vùng hoạt động sẽ cảm nhận được sóng điện từ này và thu nhận năng lượng từ đó phát lại cho thiết bị RFID Reader biết mã số của mình. Từ đó thiết bị RFID reader nhận biết được tag nào đang trong vùng hoạt động



**Hình 2. 16: Cơ chế hoạt động của thẻ RFID**

### 4.2. Bluetooth

Bluetooth là một tiêu chuẩn công nghệ không dây mở để truyền dữ liệu thiết bị điện tử di động và cố định trong khoảng cách ngắn. Bluetooth được giới thiệu vào năm 1994 như một giải pháp thay thế không dây cho cáp RS-232.

Một số đặc điểm chính của BlueTooth:

Băng tần: 2.4GHz

Tốc độ truyền dữ liệu: 2.1 Mbps hoặc 1 Mbps (Bluetooth Low-Energy)

Khoảng cách ngắn (tối đa 100m trong điều kiện lý tưởng ).

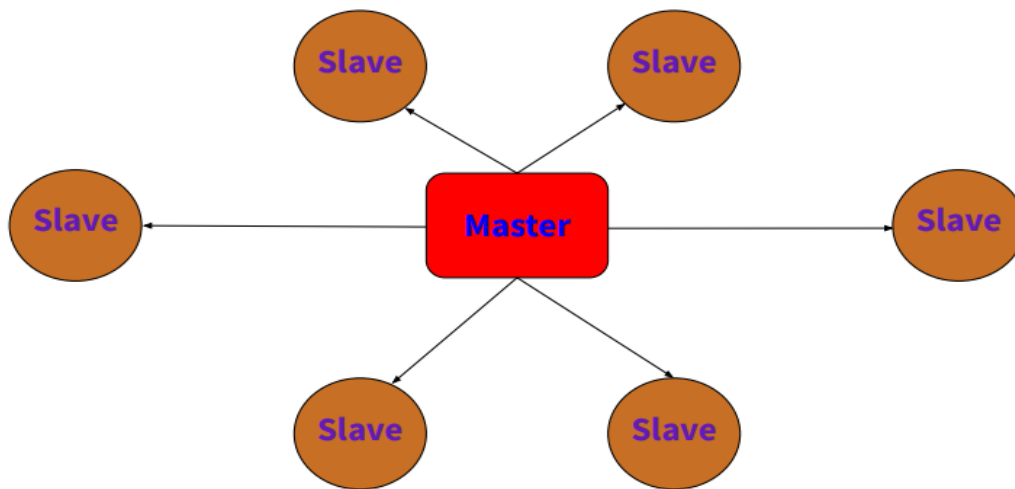
Tiêu thụ năng lượng thấp

Giá thành rẻ

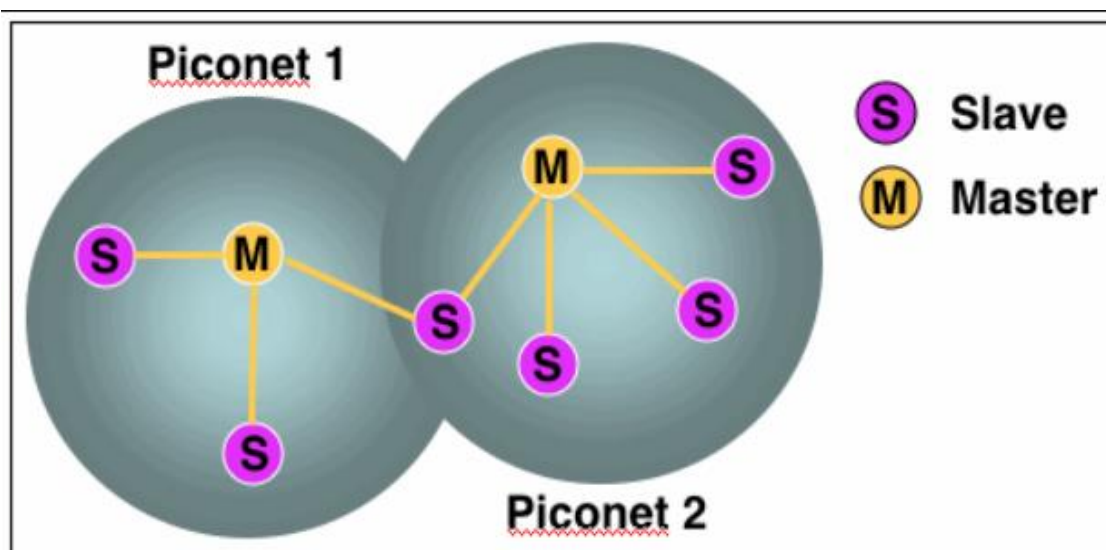
Được ứng dụng rộng rãi trong nhiều thiết bị điện tử ngày nay.

An toàn và bảo mật cao.

### ***Kiến Trúc Mạng Bluetooth***



***Hình 2. 17: Kiến Trúc Mạng Bluetooth Piconet***



***Hình 2. 18: Kiến Trúc Mạng Bluetooth ScatterNet***

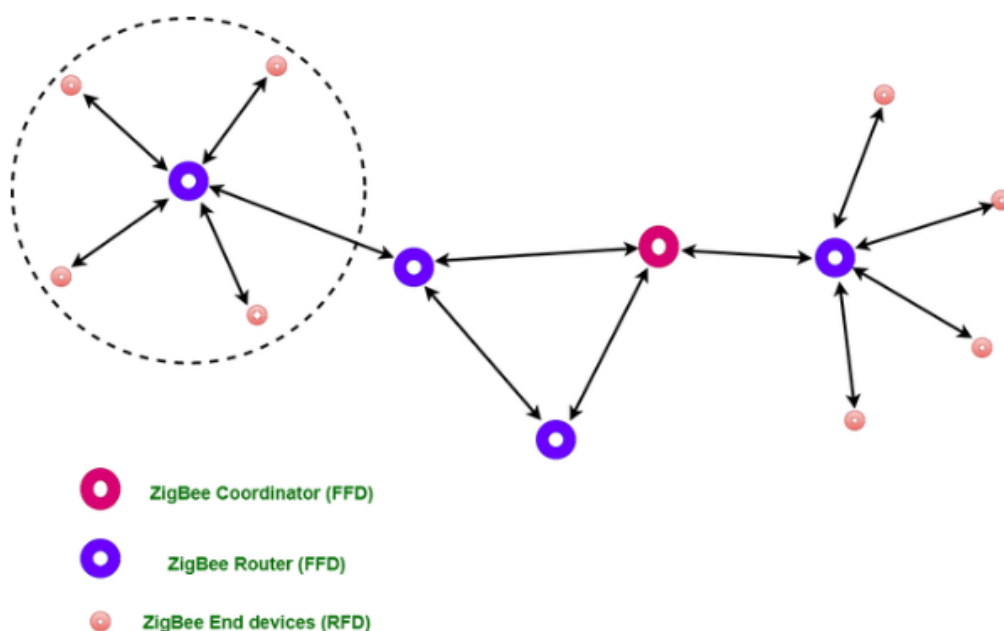
Các thiết bị bluetooth kết nối với nhau thông qua “Bluetooth Profiles”. Bluetooth profiles là một tập hợp các quy tắc và thủ tục được thiết kế để cho phép các thiết bị Bluetooth giao tiếp với nhau cho một mục đích cụ thể.

### 4.3. Zigbee

ZigBee là Mạng khu vực cá nhân thuộc nhóm nhiệm vụ 4 nên nó dựa trên chuẩn IEEE 802.15.4 và được tạo ra bởi Zigbee Alliance.

ZigBee là một giao thức mở, toàn cầu, dựa trên gói được thiết kế để cung cấp kiến trúc dễ sử dụng cho các mạng không dây an toàn, đáng tin cậy, tiêu thụ điện năng thấp.

Đặc tả ZigBee hỗ trợ cấu trúc hình sao và hai loại cấu trúc liên kết ngang hàng, lưới và cây cụm. Các thiết bị tương thích với ZigBee đôi khi được chỉ định là hỗ trợ các cấu trúc liên kết điểm-điểm và điểm-đa điểm. Mỗi dạng hình đều có những ưu điểm riêng và được ứng dụng trong các trường hợp khác nhau.



**Hình 2. 19: Mô hình mạng Zigbee**

Các đặc điểm chính của ZigBee

- Stochastic Addressing: địa chỉ được thông báo và chỉ định ngẫu nhiên. Có cơ chế chống xung đột địa chỉ. Node cha không cần lưu bảng địa chỉ. ZigBee sử dụng 16bit địa chỉ.
- Link Management: Mỗi node tự duy trì liên kết với các node lân cận. Chất lượng kết nối được sử dụng để đo lường mất mát trong định tuyến.
- Frequency Agility: Khi gặp can nhiễu, node mạng có thể thay đổi kênh truyền.

- Asymmetric Link: Mỗi node có công suất truyền và độ nhạy khác nhau, do đó đường liên kết có thể bất đối xứng.
- Power Management: Gateway/Coordinator và router dùng nguồn cố định. Trong khi thiết bị cuối sử dụng Pin.

Các điểm hạn chế của ZigBee:

- Phạm vi ngắn
- Tốc độ thấp
- Interoperability
- Security

Cấu trúc của Zigbee: Ngoài 2 tầng vật lý và tầng MAC xác định bởi tiêu chuẩn 802.15.4 ở, tiêu chuẩn ZigBee còn có thêm các tầng trên của hệ thống bao gồm: tầng mạng, tầng hỗ trợ ứng dụng, tầng đối tượng thiết bị và các đối tượng ứng dụng

#### **4.4. Wifi**

- Wi-Fi viết tắt từ Wireless Fidelity hay mạng 802.21 là hệ thống mạng không dây sử dụng sóng vô tuyến, giống như điện thoại di động, truyền hình và radio. Hệ thống này đã hoạt động ở một số sân bay, quán cafe, thư viện hoặc khách sạn. Hệ thống cho phép truy cập Internet tại những khu vực có sóng của hệ thống này, hoàn toàn không cần đến cáp nối. Ngoài các điểm kết nối công cộng (hotspots), WiFi có thể được thiết lập ngay tại nhà riêng.

- Tên gọi 802.21 bắt nguồn từ viện IEEE (Institute of Electrical and Electronics Engineers). Viện này tạo ra nhiều chuẩn cho nhiều giao thức kỹ thuật khác nhau, và nó sử dụng một hệ thống số nhằm phân loại chúng; 4 chuẩn thông dụng của WiFi hiện nay là 802.21a/b/g/n.

- Truyền thông qua mạng không dây là truyền thông vô tuyến hai chiều. Cụ thể: Thiết bị adapter không dây (hay bộ chuyển tín hiệu không dây) của máy tính chuyển đổi dữ liệu sang tín hiệu vô tuyến và phát những tín hiệu này đi bằng một ăng-ten. Thiết bị router không dây nhận những tín hiệu này và giải mã chúng. Nó gửi thông tin

tới Internet thông qua kết nối hữu tuyến Ethernet. Quy trình này vẫn hoạt động với chiều ngược lại, router nhận thông tin từ Internet, chuyển chúng thành tín hiệu vô tuyến và gửi đến adapter không dây của máy tính.

- Các sóng vô tuyến sử dụng cho WiFi gần giống với các sóng vô tuyến sử dụng cho thiết bị cầm tay, điện thoại di động và các thiết bị khác. Nó có thể chuyển và nhận sóng vô tuyến, chuyển đổi các mã nhị phân 1 và 0 sang sóng vô tuyến và ngược lại.

Tuy nhiên, sóng WiFi có một số khác biệt so với các sóng vô tuyến khác ở chỗ:

- Chúng truyền và phát tín hiệu ở tần số 2.4 GHz hoặc 5 GHz. Tần số này cao hơn so với các tần số sử dụng cho điện thoại di động, các thiết bị cầm tay và truyền hình.

Tần số cao hơn cho phép tín hiệu mang theo nhiều dữ liệu hơn.

- Chúng dùng chuẩn 802.11:

- Chuẩn 802.11b là phiên bản đầu tiên trên thị trường. Đây là chuẩn chậm nhất và rẻ tiền nhất, và nó trở nên ít phổ biến hơn so với các chuẩn khác. 802.11b phát tín hiệu ở tần số 2.4 GHz, nó có thể xử lý đến 11 megabit/giây, và nó sử dụng mã CCK (complementary code keying).

- Chuẩn 802.11g cũng phát ở tần số 2.4 GHz, nhưng nhanh hơn so với chuẩn 802.11b, tốc độ xử lý đạt 54 megabit/giây. Chuẩn 802.11g nhanh hơn vì nó sử dụng mã OFDM (orthogonal frequency-division multiplexing), một công nghệ mã hóa hiệu quả hơn.

- Chuẩn 802.11a phát ở tần số 5GHz và có thể đạt đến 54 megabit/ giây. Nó cũng sử dụng mã OFDM. Những chuẩn mới hơn sau này như 802.11n còn nhanh hơn chuẩn 802.11a, nhưng 802.11n vẫn chưa phải là chuẩn cuối cùng.

- Chuẩn 802.11n cũng phát ở tần số 2.4 GHz, nhưng nhanh hơn so với chuẩn

802.11a, tốc độ xử lý đạt 300 megabit/giây.

- Chuẩn 802.11ac phát ở tần số 5 GHz
- WiFi có thể hoạt động trên cả ba tần số và có thể nhảy qua lại giữa các tần số khác nhau một cách nhanh chóng. Việc nhảy qua lại giữa các tần số giúp giảm thiểu sự nhiễu sóng và cho phép nhiều thiết bị kết nối không dây cùng một lúc.

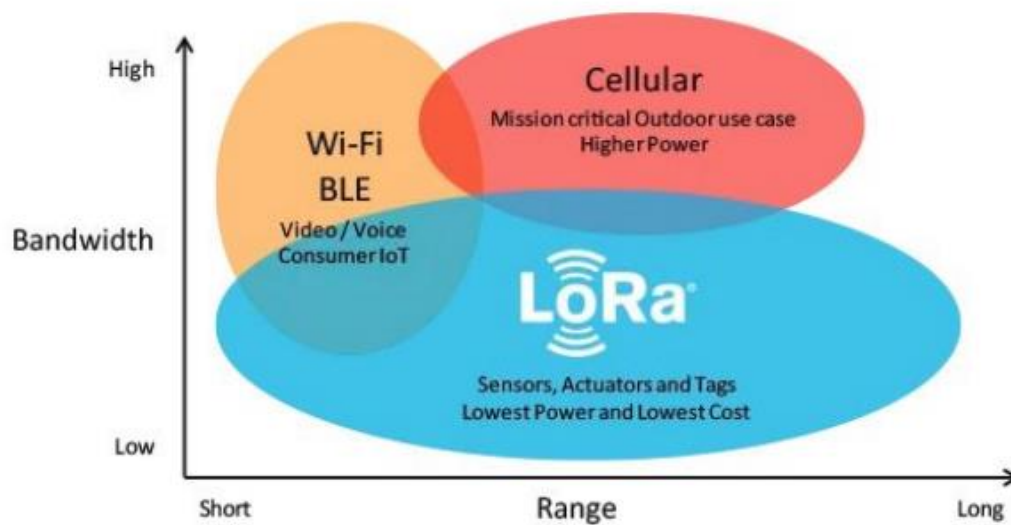
#### **4.5. Lora LoraWAN**

LoRa là một kỹ thuật điều chế không dây bắt nguồn từ công nghệ Chirp Spread Spectrum (CSS). Nó mã hóa thông tin trên sóng vô tuyến bằng các xung chirp tương tự như cách cá heo và dơi giao tiếp! LoRa có khả năng chống nhiễu tốt và có thể thu được ở khoảng cách rất xa.

LoRa có thể được vận hành trên các băng tần dưới gigahertz không có giấy phép, ví dụ: 915 MHz, 868 MHz và 433 MHz. Nó cũng có thể hoạt động trên tần số 2,4 GHz để đạt được tốc độ dữ liệu cao hơn so với các băng tần dưới gigahertz, với chi phí về phạm vi. Các tần số này rơi vào các băng tần ISM được dành riêng trên phạm vi quốc tế cho các mục đích công nghiệp, khoa học và y tế.

LoRaWAN là giao thức lớp Kiểm soát truy cập phương tiện (MAC) được xây dựng dựa trên điều chế LoRa. Đây là lớp phần mềm xác định cách các thiết bị sử dụng phần cứng LoRa, ví dụ như khi chúng truyền và định dạng của tin nhắn

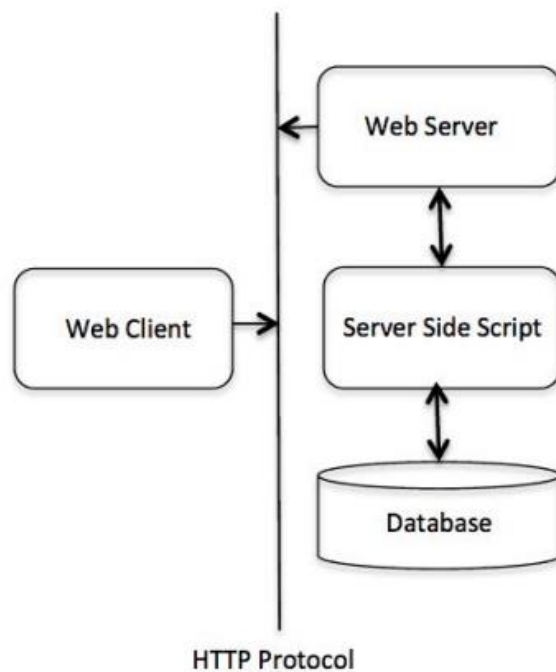




*Hình 2. 20: Một số công nghệ truy cập có thể được sử dụng để truyền dữ liệu không dây và phạm vi truyền dự kiến của chúng so với băng thông*

## 5. Các Giao Thức Truyền Tin

### 5.1 HTTP

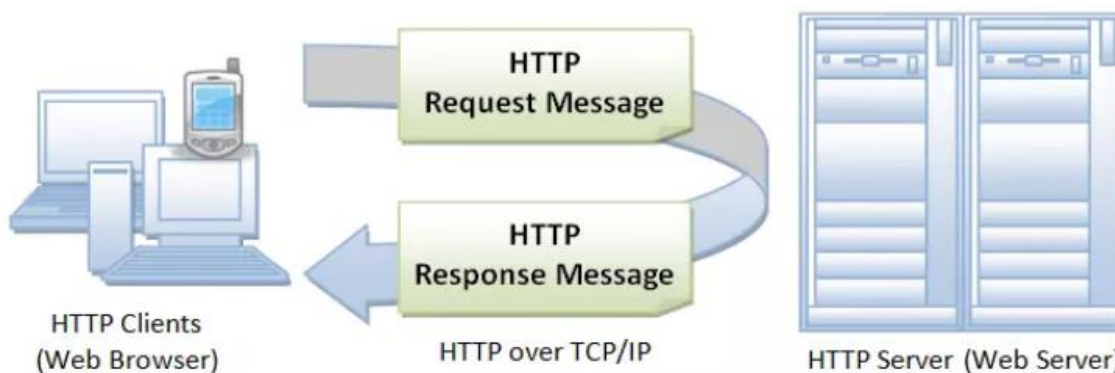


*Hình 2. 21: Cấu trúc cơ bản của 1 ứng dụng web*

HTTP còn là 1 giao thức **Yêu cầu – Phản hồi** dựa trên cấu trúc Client – Server. Client và Server giao tiếp với nhau bằng cách trao đổi các message độc lập (trái ngược với 1 luồng dữ liệu). Các message được gửi bởi client, thông thường là 1 trình duyệt web, được gọi là các **yêu cầu** và message được gửi bởi server như 1 sự trả lời, được gọi là **phản hồi**.

HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho Internet). Bộ giao thức TCP/IP là một bộ các giao thức truyền thông cài đặt chồng giao thức mà Internet và hầu hết các mạng máy tính thương mại đang chạy trên đó. Bộ giao thức này được đặt theo tên hai giao thức chính là TCP (Transmission Control Protocol – Giao thức điều khiển truyền vận) và IP (Internet Protocol – Giao thức Internet).

HTTP là một giao thức chung và không trạng thái có thể được sử dụng cho các mục đích khác cũng như sử dụng các phần mở rộng của phương thức yêu cầu, mã lỗi và tiêu đề của nó.



**Hình 2. 22: Sơ đồ hoạt động của HTTP**

HTTP hoạt động dựa trên mô hình Client – Server. Trong mô hình này, các máy tính của người dùng sẽ đóng vai trò làm máy khách (Client). Sau một thao tác nào đó của người dùng, các máy khách sẽ gửi yêu cầu đến máy chủ (Server) và chờ đợi câu trả lời từ những máy chủ này.

Các thành phần chính của HTTP:

- **HTTP -Requests** : Là phương thức để chỉ ra hành động mong muốn được thực hiện trên tài nguyên đã xác định.
- **Cấu trúc của một HTTP Request:**
  - Một Request-line = Phương thức + URI-Request + Phiên bản HTTP

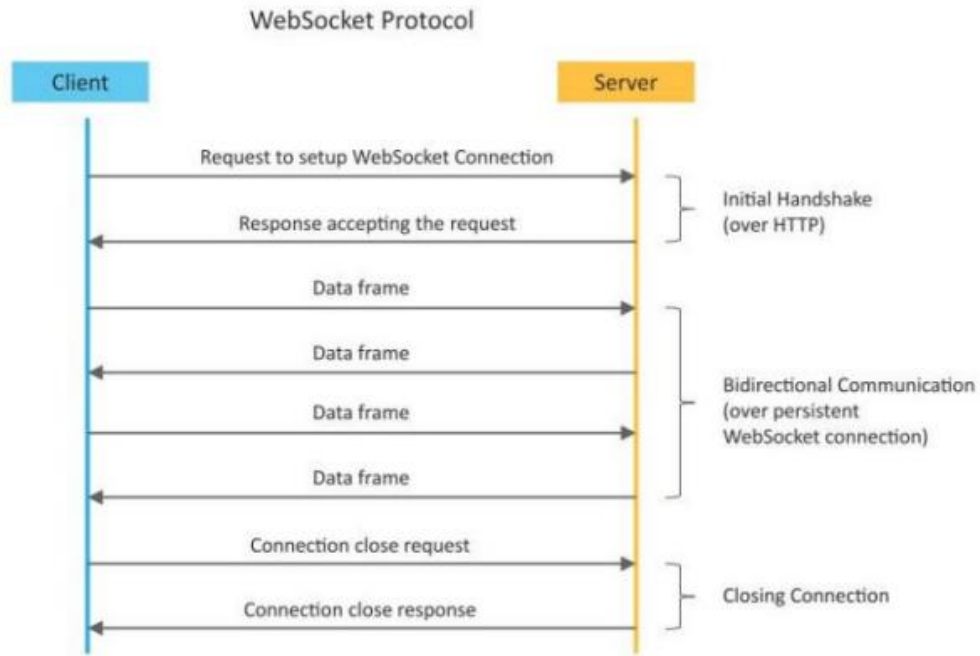
- Có thể có hoặc không các trường header
- Một dòng trống để đánh dấu sự kết thúc của các trường Header.
- Tùy chọn một thông điệp

Method	Hoạt động	Chú thích
GET	được sử dụng để lấy lại thông tin từ Server một tài nguyên xác định.	Các yêu cầu sử dụng GET chỉ nên nhận dữ liệu và không nên có ảnh hưởng gì tới dữ liệu
POST	yêu cầu máy chủ chấp nhận thực thể được đính kèm trong request được xác định bởi URI, ví dụ, thông tin khách hàng, file tải lên, ...	
PUT	Nếu URI đề cập đến một tài nguyên đã có, nó sẽ bị sửa đổi; nếu URI không trỏ đến một tài nguyên hiện có, thì máy chủ có thể tạo ra tài nguyên với URI đó.	
DELETE	Xóa bỏ tất cả các đại diện của tài nguyên được chỉ định bởi URI.	
PATCH	Áp dụng cho việc sửa đổi một phần của tài nguyên được xác định.	
...	...	...

**Hình 2. 23: Một số HTTP Request thường dùng**

- **HTTP-Response:** Cấu trúc của một HTTP response:
  - Một Status-line = Phiên bản HTTP + Mã trạng thái + Trạng thái
  - Có thể có hoặc không có các trường header
  - Một dòng trống để đánh dấu sự kết thúc của các trường header
  - Tùy chọn một thông điệp

## 5.2 WebSocket



**Hình 2. 24: Quy trình hoạt động của Websocket**

WebSockets là một giao thức truyền thông tạo điều kiện cho việc truyền thông liên tục, hai chiều, thời gian thực giữa máy khách và máy chủ.

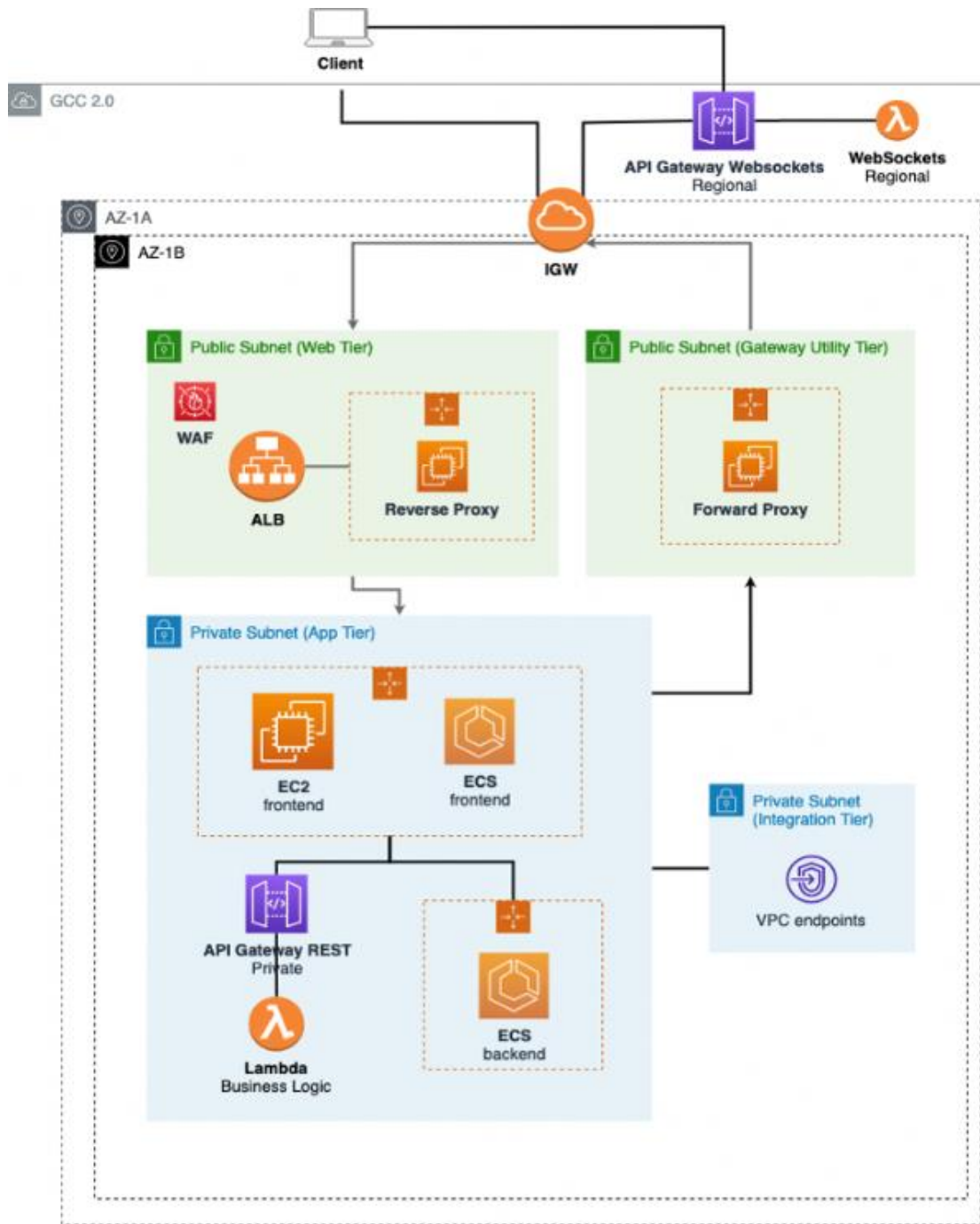
Không giống như HTTP, hoạt động trên cơ sở yêu cầu-phản hồi, WebSockets cho phép máy chủ gửi dữ liệu đến máy khách mà không cần máy khách phải khởi tạo yêu cầu. Điều này cho phép Websocket trở thành lựa chọn đặc biệt phù hợp cho các ứng dụng thời gian thực như hệ thống trò chuyện, trò chơi nhiều người chơi và máy đánh dấu tài chính. Quan trọng hơn, thời gian chờ không còn là vấn đề đáng lo ngại nữa vì kết nối sẽ vẫn hoạt động cho đến khi máy khách hoặc máy chủ chấm dứt kết nối. Hình 16 giới thiệu cơ chế hoạt động của WebSocket.

WebSocket hoạt động theo các quy trình sau:

- Thiết lập kết nối và nâng cấp lên giao thức WebSocket
- Gửi và nhận dữ liệu
- Đóng kết nối

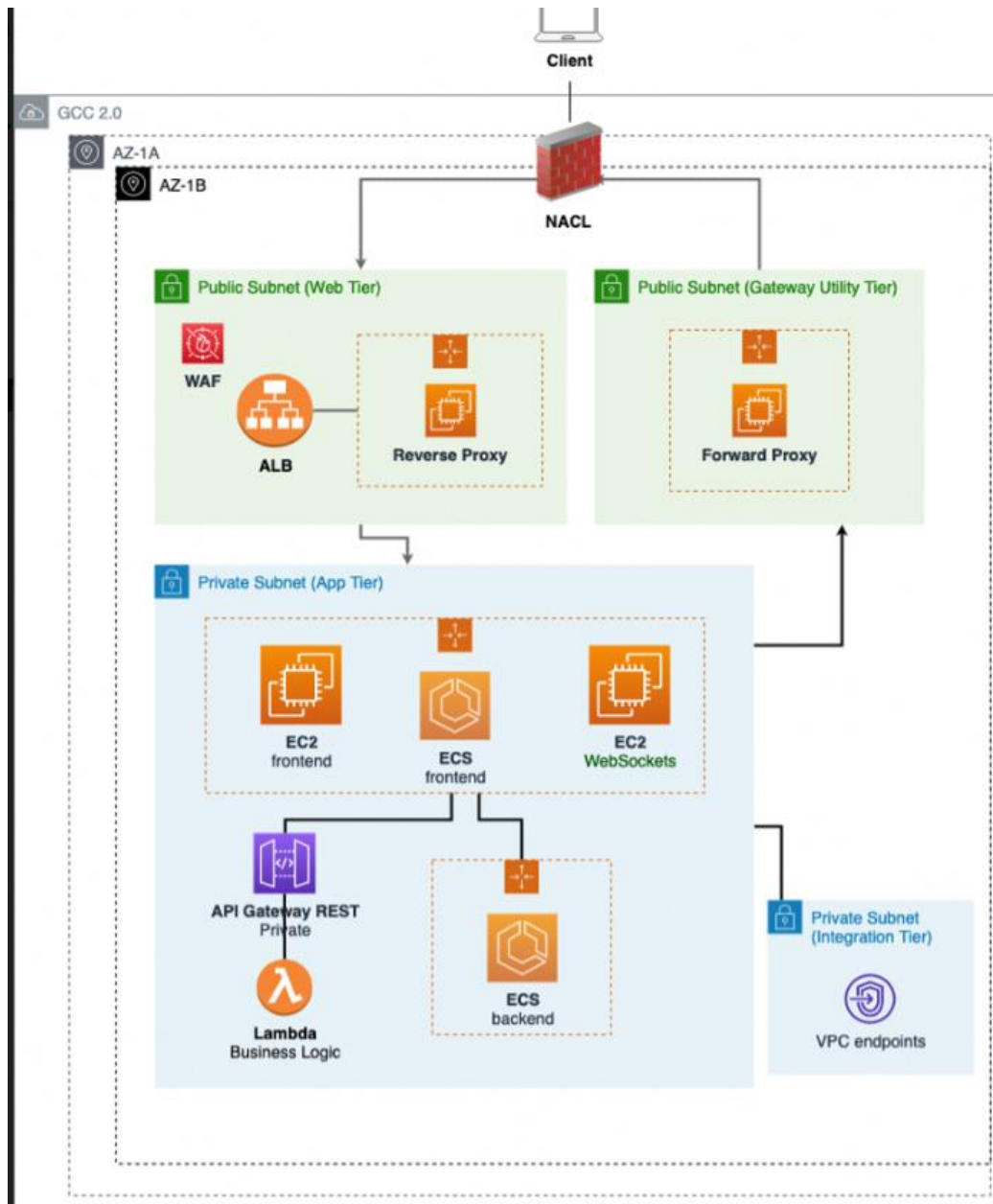
Sơ đồ kiến trúc:

- Triển khai dựa trên internet (AWS):



**Hình 2. 25: Sơ đồ kiến trúc WebSocket dựa trên internet**

- Triển khai dựa trên mạng nội bộ (AWS):



**Hình 2. 26: Sơ đồ kiến trúc WebSocket dựa trên mạng nội bộ**

### 5.3 MQTT

MQTT là giao thức nhắn tin đăng ký/xuất bản nhẹ được thiết kế cho mục đích đo từ xa M2M (máy với máy) trong môi trường băng thông thấp.

MQTT ban đầu được thiết kế vào năm 1999 và đã được sử dụng trong nhiều năm và được thiết kế cho mạng TCP/IP .

MQTT đang nhanh chóng trở thành một trong những giao thức chính cho việc triển khai IOT (internet vạn vật).

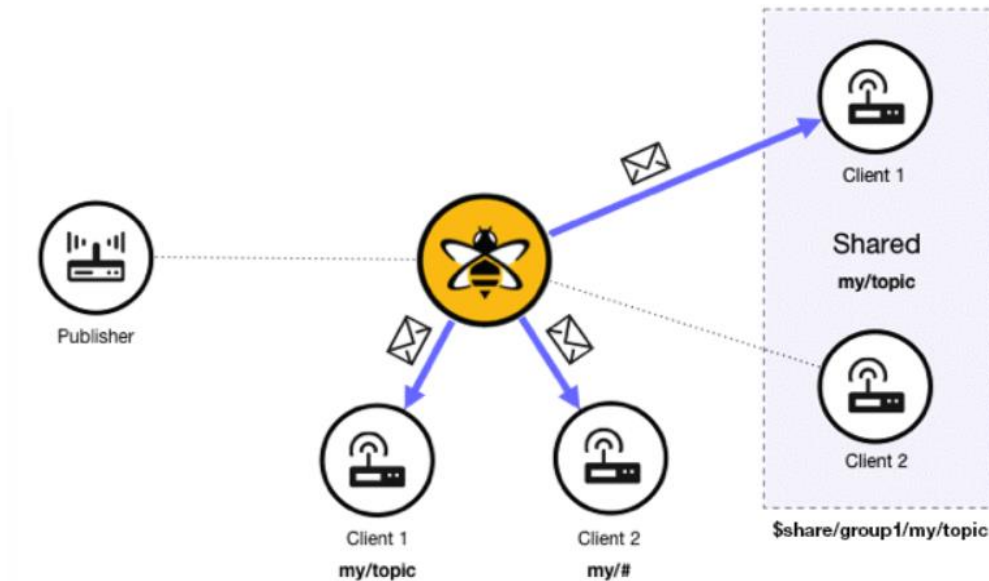
***MQTT cung cấp một số lợi ích chính:***

- Nhẹ và hiệu quả: MQTT giảm thiểu tài nguyên mà khách hàng và băng thông mạng yêu cầu.
- Giao tiếp hai chiều: MQTT tạo điều kiện giao tiếp giữa các thiết bị và máy chủ, hỗ trợ xuất bản và đăng ký. Nó cũng cho phép truyền phát tin nhắn tới các nhóm thiết bị.
- Khả năng mở rộng: MQTT có thể mở rộng quy mô để hỗ trợ hàng triệu thiết bị hoặc “mọi thứ” trong hệ sinh thái IoT hoặc IIoT.
- Mức chất lượng dịch vụ (QoS): MQTT chỉ định các mức QoS khác nhau để đảm bảo gửi tin nhắn đáng tin cậy.
- Phiên liên tục: MQTT hỗ trợ phiên liên tục giữa thiết bị và máy chủ, giảm thời gian kết nối lại qua các mạng không đáng tin cậy.

Thành phần:

- Thành phần chính của MQTT là Client (Publisher/Subscriber), Server (Broker), Sessions (tạm dịch là Phiên làm việc), Subscriptions và Topics.
- MQTT Client (Publisher/Subscriber): Clients sẽ subscribe một hoặc nhiều topics để gửi và nhận thông điệp từ những topic tương ứng.
- MQTT Server (Broker): Broker nhận những thông tin subscribe (Subscriptions) từ client, nhận thông điệp, chuyển những thông điệp đến các Subscriber tương ứng dựa trên Subscriptions từ client.
- Topic: Có thể coi Topic là một hàng đợi các thông điệp, và có sẵn khuôn mẫu dành cho Subscriber hoặc Publisher. Một cách logic thì các topic cho phép Client trao đổi thông tin với những ngữ nghĩa đã được định nghĩa sẵn. Ví dụ: Dữ liệu cảm biến nhiệt độ của một tòa nhà.
- Session: Một session được định nghĩa là kết nối từ client đến server. Tất cả các giao tiếp giữa client và server đều là 1 phần của session.
- Subscription: Không giống như session, subscription về mặt logic là kết nối từ client đến topic. Khi đã subscribe một topic, Client có thể nhận/gửi thông điệp (message) với topic đó.





**Hình 2. 27: Mô hình Publish/Subscribe của MQTT**

Client thì được chia thành hai nhóm là Publisher và Subscriber. Client chỉ làm ít nhất một trong 2 việc là publish các thông điệp (message) lên một/nhiều topic cụ thể hoặc subscribe một/nhiều topic nào đó để nhận message từ topic này.

- Các Clients không có địa chỉ và Clients không trực tiếp gửi message cho nhau.
- Message được truyền đi thông qua Broker
- Broker sẽ phân phối Message tới clients thông qua các topics
- Giữa các clients không có kết nối trực tiếp
- Client có thể vừa publish vừa subscribe

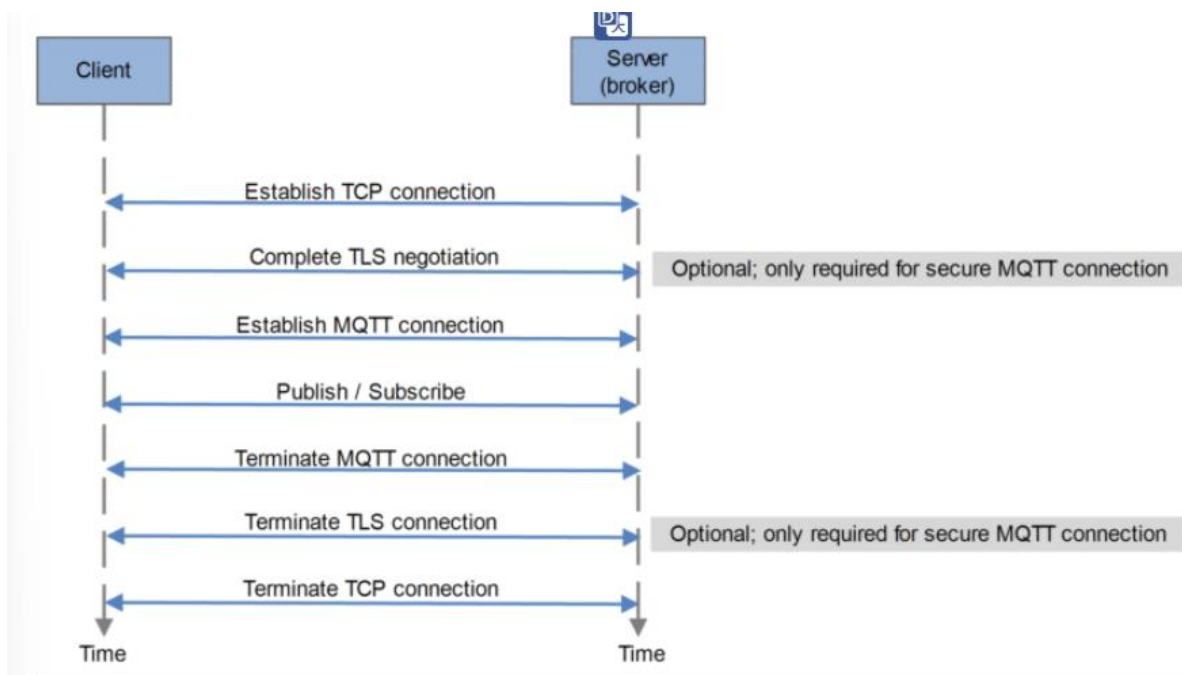
Cơ chế hoạt động của giao thức MQTT

Một phiên MQTT được chia thành bốn giai đoạn: kết nối, xác thực, giao tiếp và kết thúc.

- Client (máy khách) bắt đầu bằng cách tạo kết nối Transmission Control Protocol/Internet Protocol (TCP/IP) tới broker bằng cách sử dụng cổng tiêu chuẩn hoặc cổng tùy chỉnh được xác định bởi các nhà phát triển broker.



- Các cổng tiêu chuẩn là 1883 cho giao tiếp không mã hóa và 8883 cho giao tiếp được mã hóa – sử dụng Lớp cổng bảo mật (SSL) / Bảo mật lớp truyền tải (TLS). Trong quá trình giao tiếp SSL/TLS, máy khách cần kiểm chứng và xác thực máy chủ.
- Sau đó, Client sẽ gửi bản tin lên broker nếu là Publisher hoặc nhận bản tin từ broker về nếu là Subscriber. Quá trình kết nối này sẽ được giữ đến khi Kết thúc kết nối.
- Sau khi kết thúc để có thể truyền nhận MQTT, chúng ta lại tiếp tục quay lại các bước trên.



**Hình 2. 28: Cơ chế hoạt động của MQTT**

### Tồn tại ba mức độ tin cậy cho việc truyền dữ liệu (QoS: Quality of service)

- QoS0: broker/client sẽ gửi dữ liệu 1 lần. Không cần xác nhận.
  - QoS1: broker/client sẽ gửi dữ liệu với ít nhất 1 lần xác nhận.
  - QoS2: broker/client đảm bảo khi dữ liệu được gửi phía nhận chỉ nhận được đúng 1 lần.
- Quá trình này yêu cầu phải trải qua 4 bước bắt tay.

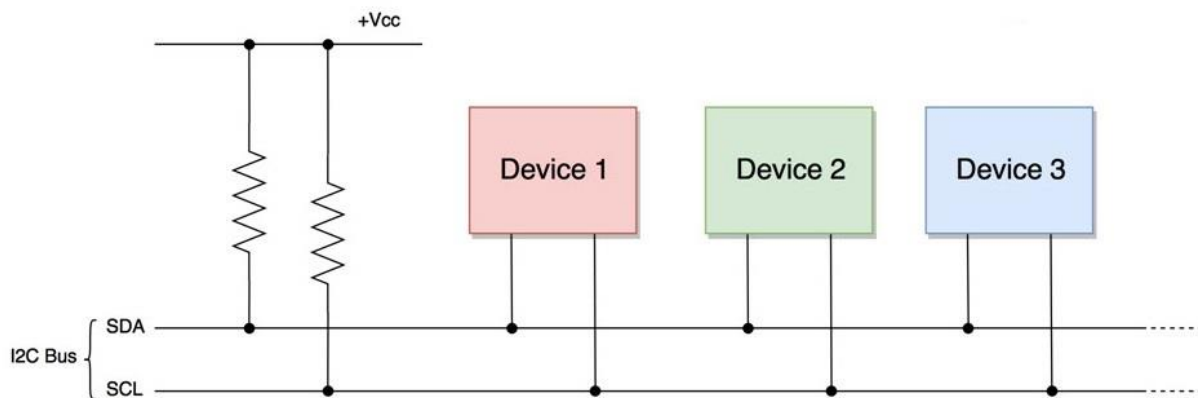
Một gói tin có thể được gửi ở bất kỳ mức QoS nào, và client cũng có thể subscribe ở bất kỳ QoS nào

## 6. Giao tiếp ngoại vi

### 6.1 Giao tiếp I2C

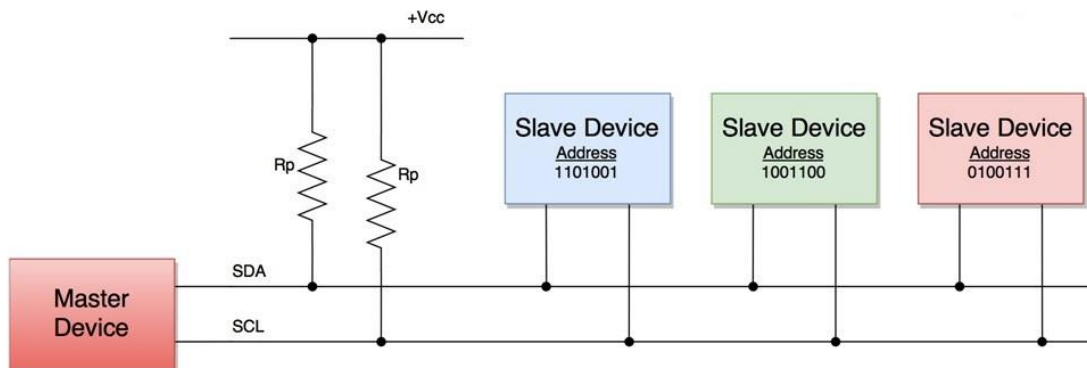
Đặc điểm chính của I2C

- **Số dây kết nối:** I2C chỉ cần hai dây để truyền dữ liệu:
  - **SDA (Serial Data Line):** Đường truyền dữ liệu.
  - **SCL (Serial Clock Line):** Đường truyền tín hiệu đồng hồ.



*Hình 2. 29: Bus vật lý I2C*

- **Địa chỉ thiết bị:** Mỗi thiết bị trên bus I2C có một địa chỉ duy nhất, thường là 7-bit hoặc 10-bit.
- **Chế độ truyền dữ liệu:** I2C hỗ trợ nhiều chế độ truyền dữ liệu với tốc độ khác nhau:
  - **Standard Mode:** 100 kbps
  - **Fast Mode:** 400 kbps
  - **Fast Mode Plus:** 1 Mbps
  - **High-Speed Mode:** 3.4 Mbps
- **Master và Slave:** Trong mạng I2C, có thể có nhiều thiết bị Master và Slave. Master điều khiển tín hiệu đồng hồ và khởi tạo truyền dữ liệu, trong khi Slave đáp ứng các yêu cầu từ Master.



**Hình 2. 30: Bus vật lý I2C**

Cách hoạt động của I2C

- **Khởi tạo truyền dữ liệu:** Master tạo tín hiệu bắt đầu bằng cách kéo SDA từ mức cao xuống mức thấp trong khi SCL vẫn ở mức cao.
- **Địa chỉ thiết bị:** Master gửi địa chỉ của thiết bị Slave mà nó muốn giao tiếp.
- **Bit đọc/ghi:** Master gửi một bit để xác định xem nó muốn đọc từ hay ghi vào Slave.
- **Truyền dữ liệu:** Dữ liệu được truyền từng byte một, với mỗi byte được xác nhận bằng một bit ACK/NACK từ Slave.
- **Kết thúc truyền dữ liệu:** Master tạo tín hiệu kết thúc bằng cách kéo SDA từ mức thấp lên mức cao trong khi SCL vẫn ở mức cao.

Ưu điểm của I2C

- **Đơn giản và tiết kiệm dây:** Chỉ cần hai dây để kết nối nhiều thiết bị.
- **Địa chỉ hóa linh hoạt:** Hỗ trợ nhiều thiết bị trên cùng một bus.
- **Đồng bộ hóa:** Dữ liệu được truyền đồng bộ với tín hiệu đồng hồ, giúp giảm thiểu lỗi truyền dữ liệu.

Nhược điểm của I2C

- **Tốc độ truyền dữ liệu hạn chế:** So với các giao thức khác như SPI, tốc độ truyền dữ liệu của I2C thấp hơn.
- **Độ phức tạp trong thiết kế phần cứng:** Cần sử dụng điện trở kéo lên cho các đường SDA và SCL.

## CHƯƠNG 3. PHÂN TÍCH YÊU CẦU BÀI TOÁN

### 1. Requirement

#### *1.1 Mô tả bằng ngôn ngữ tự nhiên*

##### *1.1.1 Giới thiệu mục đích hệ thống*

Hệ thống quản lý cảm biến được thiết kế để giám sát và điều khiển các trạm cảm biến từ xa. Hệ thống này cho phép người dùng theo dõi nhiệt độ, độ ẩm và các thông số môi trường khác. Ngoài ra, hệ thống cũng cho phép điều khiển các thiết bị như quạt, đèn LED và còi báo động khi các thông số vượt quá ngưỡng cho phép. Mục tiêu chính là đảm bảo môi trường được kiểm soát một cách hiệu quả và kịp thời.

##### *1.1.2 Phạm vi hệ thống*

- Thành viên hệ thống:
  - Đăng nhập
  - Đăng xuất
- Người dùng cấp 1 (Quản trị viên):
  - Có quyền truy cập toàn bộ hệ thống.
  - Quản lý người dùng (tạo, sửa, xóa người dùng cấp 2).
  - Quản lý trạm (tạo, sửa, xóa trạm).
  - Thiết lập và điều chỉnh các ngưỡng cảm biến.
  - Xem và điều khiển tất cả các trạm và cảm biến.
- Người dùng cấp 2 (Người vận hành trạm):
  - Chỉ được truy cập các trạm được phân công.
  - Xem thông tin và trạng thái của trạm và cảm biến.
  - Điều khiển các thiết bị trong trạm của họ (quạt, đèn LED, còi báo động).
  - Không có quyền thay đổi cấu hình hệ thống hoặc quản lý người dùng khác.

##### *1.1.3 Mô tả chi tiết hoạt động nghiệp vụ*

- Người dùng đăng nhập vào hệ thống: Người dùng (cấp 1 hoặc cấp 2) sử dụng thông tin đăng nhập để truy cập vào hệ thống quản lý cảm biến. Hệ thống kiểm tra thông tin đăng nhập và xác thực quyền truy cập.
- Chọn trạm quản lý (dành cho người dùng cấp 1) : Sau khi đăng nhập, người dùng cấp 1 sẽ được hiển thị danh sách các trạm quản lý. Họ có thể chọn một trạm cụ thể để theo dõi và điều khiển các cảm biến trong trạm đó.

- **Hiện thị danh sách trạm và cảm biến (dành cho người dùng cấp 2) :** Người dùng cấp 2 chỉ thấy danh sách các trạm mà họ được phân công quản lý. Mỗi trạm sẽ hiển thị các cảm biến liên quan, bao gồm tên, loại cảm biến, trạng thái hiện tại (OK, lỗi), giá trị đo lường (nếu có), và thời gian đo lần cuối.
- **Chọn cảm biến và điều khiển thiết bị :** Người dùng có thể chọn từng cảm biến trong trạm để thực hiện các hành động:
  - **Xem chi tiết cảm biến:** Hệ thống hiển thị thông tin chi tiết về cảm biến, bao gồm loại cảm biến (nhiệt độ, độ ẩm, quạt, LED, còi báo động), giá trị đo lường hiện tại, và trạng thái hoạt động.
  - **Điều khiển thiết bị:** Nếu cảm biến liên quan đến thiết bị điều khiển (ví dụ, quạt, đèn LED), người dùng có thể bật/tắt thiết bị hoặc điều chỉnh tốc độ (quạt).
- **Kiểm tra trạng thái thiết bị và cảm biến :** Hệ thống tự động kiểm tra trạng thái của từng cảm biến và thiết bị, sau đó hiển thị các thông báo lỗi nếu có cảm biến gặp sự cố. Đối với cảm biến DHT (nhiệt độ và độ ẩm), hệ thống sẽ liên tục giám sát các giá trị đo đạc và so sánh với ngưỡng đã đặt. Nếu vượt quá ngưỡng, hệ thống sẽ kích hoạt quạt, LED hoặc còi báo động.
- **Điều khiển trạm theo thời gian thực (dành cho người dùng cấp 2) :** Người dùng cấp 2 có thể điều khiển các thiết bị trong trạm của họ theo thời gian thực thông qua giao diện điều khiển. Các thay đổi sẽ được thực hiện ngay lập tức và phản hồi trạng thái của thiết bị sẽ được hiển thị trên giao diện người dùng.
- **Quay lại trang chính :** Sau khi hoàn tất điều khiển hoặc kiểm tra cảm biến, người dùng có thể quay lại giao diện chính, nơi họ có thể theo dõi tổng quan tất cả các trạm và cảm biến được phân công (đối với người dùng cấp 2) hoặc toàn bộ hệ thống (đối với người dùng cấp 1).
- **Lưu thay đổi và cập nhật hệ thống :** Sau khi thực hiện bất kỳ thay đổi nào trong trạng thái thiết bị (bật/tắt quạt, đèn LED, điều chỉnh tốc độ quạt), người dùng có thể nhấn nút "Update" để chính thức cập nhật các thay đổi lên hệ thống. Thông tin mới sẽ được lưu vào cơ sở dữ liệu, và người dùng sẽ quay về giao diện chính với các thay đổi được áp dụng.

#### ***1.1.4 Thông tin các đối tượng cần xử lý, quản lý***

- ***Nhóm các thông tin liên quan đến con người:***

- ❖ Người dùng:
  - Thành viên: tên đăng nhập, mật khẩu (hashed), họ tên, địa chỉ, email, số điện thoại
  - Người dùng cấp 1 (Admin):
    - Có thể tạo và quản lý các trạm, tài khoản người dùng cấp 2, quản lý cảm biến của tất cả các trạm
  - Người dùng cấp 2 (Người dùng trạm):
    - Quản lý các trạm mà họ được chỉ định, bao gồm giám sát và điều khiển cảm biến của các trạm
- **Nhóm các thông tin liên quan đến trạm và cảm biến:**
  - ❖ Trạm:
    - Tên trạm, vị trí, WebSocket URI, cổng kết nối, người dùng quản lý, các cảm biến trong trạm
    - Thông số môi trường: nhiệt độ, độ ẩm, trạng thái LED, tốc độ quạt, trạng thái còi báo động
    - Thời gian tạo, thời gian cập nhật
  - ❖ Cảm biến:
    - Mã cảm biến, loại cảm biến (DHT, LED, Quạt, Buzzer), trạng thái cảm biến (OK, ERROR), thời gian kiểm tra gần nhất
    - Mỗi cảm biến liên kết với một trạm
- **Nhóm các thông tin liên quan đến hệ thống:**
  - ❖ Thiết bị điều khiển:
    - Bao gồm LED, Quạt, và còi báo động, trạng thái hoặc tốc độ của các thiết bị này có thể được điều chỉnh bởi người dùng cấp 2 và người dùng cấp 1
    - Trạng thái thiết bị sẽ được lưu trữ và cập nhật tự động dựa trên thông số từ cảm biến
- **Nhóm các thông tin liên quan đến vận hành hệ thống:**
  - ❖ Ngưỡng cảnh báo
    - Người dùng cấp 1 có thể thiết lập ngưỡng cảnh báo cho từng trạm và người dùng cấp 2 có thể thiết lập ngưỡng cảnh báo cho trạm được phân công

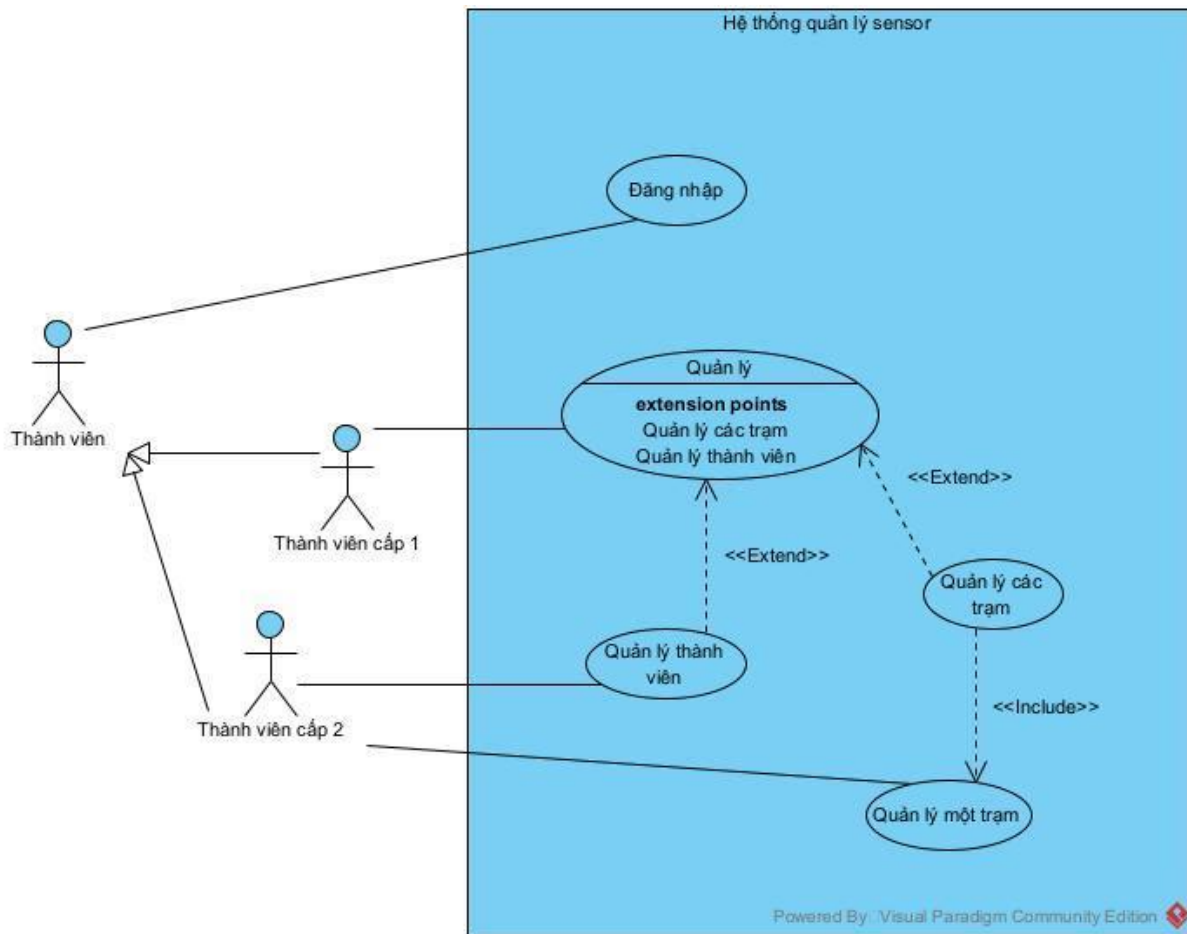
- Khi nhiệt độ hoặc độ ẩm vượt quá ngưỡng đã thiết lập, hệ thống sẽ tự động điều chỉnh thiết bị (bật/tắt quạt, LED, còi báo động)
- ***Nhóm thông tin liên quan đến thống kê:***
  - ❖ **Thống kê theo trạm:**
    - Báo cáo tình trạng hoạt động của các trạm và cảm biến theo thời gian
    - Lưu lại log trạng thái của các thiết bị điều khiển (bật/tắt) và thời gian hoạt động của chúng
    - Thống kê số lượng cảnh báo đã phát sinh trong một khoảng thời gian nhất định
    - Thống kê thời gian lỗi cảm biến và thời gian bảo trì

#### ***1.1.5 Quan hệ giữa các đối tượng:***

- Một người dùng cấp 1 có thể quản lý nhiều trạm
- Một người dùng cấp 2 chỉ có thể quản lý các trạm mà họ được phân quyền
- Một trạm có nhiều hệ thống cảm biến
- Một trạm có thể được giám sát và điều khiển bởi một hoặc nhiều người dùng
- Một hệ thống cảm biến liên kết với một trạm duy nhất
- Một hệ thống cảm biến có thể có nhiều thiết bị điều khiển (LED, quạt, còi báo động), và trạng thái của các thiết bị này được cập nhật dựa trên dữ liệu từ cảm biến
- Một thiết bị (LED, quạt, còi báo động) có thể được điều chỉnh bởi người dùng cấp 2, trong phạm vi quyền hạn của họ.

## 1.2. Mô tả bằng UML

### 1.2.1 Mô hình tổng quát



**Hình 3. 1: Mô hình tổng quát**

Mô tả usecase tổng quát

- Đăng nhập: UC này cho phép thành viên của hệ thống (thành viên cấp 1 hoặc cấp 2) đăng nhập vào hệ thống bằng tên đăng nhập và mật khẩu đã được cấp.
- Quản lý trạm: UC này cho phép thành viên cấp 1 quản lý các trạm trong hệ thống, bao gồm việc thêm, sửa, xóa, và xem chi tiết các trạm và cảm biến liên quan.
- Quản lý một trạm: UC này cho phép thành viên cấp 2 quản lý và giám sát các cảm biến của trạm mà họ được chỉ định, bao gồm việc điều chỉnh thiết bị điều khiển và theo dõi trạng thái cảm biến.



- Quản lý thành viên: UC này cho phép thành viên cấp 1 quản lý các tài khoản thành viên khác, bao gồm việc tạo mới, sửa, hoặc xóa tài khoản thành viên cấp 2 và cấp quyền cho các thành viên mới.

### ***1.2.2. Mô hình chi tiết***

#### ***1. Đăng nhập***

- Mã Use Case: UC01
- Mô tả: Cho phép thành viên đăng nhập vào hệ thống quản lý cảm biến.
- Diễn viên: Thành viên (cấp 1 và cấp 2)
- Tiền điều kiện:
- Thành viên đã được cấp tài khoản với tên đăng nhập và mật khẩu hợp lệ.
- Dòng sự kiện chính:

1. Thành viên mở giao diện đăng nhập.

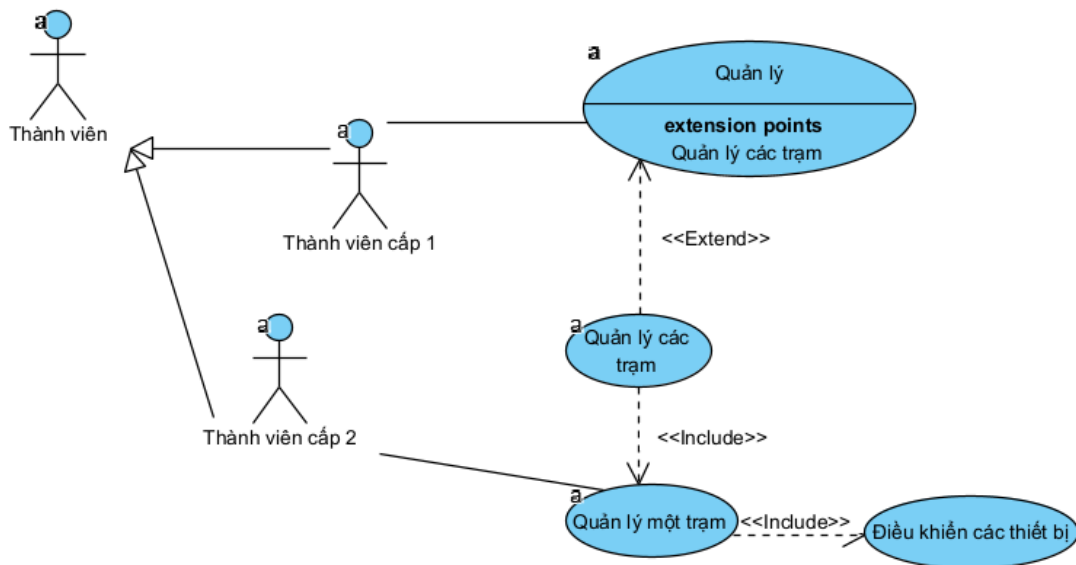
2. Thành viên nhập tên đăng nhập và mật khẩu.

3. Hệ thống xác thực thông tin đăng nhập.

4. Nếu thông tin chính xác, hệ thống cho phép thành viên truy cập vào trang quản lý cảm biến hoặc trang chính của thành viên.

- Dòng sự kiện mở rộng:
  - Nếu thông tin không chính xác, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.
- Hậu điều kiện:
  - Thành viên đã đăng nhập thành công và có thể truy cập các chức năng được phân quyền.

#### ***2. Quản lý các trạm***



**Hình 3. 2: Mô hình Quản lý các trạm**

- Mã Use Case: UC02
- Mô tả: Cho phép thành viên cấp 1 quản lý các trạm, bao gồm thêm, sửa, xóa, và xem danh sách trạm cảm biến.
- Diễn viên: Thành viên cấp 1
- Tiền điều kiện:
- Thành viên cấp 1 đã đăng nhập thành công.
- Dòng sự kiện chính:
  1. Thành viên cấp 1 chọn chức năng "Quản lý các trạm".
  2. Hệ thống hiển thị danh sách các trạm đã được cấu hình trong hệ thống.
  3. Thành viên cấp 1 có thể thực hiện các hành động:
    - Thêm mới trạm: Nhập thông tin về trạm và các cảm biến liên quan.
    - Sửa thông tin trạm: Cập nhật thông tin liên quan đến trạm.
    - Xóa trạm: Loại bỏ trạm khỏi hệ thống.
    - Điều khiển các thiết bị ở các trạm
  4. Sau khi thực hiện hành động, hệ thống cập nhật trạng thái của các trạm.

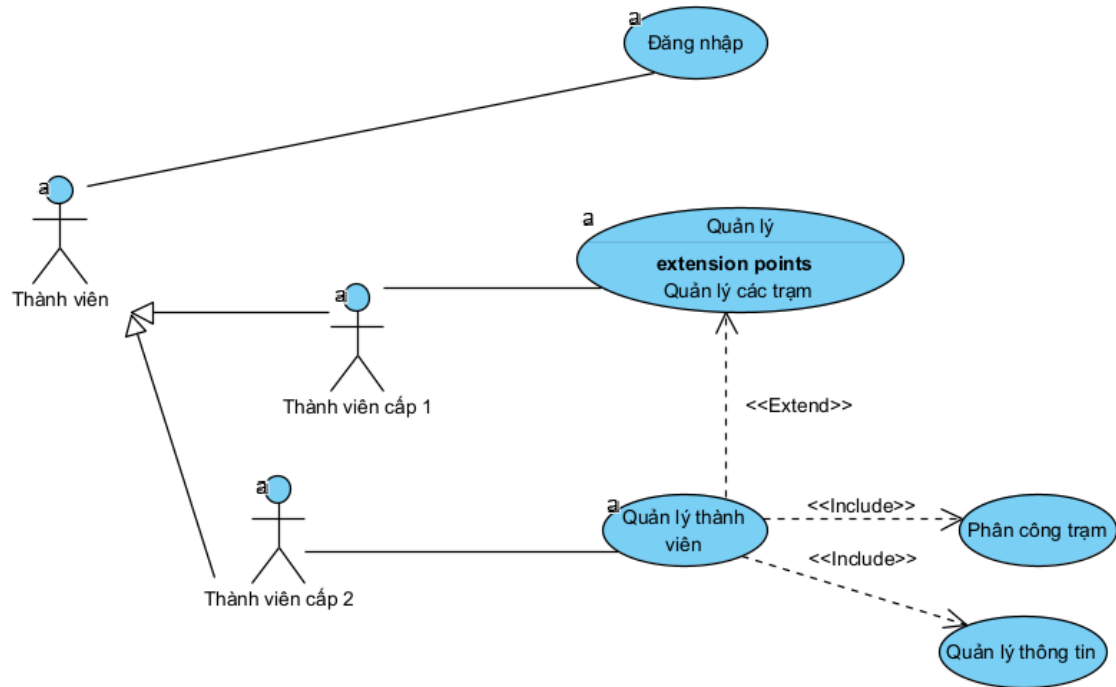
- Dòng sự kiện mở rộng:
  - Nếu thành viên không có quyền thực hiện thao tác, hệ thống sẽ hiển thị thông báo lỗi.
- Hậu điều kiện:
  - Hệ thống cập nhật trạng thái của các trạm theo hành động của thành viên cấp 1.

### **3. Quản lý một trạm**

- Mã Use Case: UC03
- Mô tả: Cho phép thành viên cấp 2 quản lý các cảm biến thuộc trạm được phân quyền.
- Diễn viên: Thành viên cấp 2
- Tiền điều kiện:
- Thành viên cấp 2 đã đăng nhập thành công và được phân quyền quản lý một hoặc nhiều trạm.
- Dòng sự kiện chính:
  1. Thành viên cấp 2 chọn trạm mà họ được phân quyền.
  2. Hệ thống hiển thị danh sách các cảm biến thuộc trạm đó.
  3. Thành viên cấp 2 có thể:
    - Kiểm tra trạng thái của các cảm biến (nhiệt độ, độ ẩm, LED, quạt, còi).
    - Điều chỉnh các thiết bị như LED, quạt hoặc còi dựa trên các ngưỡng cảnh báo.
  4. Hệ thống ghi nhận và phản hồi các thay đổi trong trạng thái cảm biến.
- Dòng sự kiện mở rộng:
  - Nếu thành viên không có quyền truy cập trạm, hệ thống hiển thị thông báo lỗi.
- Hậu điều kiện:

- Cảm biến và thiết bị trong trạm được quản lý theo thông tin cập nhật từ thành viên cấp 2.

#### 4. Quản lý thành viên



**Hình 3. 3: Mô hình Quản lý thành viên**

- Mã Use Case: UC04
- Mô tả: Cho phép thành viên cấp 1 quản lý thông tin tài khoản của các thành viên khác.
- Diễn viên: Thành viên cấp 1
- Tiền điều kiện:
- Thành viên cấp 1 đã đăng nhập thành công.
- Dòng sự kiện chính:
  1. Thành viên cấp 1 chọn chức năng "Quản lý thành viên".
  2. Hệ thống hiển thị danh sách các thành viên hiện có trong hệ thống.
  3. Thành viên cấp 1 có thể thực hiện các hành động:

- Thêm mới thành viên: Nhập thông tin cá nhân của thành viên mới (họ tên, địa chỉ, email, mật khẩu, cấp độ).
- Sửa thông tin thành viên: Cập nhật thông tin cá nhân hoặc phân quyền.
- Xóa thành viên: Loại bỏ tài khoản thành viên ra khỏi hệ thống.
- Phân công trạm cho nhân viên cấp 2

#### 4. Hệ thống ghi nhận và cập nhật thông tin các thành viên.

- Dòng sự kiện mở rộng:
  - Nếu thành viên không có quyền quản lý thành viên, hệ thống hiển thị thông báo lỗi.
- Hậu điều kiện:
  - Thông tin tài khoản thành viên được cập nhật hoặc thay đổi theo hành động của thành viên cấp 1.

#### **5. Quản lý các cảm biến của trạm**

- Mã Use Case: UC05
- Mô tả: Cho phép thành viên cấp 2 kiểm tra và điều khiển các cảm biến trong một trạm cụ thể.
- Diễn viên: Thành viên cấp 2
- Tiền điều kiện:
- Thành viên cấp 2 đã đăng nhập thành công và được phân quyền quản lý trạm.
- Dòng sự kiện chính:

1. Thành viên cấp 2 chọn trạm cụ thể mà họ được quản lý.

2. Hệ thống hiển thị trạng thái của các cảm biến như nhiệt độ, độ ẩm, LED, quạt, và còi.

3. Thành viên cấp 2 có thể điều khiển các thiết bị liên quan như bật/tắt LED, kích hoạt quạt hoặc còi khi cần thiết.

4. Hệ thống ghi nhận các thay đổi và phản hồi trạng thái mới của các cảm biến và thiết bị.

- Dòng sự kiện mở rộng:
- Nếu cảm biến không phản hồi, hệ thống hiển thị lỗi kỹ thuật.
- Hậu điều kiện:
- Cảm biến được kiểm tra và thiết bị được điều khiển theo hành động của thành viên cấp 2.

## ***6. Xem thông tin tổng quan các trạm***

Mã Use Case: UC06

- Mô tả: Cho phép thành viên cấp 1 xem tổng quan trạng thái của tất cả các trạm trong hệ thống.
- Diễn viên: Thành viên cấp 1
- Tiền điều kiện:
- Thành viên cấp 1 đã đăng nhập thành công.
- Dòng sự kiện chính:
  1. Thành viên cấp 1 chọn chức năng "Xem thông tin tổng quan".
  2. Hệ thống hiển thị trạng thái hoạt động của tất cả các trạm, bao gồm các thông số như nhiệt độ, độ ẩm, trạng thái LED, quạt, còi.
  3. Thành viên cấp 1 có thể chọn một trạm để xem chi tiết trạng thái từng cảm biến.
  4. Hệ thống hiển thị thông tin chi tiết theo yêu cầu.
- Hậu điều kiện:
- Thành viên cấp 1 nắm bắt được tổng quan trạng thái của hệ thống và từng trạm cảm biến.

## **2. Giải pháp thiết kế**

### ***2.1 Sơ đồ khối***

- Cảm biến DHT11 -> ESP32 -> Màn hình LCD
- ESP32 -> Quạt

- ESP32 -> Đèn LED
- ESP32 -> Loa buzzer
- ESP32 -> Gửi cảnh báo (email/thông báo)
- ESP32 -> Web app (hiển thị dữ liệu và điều khiển từ xa)

## ***2.2. Nguyên lý hoạt động của hệ thống***

- Cảm biến nhiệt độ và độ ẩm: Cảm biến DHT11 đo nhiệt độ và độ ẩm của môi trường xung quanh và gửi dữ liệu này đến ESP32.
- ESP32: Xử lý dữ liệu từ cảm biến và hiển thị lên màn hình LCD. Khi nhiệt độ hoặc độ ẩm vượt ngưỡng, ESP32 sẽ kích hoạt quạt, đèn LED và loa buzzer.
- Web App: Sử dụng Angular để xây dựng giao diện web, hiển thị dữ liệu nhiệt độ và độ ẩm từ ESP32, gửi cảnh báo và cho phép người dùng điều khiển quạt, đèn LED và loa buzzer từ xa. Người dùng cũng có thể đặt ngưỡng nhiệt độ và độ ẩm từ web app.

## CHƯƠNG 4. PHÂN TÍCH THIẾT KẾ

### 1. Mô tả hệ thống:

Hệ thống bao gồm các cảm biến nhiệt độ và độ ẩm (DHT11), một bộ vi điều khiển (ESP32) để xử lý dữ liệu, màn hình LCD để hiển thị thông tin, quạt để điều chỉnh nhiệt độ, đèn LED và loa buzzer để cảnh báo. Dữ liệu sẽ được truyền tới web app để hiển thị và điều khiển từ xa.

### 2. API và các hàm sẽ implement

#### 2.1 API:

- a. `getTemperature()`: Lấy dữ liệu nhiệt độ từ cảm biến DHT11.
- b. `getHumidity()`: Lấy dữ liệu độ ẩm từ cảm biến DHT11.
- c. `sendAlert(message)`: Gửi cảnh báo qua email hoặc thông báo.
- d. `displayData(temp, humidity)`: Hiển thị dữ liệu lên màn hình LCD.
- e. `adjustThreshold(tempThreshold, humidityThreshold)`: Điều chỉnh ngưỡng nhiệt độ và độ ẩm.
- f. `activateFan()`: Kích hoạt quạt khi nhiệt độ vượt ngưỡng.
- g. `activateLED()`: Kích hoạt đèn LED khi nhiệt độ hoặc độ ẩm vượt ngưỡng.
- h. `activateBuzzer()`: Kích hoạt loa buzzer khi nhiệt độ hoặc độ ẩm vượt ngưỡng.
- i. `sendDataToWebApp(temp, humidity)`: Truyền dữ liệu tới web app.
- j. `controlDevicesFromWebApp(command)`: Điều khiển quạt, đèn LED và loa buzzer từ web app.
- k. `setThresholdsFromWebApp(tempThreshold, humidityThreshold)`: Đặt ngưỡng nhiệt độ và độ ẩm từ web app.



## 2.2 Hàm:

- a. `readSensorData()`: Đọc dữ liệu từ cảm biến DHT11.
- b. `processData()`: Xử lý dữ liệu thu thập được.
- c. `checkThresholds(temp, humidity)`: Kiểm tra xem nhiệt độ và độ ẩm có vượt ngưỡng không.
- d. `activateDevices()`: Kích hoạt quạt, đèn LED và loa buzzer khi cần thiết.
- e. `sendData()`: Truyền dữ liệu tới web app.
- f. `receiveCommands()`: Nhận lệnh điều khiển từ web app.
- g. `setThresholds()`: Đặt ngưỡng nhiệt độ và độ ẩm từ web app.

## 3. Thực thi thiết kế

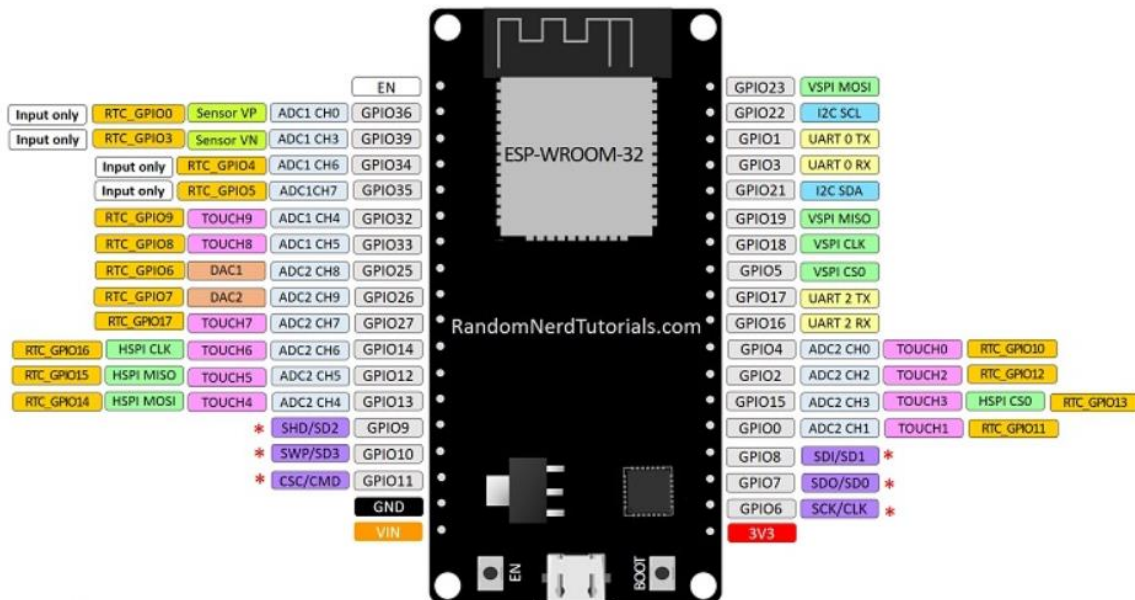
### 3.1 Thiết kế phần cứng

#### Lựa chọn linh kiện

#### 3.1.1 ESP32

Module ESP32 được sử dụng cho khối xử lý trung tâm với chức năng giao tiếp qua sóng Wi-Fi tiện lợi, đơn giản và gọn nhẹ. Để giao tiếp với ESP32, chúng ta có thể sử dụng các lệnh AT hoặc lập trình trực tiếp thông qua các thư viện như ESP-IDF hoặc Arduino. ESP32 có các thông số kỹ thuật cơ bản sau:

- **Điện áp hoạt động (+Vcc):** 3.3V
- **Giao tiếp:** UART, SPI, I2C
- **Chuẩn Wi-Fi:** 802.21 b/g/n; Wi-Fi Direct (P2P), soft-AP
- **Tích hợp giao thức TCP/IP stack**
- **Công suất đầu ra:** 20.5dBm ở chế độ 802.21b
- **Bluetooth:** BLE và Classic Bluetooth



**Hình 4. 1: Module Wi-Fi ESP32**

### Các chân GPIO

ESP32 có tổng cộng 34 chân GPIO, mỗi chân có thể được sử dụng như một đầu vào hoặc đầu ra chung hoặc kết nối với tín hiệu ngoại vi nội bộ.

### Các chức năng chính của các chân

- ADC (Analog-to-Digital Converter): 18 kênh ADC 12-bit.
- DAC (Digital-to-Analog Converter): 2 kênh DAC 8-bit.
- SPI (Serial Peripheral Interface): 3 giao diện SPI.
- I2C (Inter-Integrated Circuit): 2 giao diện I2C.
- UART (Universal Asynchronous Receiver/Transmitter): 3 giao diện UART.
- PWM (Pulse Width Modulation): 16 kênh PWM.
- I2S (Inter-IC Sound): 2 giao diện I2S.
- Cảm biến chạm: 10 chân GPIO có khả năng cảm biến chạm.

### Lưu ý khi sử dụng các chân GPIO

- Chân 0: Phải ở mức thấp để vào chế độ flash.
- Chân 2: Kết nối với đèn LED trên bo mạch, phải ở mức thấp hoặc nổi để vào chế độ flash.
- Chân 12: Boot sẽ thất bại nếu chân này ở mức cao.

- Chân 15: Phải ở mức thấp để vào chế độ flash

Chip ESP32 là chip được tích hợp trên module Wi-Fi ESP32. ESP32 là một trong những chip Wi-Fi tích hợp nhất trong ngành công nghiệp. Với kích thước nhỏ gọn, ESP32 đòi hỏi mạch điện bên ngoài tối thiểu và tích hợp vi điều khiển 32-bit Tensilica, tiêu chuẩn kỹ thuật số giao diện ngoại vi, chuyển mạch ăng-ten, RF balun, bộ khuếch đại lớn, tiếng ồn thấp nhận được khuếch đại, bộ lọc và các module quản lý điện năng tất cả trong một gói nhỏ.

Chip ESP32 tích hợp Tensilica Xtensa LX6 32-bit, vi điều khiển (MCU) có tính năng tiêu thụ điện năng thấp và đạt tốc độ tối đa là 240MHz. Được thiết kế cho các thiết bị di động, thiết bị điện tử đeo trên người và các ứng dụng Internet of Things (IoT), ESP32 đạt được mức tiêu thụ điện năng thấp với một sự kết hợp của nhiều công nghệ độc quyền. Các kiến trúc tiết kiệm điện có ba chế độ hoạt động Chế độ hoạt động, chế độ ngủ và chế độ ngủ sâu, do đó cho phép thiết kế pin chạy lâu hơn.

Với phạm vi nhiệt độ hoạt động rộng từ -40°C đến 125°C, ESP32 có khả năng hoạt động trong môi trường công nghiệp. Với tính năng tích hợp cao trên chip và tối thiểu số thành phần rời rạc bên ngoài, các chip cung cấp độ tin cậy cao, nhỏ gọn và mạnh mẽ.

### **3.1.2. DHT11**

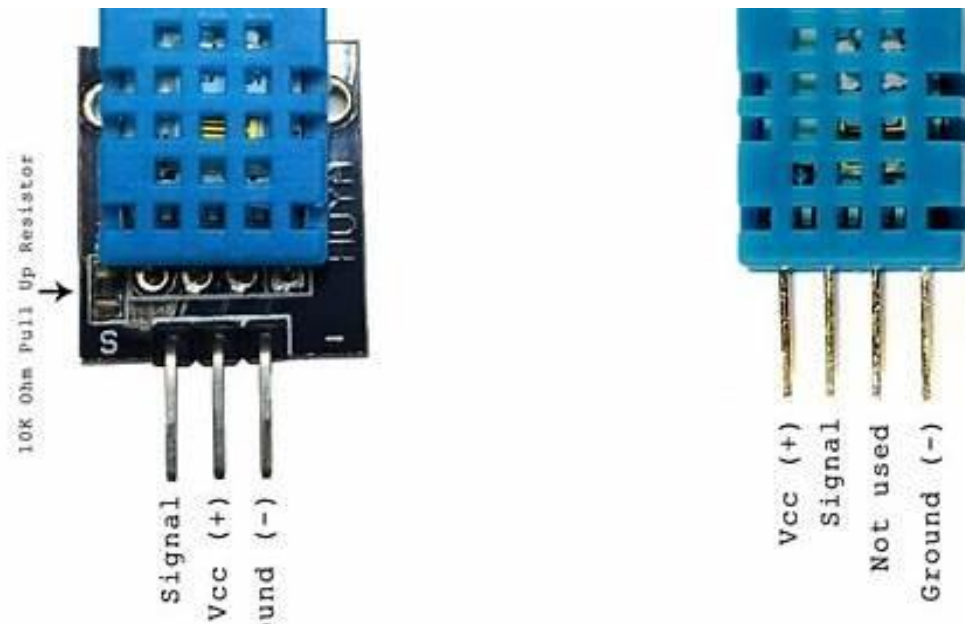
**Thông số kỹ thuật cơ bản:**

- **Điện áp hoạt động:** 3.3V – 5.5V
- **Dải đo độ ẩm:** 20% 80%
- **Sai số độ ẩm:**  $\pm 5\%$
- **Dải đo nhiệt độ:** 0°C – 50°C
- **Sai số nhiệt độ:**  $\pm 2^\circ\text{C}$

**Chức năng chân của DHT11:**

- **Chân 1 (VDD):** Chân nối nguồn.
- **Chân 2 (DATA):** Chân dữ liệu vào ra.

- **Chân 3 (NC):** Không dùng.
- **Chân 4 (GND):** Chân nối mass (0V).



**Hình 4. 2: Hình DHT11**

### **Nguyên lý hoạt động**

DHT11 sử dụng một cảm biến độ ẩm điện dung và một nhiệt điện trở (thermistor) để đo độ ẩm và nhiệt độ. Cảm biến độ ẩm điện dung có hai điện cực với một chất nền giữ ẩm làm chất điện môi giữa chúng. Khi độ ẩm thay đổi, giá trị điện dung cũng thay đổi. Nhiệt điện trở có hệ số nhiệt độ âm (NTC), nghĩa là điện trở giảm khi nhiệt độ tăng.

### **Giao tiếp với DHT11:**

Để giao tiếp với DHT11, vi điều khiển thực hiện theo 2 bước chính:

#### **Bước 1: Gửi tín hiệu muốn đo (Start) tới DHT11**

1. **Thiết lập chân DATA là output và kéo chân DATA xuống 0** trong khoảng thời gian  $\geq 18\text{ms}$ . Khi đó, DHT11 sẽ hiểu là vi điều khiển muốn đo nhiệt độ và độ ẩm.
2. **Đưa chân DATA lên 1** và sau đó thiết lập lại là chân đầu vào.

3. Sau khoảng 20-40 $\mu$ s, DHT11 sẽ kéo chân DATA xuống thấp. Nếu sau >40 $\mu$ s mà chân DATA chưa được kéo xuống thấp, nghĩa là chưa giao tiếp được với DHT11.
4. Chân DATA sẽ ở mức thấp 80 $\mu$ s sau đó được DHT11 kéo lên mức cao trong 80 $\mu$ s. Bằng việc giám sát chân DATA, vi điều khiển có thể biết được có giao tiếp được với DHT11 hay không. Nếu tín hiệu đo được lên cao, quá trình giao tiếp của vi điều khiển với DHT11 hoàn thiện.

## **Bước 2: Đọc giá trị trên DHT11**

Cảm biến sẽ gửi lại 5 byte dữ liệu đo được:

- **Byte 1:** Giá trị phần nguyên của độ ẩm.
- **Byte 2:** Giá trị phần thập phân của độ ẩm.
- **Byte 3:** Giá trị phần nguyên của nhiệt độ.
- **Byte 4:** Giá trị phần thập phân của nhiệt độ.
- **Byte 5:** Kiểm tra tổng.

Nếu Byte 5 = Byte 1 + Byte 2 + Byte 3 + Byte 4 thì giá trị nhiệt độ và độ ẩm là chính xác, còn nếu không thì kết quả đo bị sai.

### **Cách tính nhiệt độ và độ ẩm:**

- **Công thức:** (Byte cao \* 256 + Byte thấp) / 10

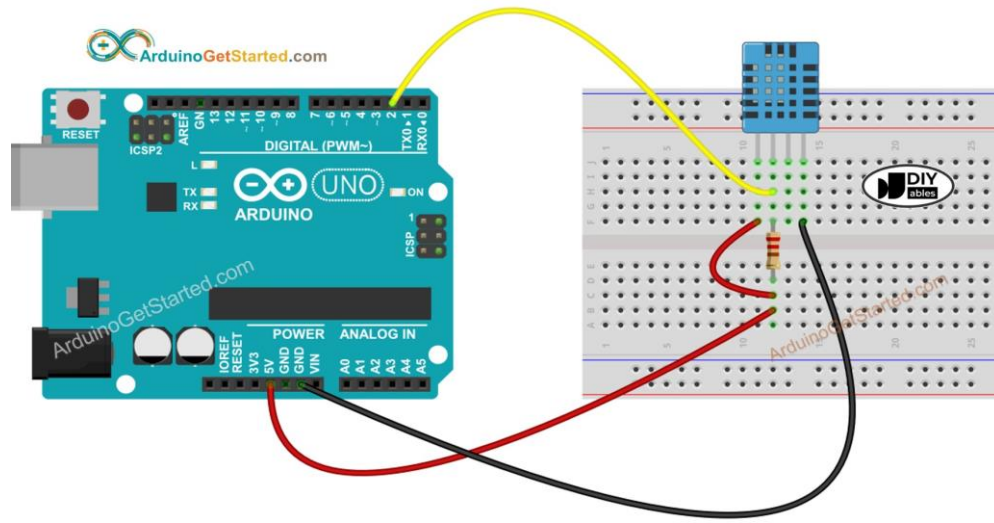
### **Đọc dữ liệu:**

Sau khi giao tiếp được với DHT11, DHT11 sẽ gửi liên tiếp 40 bit 0 hoặc 1 về vi điều khiển tương ứng với 5 byte giá trị nhiệt độ và độ ẩm.

- **Bit 0:** Sau khi tín hiệu được đưa về 0, đợi chân DATA của vi điều khiển được DHT11 kéo lên 1. Nếu chân DATA trong khoảng 26-28 $\mu$ s thì là 0.
- **Bit 1:** Nếu chân DATA tồn tại trong khoảng 70 $\mu$ s thì là 1.

Trong lập trình, bắt sườn lên của DATA sau đó delay 50 $\mu$ s. Nếu giá trị đo được là 0 thì đọc được bit 0, nếu giá trị đo được là 1 thì đọc được bit 1. Cứ thế đọc các bit tiếp theo.

## Kết nối với ESP32



**Hình 4. 3: Kết nối DHT11 với ESP32**

- VCC của DHT11 kết nối với chân 3.3V của ESP32.
- GND của DHT11 kết nối với chân GND của ESP32.
- Data của DHT11 kết nối với một chân GPIO của ESP32 (ví dụ: GPIO 21). Bạn có thể cần một điện trở kéo lên 10kΩ giữa chân Data và VCC để đảm bảo tín hiệu ổn định.

### 3.1.3. Quạt

#### Thông tin về động cơ quạt mini

- **Mô tả:** Động cơ quạt mini gồm hai bộ phận: động cơ và cánh quạt, giúp bạn có thể tạo thành một quạt máy mini và ứng dụng trong nhiều dự án sáng tạo khác nhau. Ví dụ: Bạn có thể lập trình để quạt tự động bật/tắt dựa theo nhiệt độ, tự động bật quạt khi có người.
- **Điện áp hoạt động:** 3.3V
- **Tín hiệu điều khiển:** 2 pins
- **Kích thước của mạch:** 24mm x 48mm x 16mm



*Hình 4. 4: Động cơ quạt mini*

#### **Pinout của động cơ quạt mini**

- Module động cơ quạt mini có 4 chân, và mỗi chân có chức năng như sau:
  - **GND: Nối đất**
  - **VCC: Cấp nguồn (3.3V)**
  - **S2: Tín hiệu điều khiển quay nghịch**
  - **S1: Tín hiệu điều khiển quay thuận**

#### **Nguyên lý hoạt động:**

- Quạt hoạt động dựa trên nguyên lý điện từ, khi dòng điện chạy qua cuộn dây, nó tạo ra từ trường làm quay cánh quạt.

#### **Các chân của quạt mini**

- **GND (Ground):** Chân nối đất, kết nối với GND của hệ thống.
- **VCC (Voltage Common Collector):** Chân cấp nguồn cho quạt, thường kết nối với nguồn 3.3V hoặc 5V.
- **NC (Not Connected):** Chân không kết nối, không sử dụng trong các ứng dụng thông thường.
- **SIG (Signal):** Chân tín hiệu, dùng để điều khiển quạt.

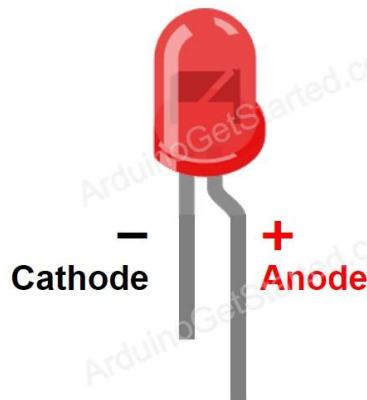
#### **Nguồn tham khảo**

- OhStem Education, 2024. Động cơ quạt mini. [online] Available at: <https://ohstem.vn/product/dong-co-quat-mini/> [Accessed 27 September 2024].

### 3.1.4. Đèn LED

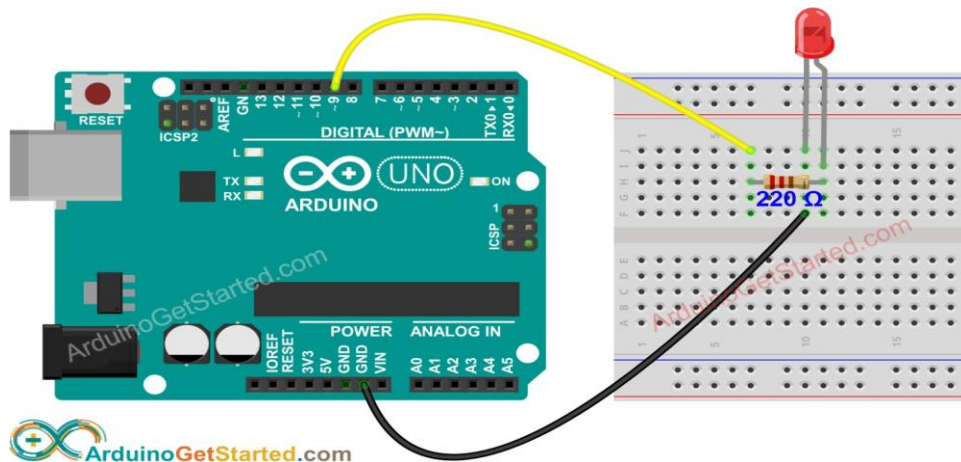
Thông tin về LED

- **LED (Light Emitting Diode)** là một loại diode bán dẫn phát sáng khi có dòng điện chạy qua.
- **Pinout của LED:**
  - **Cathode (-):** Chân này cần được kết nối với GND (0V).
  - **Anode (+):** Chân này được sử dụng để điều khiển trạng thái của LED.



Hình 4. 5: Ảnh LED

- **Cách điều khiển LED với ESP32:**
  - Chân dương của LED: Nối với một chân GPIO của ESP32 qua một điện trở 220 ohm.
  - Chân âm của LED: Nối với GND.

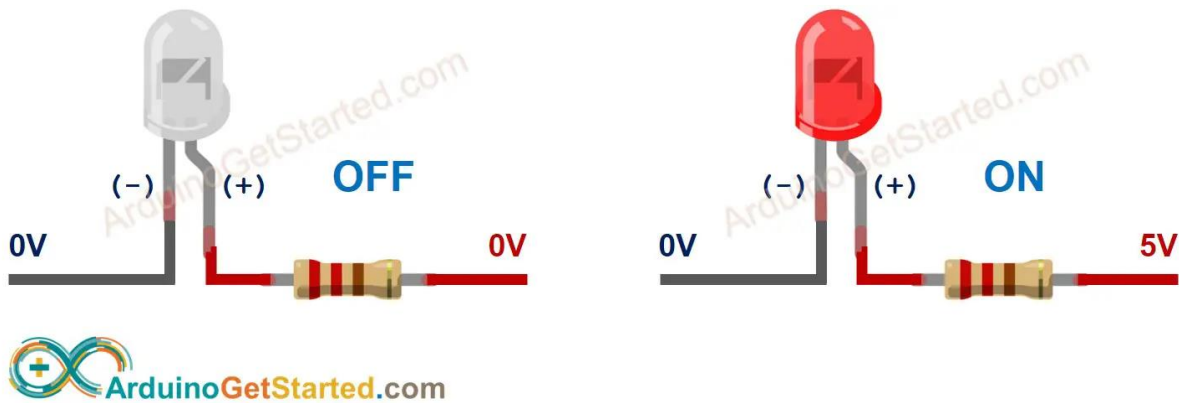


Hình 4. 6: Kết nối LED với ESP



- **Nguyên lý hoạt động:**

- Khi kết nối chân Cathode với GND và chân Anode với VCC, LED sẽ sáng.
- Khi kết nối chân Cathode với GND và chân Anode với GND, LED sẽ tắt.
- Nếu tạo tín hiệu PWM (Pulse Width Modulation) đến chân Anode, độ sáng của LED sẽ thay đổi theo giá trị PWM.



*Hình 4. 7: Nguyên lý hoạt động của LED*

### 3.1.5. Loa Buzzer

Thông tin về Piezo Buzzer

- **Piezo Buzzer** là một thiết bị phát âm thanh sử dụng hiệu ứng áp điện để tạo ra âm thanh.
- **Loại Buzzer:**
  - **Active Buzzer:** Tạo ra âm thanh khi có điện áp được áp dụng. Dễ sử dụng, chỉ cần nguồn điện.
  - **Passive Buzzer:** Cần tín hiệu dao động bên ngoài để tạo ra âm thanh. Có thể tạo ra các âm thanh khác nhau bằng cách thay đổi tần số đầu vào.
- **Pinout của Piezo Buzzer:**
  - Chân dương (+): Nhận tín hiệu điều khiển từ Arduino.
  - Chân âm (-): Kết nối với GND (0V).



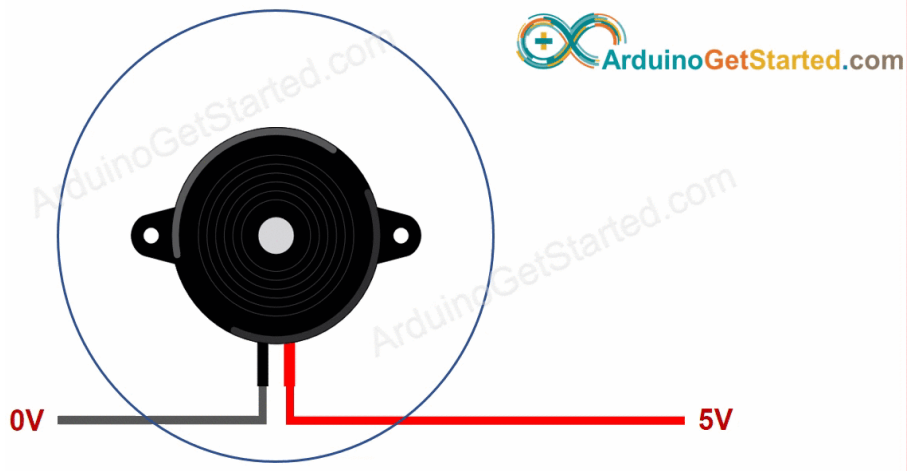
#### 3-5V Buzzer

- Connect directly to Arduino => normal sound

**Hình 4. 8: Piezo Buzzer**

Nguyên lý hoạt động

- **Active Buzzer:** Khi kết nối VCC với chân dương, Piezo Buzzer sẽ tạo ra âm thanh liên tục.



**Hình 4. 9: Active Buzzer hoạt động**

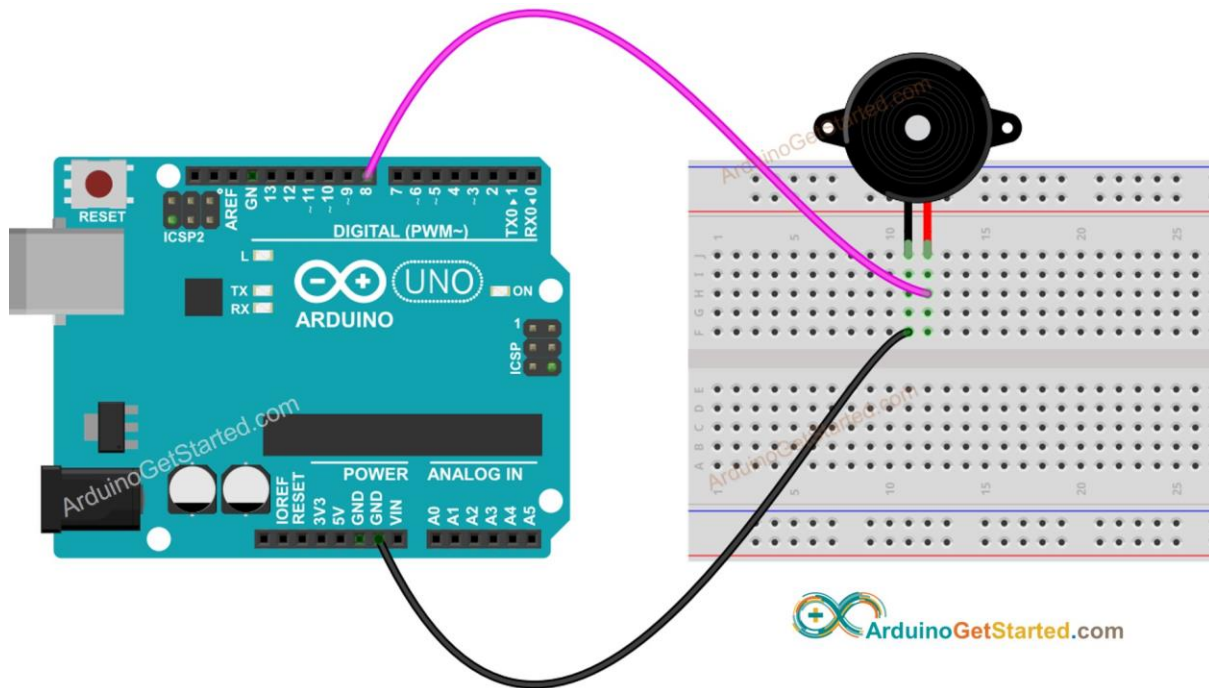
- **Passive Buzzer:** Khi tạo tín hiệu sóng vuông với tần số xác định trên chân dương, Piezo Buzzer sẽ tạo ra các âm thanh với tần số tương ứng.



Hình 4. 10: Passive Buzzer hoạt động

• **Kết nối với ESP:**

- Chân dương (+): Nhận tín hiệu điều khiển từ Arduino.
- Chân âm (-): Kết nối với GND (0V).



Hình 4. 11: Kết nối ESP với Buzzer

### 3.1.6. Màn hình LCD I2C

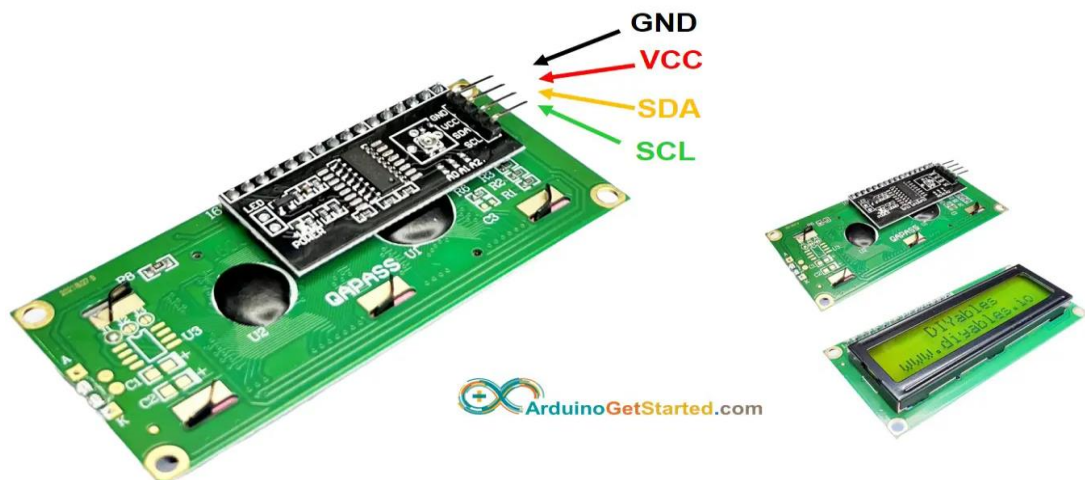
Thông tin về LCD I2C

- **LCD I2C:** Là loại màn hình LCD sử dụng giao tiếp I2C để đơn giản hóa việc kết nối với vi điều khiển. LCD I2C thường bao gồm một màn hình LCD thông thường, một module I2C và một biến trở.



*Hình 4. 12: Màn hình LCD I2C*

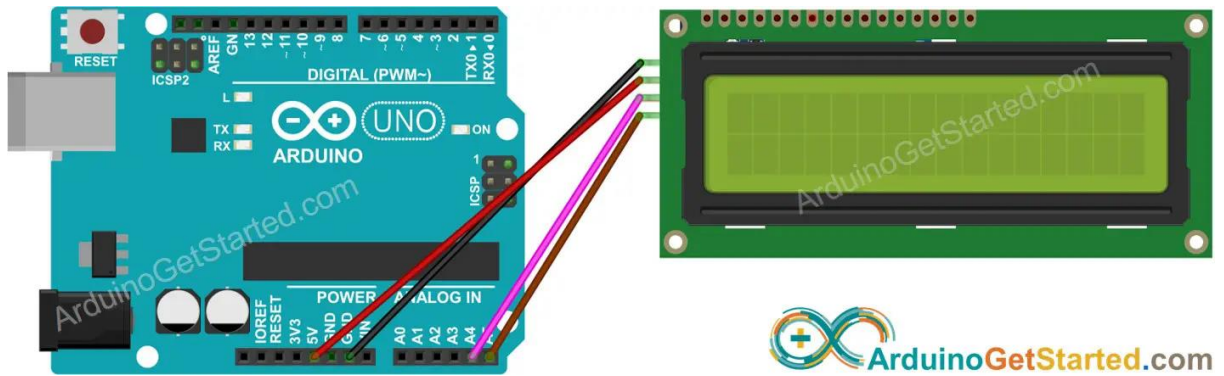
- **Pinout của LCD I2C:**
  - **GND:** Kết nối với GND (0V).
  - **VCC:** Cấp nguồn cho LCD, thường kết nối với 5V.
  - **SDA:** Tín hiệu dữ liệu I2C.
  - **SCL:** Tín hiệu đồng hồ I2C.



*Hình 4. 13: Màn hình LCD I2C*

## Kết nối với ESP

- GND của LCD I2C kết nối với GND của ESP32.
- VCC của LCD I2C kết nối với chân 3.3V hoặc 5V của ESP32.
- SDA của LCD I2C kết nối với GPIO 21 của ESP32.
- SCL của LCD I2C kết nối với GPIO 22 của ESP32.



Hình 4. 14: Kết nối màn hình LCD I2C với ESP

### 3.2 Thiết kế phần mềm

(Code chương trình ở phần phụ lục)

### 3.3 Lưu đồ thuật toán

## 4. Ngôn ngữ lập trình và phần mềm phụ trợ

## CHƯƠNG 5. ĐÁNH GIÁ KẾT QUẢ DỰ ÁN VÀ KẾT LUẬN

### 1. Bảng phân công công việc

Trịnh Lê Đức	Tìm hiểu, thu thập thông tin, lý thuyết liên quan đến IOT và dự án, tìm kiếm hướng phát triển để mở rộng hệ thống, xây dựng giao diện sensor, login, quản lý người dùng
Nguyễn Thị Lan	Tìm hiểu các công nghệ kết nối trong IOT, các xu hướng hiện tại và tương lai, tìm hiểu và lắp đặt các thiết bị, sensor, xây dựng giao diện quản lý các sensor, xây dựng các kết nối để nhận thông tin từ BE
Lê Đình Phúc	Tìm hiểu và lắp đặt các thiết bị, thiết kế hệ thống, xây dựng, phân tích sơ đồ, thiết kế cơ sở dữ liệu cho các cảm biến, thiết kế backend thực hiện kết nối tới các sensor, nhận thông tin từ các esp
Nguyễn Quang Anh	Tìm hiểu về các công nghệ truyền thông, thiết kế các chức năng quản lý người và các thông tin về quản lý sensor, nghiên cứu lắp đặt hệ thống cảm biến, xây dựng các API kết nối tới FE

### 2. Đánh giá kết quả dự án

### 3. Kết luận



## TÀI LIỆU THAM KHẢO

- 1) TopDev, 2024. \*HTTP là gì?\*. [online] Available at: <<https://topdev.vn/blog/http-la-gi/>> [Accessed 26 September 2024].
- 2) Khue Nguyen Creator, 2024. \*Giao thức MQTT là gì? Cách sử dụng?\*. [online] Available at: <<https://khuenguyencreator.com/giao-thuc-mqtt-la-gi-cach-su-dung/>> [Accessed 26 September 2024].
- 3) HiveMQ, 2024. \*How to get started with MQTT\*. [online] Available at: <<https://www.hivemq.com/blog/how-to-get-started-with-mqtt/>> [Accessed 26 September 2024].
- 4) Steve's Internet Guide, 2024. \*MQTT\*. [online] Available at: <<http://www.steves-internet-guide.com/mqtt/>> [Accessed 26 September 2024].
- 5) Techopedia, 2024. \*Bluetooth\*. [online] Available at: <<https://www.techopedia.com/definition/26198/bluetooth>> [Accessed 26 September 2024].
- 6) GeeksforGeeks, 2024. \*Introduction of Zigbee\*. [online] Available at: <<https://www.geeksforgeeks.org/introduction-of-zigbee/>> [Accessed 26 September 2024].
- 7) The Things Network, 2024. \*What is LoRaWAN\*. [online] Available at: <<https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/>> [Accessed 26 September 2024].
- 8) TutorialsPoint, 2024. \*HTTP\*. [online] Available at: <<https://www.tutorialspoint.com/http/index.htm>> [Accessed 26 September 2024].
- 9) GovTech, 2024. \*Introduction to WebSockets\*. [online] Available at: <[https://docs.developer.tech.gov.sg/docs/data-engineering-initiative-playbook/Chapter5/Introduction\\_to\\_WebSockets](https://docs.developer.tech.gov.sg/docs/data-engineering-initiative-playbook/Chapter5/Introduction_to_WebSockets)> [Accessed 26 September 2024].
- 10) Espressif Systems, 2020. \*ESP32 Series Datasheet\*. [online] Available at: <[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)> [Accessed 26 September 2024].
- 11) ArduinoGetStarted.com, 2024. \*Arduino Piezo Buzzer\*. [online] Available at: <<https://arduinogetstarted.com/tutorials/arduino-piezo-buzzer>> [Accessed 27 September 2024].
- 12) ArduinoGetStarted.com, 2024. \*Arduino LCD I2C\*. [online] Available at: <<https://arduinogetstarted.com/tutorials/arduino-lcd-i2c>> [Accessed 27 September 2024].
- 13) ArduinoGetStarted.com, 2024. \*Arduino LED Blink\*. [online] Available at: <<https://arduinogetstarted.com/tutorials/arduino-led-blink>> [Accessed 27 September 2024].
- 14) ArduinoGetStarted.com, 2024. \*Arduino DHT11\*. [online] Available at: <<https://arduinogetstarted.com/tutorials/arduino-dht11>> [Accessed 27 September 2024].



## **PHỤ LỤC**