

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KỸ THUẬT MÁY TÍNH**

**Đồng Quang Quyền – 19522111**

**Trương Bảo Nguyên – 19521924**

**Trịnh Trần Trung – 19522437**

**BÁO CÁO CE IOT CHALLENGE**  
**HỆ THỐNG XÁC ĐỊNH YÊU CẦU CỨU HỘ**  
**TRÊN DIỆN RỘNG SỬ DỤNG GIAO THỨC LORA**

**ĐÀ LẠT, 2021**

# MỤC LỤC

<b>Chương 1. GIỚI THIỆU ĐỀ TÀI .....</b>	<b>5</b>
1.1. Lý do chọn đề tài .....	5
1.2. Các nghiên cứu liên quan .....	5
1.3. Giải pháp đề xuất .....	6
<b>Chương 2. CƠ SỞ LÝ THUYẾT .....</b>	<b>6</b>
2.1. Công nghệ LoRaWAN .....	6
2.1.1. Khái niệm LoRa .....	6
2.1.2. Khái niệm LoRaWAN .....	6
2.1.3. Phạm vi hoạt động của mạng LoRaWAN .....	6
2.1.4. Các thành phần của mạng LoRaWAN .....	7
2.1.5. Cách hoạt động của LoRaWAN .....	7
2.1.6. Ứng dụng của LoRaWAN .....	7
2.1.7. Ưu điểm, nhược điểm của LoRaWAN .....	8
2.2. Các loại kiến trúc phổ biến trong hệ thống IoT .....	10
2.2.1. Các bộ phận cấu thành hệ thống IoT .....	10
2.2.2. Các loại kiến trúc phổ biến trong hệ thống IoT .....	10
2.3. GPS .....	13
2.3.1. Khái niệm GPS .....	13
2.3.2. Nguyên lý hoạt động .....	13
2.4. Web Server trên mạng cục bộ .....	14
2.4.1. HyperText Transfer Protocol (HTTP) .....	14
2.4.2. Nền tảng NodeJS .....	15

<b>Chương 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>	<b>17</b>
3.1. Sơ đồ hệ thống .....	17
3.2. Sử dụng DK Kit để gửi tín hiệu LoRa .....	18
3.2.1. Cấu hình chi tiết của DK Kit .....	18
3.2.2. Các bước gửi tín hiệu LoRa .....	19
3.3. Tạo Gateway bằng ESP32-CAM và UCA Education Board .....	19
3.3.1. UCA Education Board.....	19
3.3.2. ESP32-CAM.....	20
3.3.3. Tạo Gateway bằng ESP32-CAM và UCA Board .....	21
3.4. Tạo Web Server cục bộ bằng NodeJS .....	22
3.4.1. Mô hình MVC .....	22
3.4.2. Express framework.....	23
3.4.3. Cơ sở dữ liệu - MongoDB .....	24
3.4.4. Sơ đồ hệ thống Web Server.....	24
3.5. Ghép nối hệ thống.....	25
<b>Chương 4. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ .....</b>	<b>27</b>
4.1. Phạm vi hoạt động .....	27
4.2. Giám sát thời gian truyền gói tin .....	27
4.3. Giao diện người dùng .....	28
<b>Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI .....</b>	<b>30</b>
5.1. Kết luận.....	30
5.2. Hướng phát triển đề tài .....	30
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>31</b>

## DANH MỤC HÌNH

Hình 2.1: Cấu trúc mạng LoRaWAN.....	7
Hình 2.2: Mô hình kiến trúc Device-to-Device .....	10
Hình 2.3: Mô hình kiến trúc Device-to-Cloud .....	11
Hình 2.4: Mô hình kiến trúc Device-to-Gateway-to-Cloud FOG.....	12
Hình 2.5: Mô hình kiến trúc Device-to-Gateway-to-Cloud-to-Application .....	13
Hình 3.1: Sơ đồ theo kiến trúc Device-to-Gateway-to-Cloud-to-Application.....	17
Hình 3.2: RFThings Design Kit LS200 .....	18
Hình 3.3: Các bước gửi dữ liệu thông qua phương thức LoRa.....	19
Hình 3.4: UCA Education Board .....	19
Hình 3.5: ESP32-CAM .....	20
Hình 3.6: Mối liên kết của ESP32-CAM và UCA với hệ thống.....	21
Hình 3.7: Ghép nối UART giữa UCA và ESP32-CAM .....	21
Hình 3.8: Sơ đồ hoạt động của Gateway.....	22
Hình 3.9: Mô hình MVC.....	22
Hình 3.10: Tổ chức thư mục cơ bản.....	23
Hình 3.11: Quá trình gửi dữ liệu lên web server.....	24
Hình 3.12: Hiện thị giao diện người dùng .....	25
Hình 3.13: Gateway và toàn bộ thiết bị sau khi ghép nối hoàn tất .....	26
Hình 4.1: Bảng quản lí các trường hợp cần cứu hộ .....	28
Hình 4.2: Xác định vị trí trên bản đồ Google Maps.....	29

## Chương 1. GIỚI THIỆU ĐỀ TÀI

### 1.1. Lý do chọn đề tài

Với tình hình dân số ngày càng tăng, xã hội ngày càng phát triển kéo theo đó là sự phát sinh của nhiều vấn đề về an toàn sức khỏe, tính mạng, của cải của người dân. Những câu chuyện gặp khó khăn, mất mát từ nhỏ đến lớn vẫn luôn xảy ra trên các bản tin với số lượng chưa bao giờ giảm. Chúng ta vẫn thường hay nghe thấy trên báo đài các vụ trộm cướp, bắt cóc, mất tích hoặc các tai nạn đáng tiếc do thiên tai. Ngay cả với bản thân mỗi chúng ta vẫn không thoát khỏi những vấn đề như hỏng xe, hết xăng, đi lạc... mà không có thiết bị liên lạc hoặc bị hết pin. Nhận thấy những vấn đề trên, **“Hệ thống xác định yêu cầu cứu hộ trên diện rộng sử dụng giao thức LoRa”** ra đời nhằm hỗ trợ và cứu hộ nhanh nhất, chính xác nhất với những trường hợp đang gặp vấn đề cần được giúp đỡ.

### 1.2. Các nghiên cứu liên quan

Hiện nay có nhiều nghiên cứu liên quan trong lĩnh vực này nhưng chủ yếu là các thiết bị Tracking, theo dõi và giám sát người sử dụng. Ngoài ra có các thiết bị khác để tìm kiếm đồ vật của nhiều nhãn hàng nổi tiếng.

Đề tài **“Hệ thống xác định yêu cầu cứu hộ trên diện rộng sử dụng giao thức LoRa”** có một số khác biệt như đối tượng người dùng có thể là bất kỳ ai, vì không ai chắc chắn rằng mình không bao giờ gặp phải vấn đề gì. Một khác biệt rõ ràng khác là thiết bị còn có thể biết được chính xác vấn đề và người dùng đang gặp, từ đó điều động cứu hộ khẩn cấp đến địa điểm của người gặp nạn chứ không đơn thuần là theo dõi và tìm kiếm. Ngoài ra, thiết bị không hoạt động liên tục mà chỉ kích hoạt khi người dùng khởi động. Điều này giúp đảm bảo quyền riêng tư của người dùng và tiết kiệm năng lượng.

### 1.3. Giải pháp đề xuất

*“Hệ thống xác định yêu cầu cứu hộ trên diện rộng sử dụng giao thức LoRa”* thu thập vị trí và loại cứu hộ của người dùng sau đó sử dụng công nghệ LoRaWAN để truyền gói tin LoRa đến Gateway và chuyển lên Server.

Xây dựng Web Server để hiển thị thông tin cứu hộ của người đang gặp nạn, sau đó có cách hỗ trợ phù hợp với từng loại cứu hộ.

## Chương 2. CƠ SỞ LÝ THUYẾT

### 2.1. Công nghệ LoRaWAN

#### 2.1.1. Khái niệm LoRa

LoRa là viết tắt của long-range là một công nghệ điều chế RF cho mạng diện rộng công suất thấp (LPWAN) có khả năng truyền dữ liệu lên đến hàng km. Đặc biệt công nghệ Lora yêu cầu điện năng cực thấp, cho phép tạo ra các thiết bị hoạt động bằng pin với thời gian dài thậm chí lên tới 10 năm.

#### 2.1.2. Khái niệm LoRaWAN

LoRaWAN là một giao thức mạng mở cung cấp các kết nối giữa các cổng LPWAN với các thiết bị IoT ở nút cuối được tiêu chuẩn hóa và duy trì bởi LoRa Alliance.

#### 2.1.3. Phạm vi hoạt động của mạng LoRaWAN

LoRaWAN hoạt động trong phạm vi phổ không được cấp phép dưới 1GHz.

Ở mỗi khu vực khác nhau trên thế giới thì thiết bị LoRaWAN phải được cấu hình cho chip Lora hoạt động ở dải băng tần cho phép như ví dụ như EU là 868 MHz, US là 915 MHz, AS là 430 MHz...ở Việt Nam tần số mà ta hay sử dụng là 433 MHz.

#### 2.1.4. Các thành phần của mạng LoRaWAN

End Devices (thiết bị cuối): là một cảm biến hoặc thiết bị truyền động được kết nối không dây với mạng LoRaWAN thông qua các gateway sử dụng công nghệ điều chế LoRa.

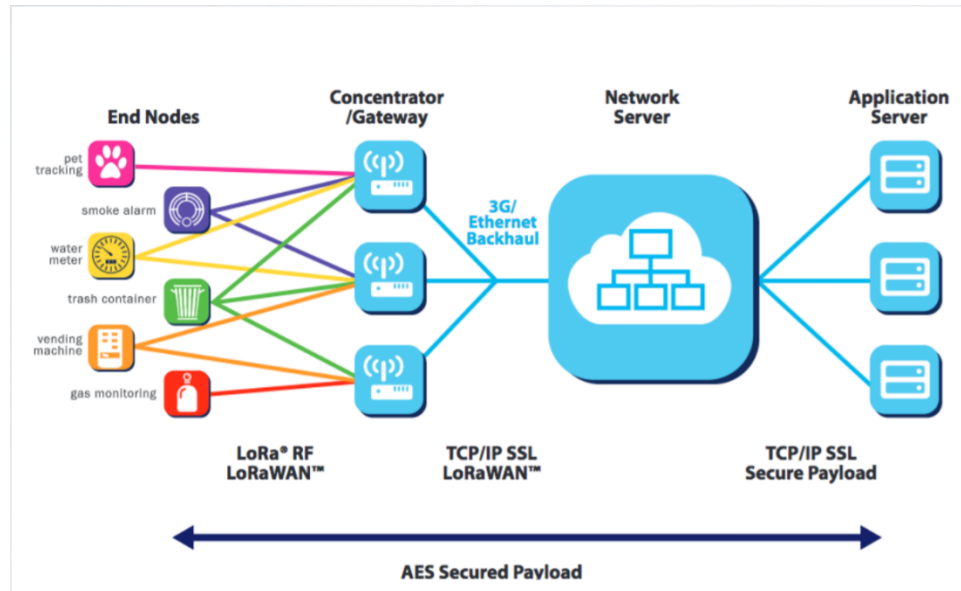
Gateway LoRaWAN (cổng LoRaWAN) : nhận các dữ liệu RF được điều chế LoRa từ các thiết bị cuối và chuyển tiếp dữ liệu này đến máy chủ ở mạng LoRaWAN.

Network server (máy chủ mạng): quản lý toàn bộ hệ thống mạng, các thông số thích hợp để điều chỉnh hệ thống và thiết lập kết nối AES 128-bit an toàn để truyền tải và kiểm soát dữ liệu.

#### 2.1.5. Cách hoạt động của LoRaWAN

Cấu trúc LoRaWAN network thường được đặt trong mô hình star-of-stars.

Giao tiếp giữa nút cuối(end node) và Gateway là hai chiều có nghĩa là nút cuối có thể gửi dữ liệu đến cổng nhưng nó cũng có thể nhận dữ liệu từ cổng.



Hình 2.1: Cấu trúc mạng LoRaWAN

#### 2.1.6. Ứng dụng của LoRaWAN

LoRaWAN được ứng dụng trong rất nhiều lĩnh vực khác nhau.

- Thành phố thông minh:
  - Hệ thống chiếu sáng thông minh
  - Giám sát chất lượng không khí và ô nhiễm
  - Quản lý phương tiện và bãi đậu xe thông minh
  - Cơ sở vật chất và quản lý cơ sở hạ tầng
  - Phát hiện và quản lý cháy
  - Quản lý chất thải
- Ứng dụng công nghiệp:
  - Phát hiện bức xạ và rò rỉ
  - Công nghệ cảm biến thông minh
  - Vị trí mặt hàng và theo dõi
  - Vận chuyển và trung chuyển
- Ứng dụng nhà thông minh:
  - Tăng cường an ninh gia đình
  - Tự động hóa nhà IoT cho phép các thiết bị thông minh
- Chăm sóc sức khỏe:
  - Quản lý và thiết bị theo dõi sức khỏe
  - Công nghệ may mặc
- Nông nghiệp:
  - Quản lý chăn nuôi và trồng trọt thông minh
  - Giám sát nhiệt độ và độ ẩm
  - Cảm biến mực nước và kiểm soát tưới tiêu

## **2.1.7. Ưu điểm, nhược điểm của LoRaWAN**

### **2.1.7.1. Ưu điểm**

- Các cảm biến được dùng với công suất thấp và vùng phủ sóng diện rộng.
- Hoạt động trên tần số miễn phí (không có giấy phép), không cần chi phí cấp phép trả trước để sử dụng công nghệ.
- Nguồn điện cấp thấp đồng nghĩa với thời lượng pin lâu dài cho các thiết bị.



- Các cổng Gateway được thiết kế để tiếp nhận hàng nghìn thiết bị cuối.
- Dễ dàng triển khai do kiến trúc mạng đơn giản.
- Nó được sử dụng rộng rãi cho các ứng dụng M2M / IoT.
- Dung lượng đường truyền lên đến (100 byte), so với SigFox là 12 byte.
- Tính “Mở”: công nghệ mở, tiêu chuẩn mở so với đối thủ cạnh tranh SigFox.
- Không hạn chế số lượng thông điệp tối đa mỗi ngày.
- LoRaWAN có các lợi ích với cách tiếp cận Mở thay vì độc quyền (SigFox).
- Tầm xa cho phép giải quyết các ứng dụng phạm vi rộng trong thành phố.
- Băng tần thấp khiến LoRaWAN trở nên lý tưởng cho các ứng dụng IoT thực tế với ít dữ liệu hơn hoặc với việc truyền dữ liệu không liên tục.
- Chi phí kết nối thấp.
- Không dây, dễ cài đặt và triển khai nhanh chóng.
- Bảo mật: một lớp cho mạng và một lớp cho ứng dụng với mã hóa AES.
- Hoàn toàn giao tiếp hai chiều.
- Được hỗ trợ bởi CISCO, IBM và 500 công ty thành viên của LoRa Alliance.

#### **2.1.7.2. Nhược điểm**

- Không dành cho các ứng dụng truyền dữ liệu lớn, giới hạn trong 100 byte.
- Không dùng để giám sát liên tục (trừ thiết bị loại C).
- Không phù hợp cho các ứng dụng thời gian thực yêu cầu độ trễ thấp và các yêu cầu về rung giật giới hạn.
- Mật độ các mạng LoRaWAN: Sự gia tăng của các công nghệ LPWAN, và đặc biệt là LoRaWAN, đặt ra những thách thức cùng nhau tồn tại khi việc triển khai các gateway vào khu vực đô thị.
- Bất lợi của tần số mở là bạn có thể bị nhiễu trên tần số đó và tốc độ dữ liệu có thể thấp. (Đối với GSM hoặc tần số được cấp phép, bạn có thể truyền trên tần số đó mà không bị nhiễu. Các nhà khai thác GSM sử dụng các tần số nhất định mà phải trả một khoản phí cấp phép lớn cho chính phủ để sử dụng các tần số đó. LoRa hoạt động trên các tần số mở và không cần trạng thái Giấy phép. (Hãy nhớ rằng các tần số mở là khác nhau giữa các quốc gia).

## 2.2. Các loại kiến trúc phổ biến trong hệ thống IoT

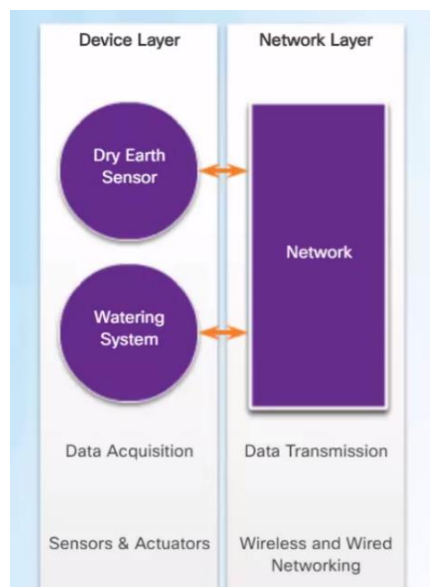
### 2.2.1. Các bộ phận cấu thành hệ thống IoT

Kiến trúc các hệ thống IoT được cấu thành từ 4 bộ phận chính:

- Vạn vật (Things): có thể là các vật dụng gia dụng, vật dụng công nghệ hoặc các vật dụng khác.
- Trạm kết nối (Gateways): là một thiết bị điện tử có thể kết nối các loại mạng khác nhau, cho phép kết nối thiết bị truyền thông cũ và thiết bị truyền thông mới lại với nhau. Trong hệ thống IOT, nó thực hiện định tuyến kết nối, tiền xử lý dữ liệu và chuyển tiếp dữ liệu giữa thiết bị và máy chủ.
- Hạ tầng (Network and Cloud): là một hệ thống nhiều mạng IP được kết nối với nhau và liên kết với các máy chủ, hệ thống lưu trữ.
- Các lớp tạo và cung cấp dịch vụ (Services – Creation and Solution Layers): các ứng dụng thu thập, phân tích dữ liệu, dự đoán, báo cáo số liệu thông qua hệ thống lưu trữ đám mây và những công cụ phần mềm (API).

### 2.2.2. Các loại kiến trúc phổ biến trong hệ thống IoT

#### 2.2.2.1. Kiến trúc Device-to-Device



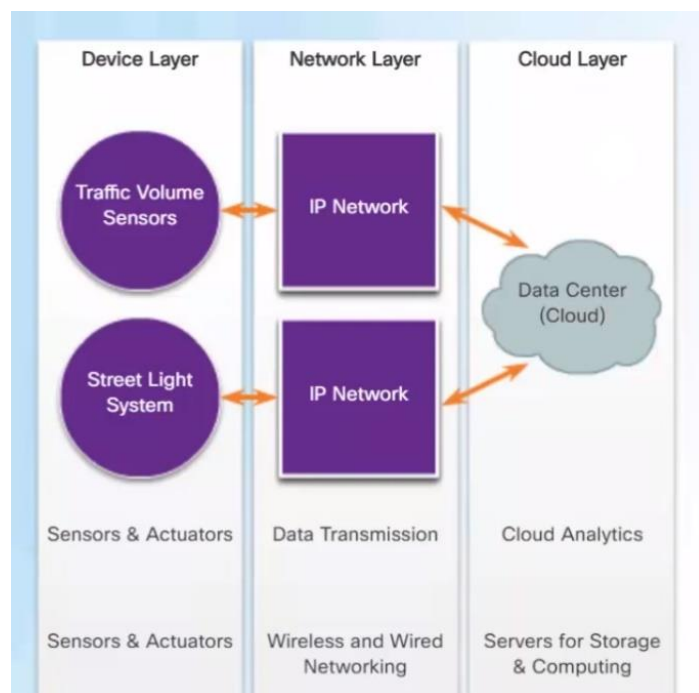
Hình 2.2: Mô hình kiến trúc Device-to-Device

Tầng thiết bị (Device Layer) chứa các cảm biến và thiết bị truyền động.

Tầng mạng (Network Layer) có thể là các thiết bị truyền thông mạng sử dụng các kết nối có dây hoặc không dây.

Thiết bị thu thập dữ liệu ở tầng Device thu thập và gửi dữ liệu đến tầng Network. Từ đó, tầng Network gửi dữ liệu của thiết bị thu thập đến thiết bị khác để thực thi.

#### 2.2.2.2. Kiến trúc Device-to-Cloud

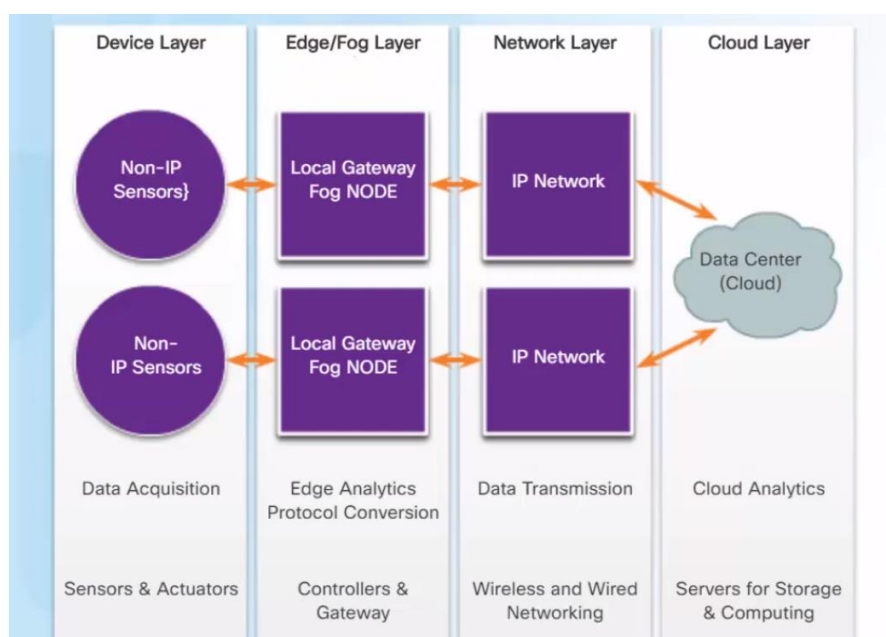


Hình 2.3: Mô hình kiến trúc Device-to-Cloud

So với kiến trúc Device-to-Device, kiến trúc này có thêm tầng đám mây, tầng này bao gồm các máy chủ dùng để lưu trữ, xử lý tính toán.

Tầng đám mây phân tích các dữ liệu từ tầng thiết bị gửi lên thông qua kênh mạng. Dữ liệu đã được xử lý được gửi ngược lại cho các thiết bị khác ở tầng thiết bị để thực thi thông qua kênh mạng.

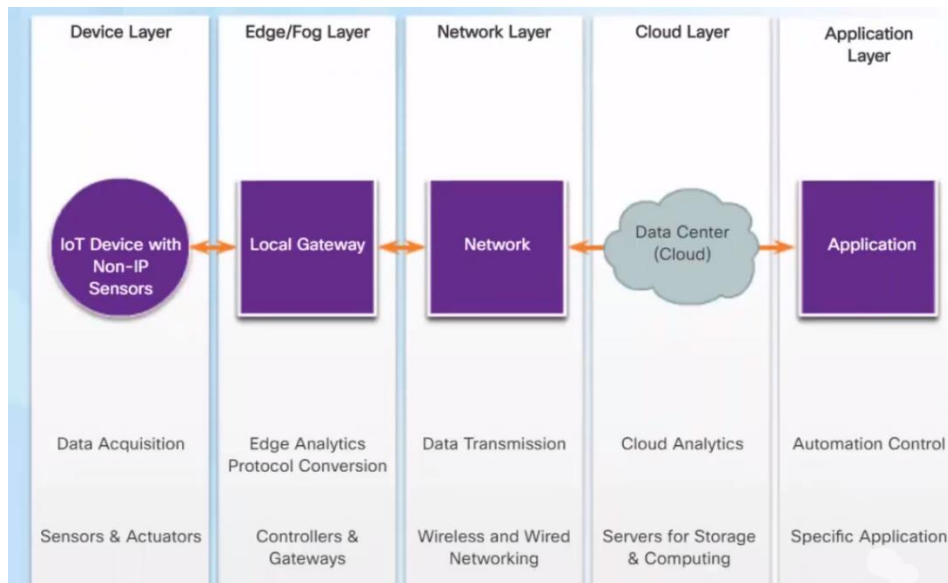
### 2.2.2.3. Kiến trúc Device-to-Gateway-to-Cloud FOG



Hình 2.4: Mô hình kiến trúc Device-to-Gateway-to-Cloud FOG

So với kiến trúc Device-to-Cloud, kiến trúc này có thêm tầng biên hay tầng sương mù (Edge/Fog Layer), tầng này bao gồm các bộ điều khiển (Controllers) và các Gateway để phục vụ cho việc tính toán cục bộ cũng như chuyển đổi giao thức giữa những cảm biến không sử dụng giao thức IP để giao tiếp mạng.

#### 2.2.2.4. Kiến trúc Device-to-Gateway-to-Cloud-to Application



Hình 2.5: Mô hình kiến trúc Device-to-Gateway-to-Cloud-to-Application

Đây là mô hình kiến trúc đầy đủ nhất và được áp dụng vào các hệ thống IoT cho đến hiện tại.

Mô hình kiến trúc có sự xuất hiện thêm của tầng ứng dụng (Application Layer) cho phép kiểm soát thông tin và điều khiển của người dùng.

### 2.3. GPS

#### 2.3.1. Khái niệm GPS

GPS là hệ thống định vị toàn cầu do Mỹ phát triển và vận hành. GPS là tên viết tắt của cụm từ “Global Positioning System”. Nó là một hệ thống bao gồm nhiều vệ tinh bay trên quỹ đạo phía trên trái đất ở độ cao 20.200 km.

GPS hoạt động trong mọi điều kiện thời tiết, mọi nơi trên Trái Đất, liên tục suốt 24 giờ và hoàn toàn miễn phí đối với một số dịch vụ.

#### 2.3.2. Nguyên lý hoạt động

Các vệ tinh GPS bay vòng quanh Trái Đất hai lần trong một ngày theo một quỹ đạo rất chính xác và phát tín hiệu có thông tin xuống Trái Đất. Các máy thu

GPS nhận thông tin này và bằng phép tính lượng giác tính được chính xác vị trí của người dùng.

Bản chất của GPS là so sánh thời gian tín hiệu được phát đi từ vệ tinh với thời gian nhận được chúng. Độ sai lệch thời gian cho biết máy thu GPS cách vệ tinh bao xa, với nhiều quỹ đạo đo được tới nhiều vệ tinh máy thu có thể tính được vị trí của người dùng và hiển thị lên bản đồ điện tử của máy.

Để tính ra được vị trí 2 chiều (kinh độ và vĩ độ) thì máy thu phải nhận được tín hiệu ít nhất là 3 vệ tinh, với ít nhất 4 vệ tinh thì có thể tính được vị trí 3 chiều (kinh độ, vĩ độ và độ cao). Một khi vị trí người dùng đã tính được thì máy thu GPS có thể tính các thông tin khác như tốc độ, hướng chuyển động, bám sát di chuyển, khoảng hành trình,...

## **2.4. Web Server trên mạng cục bộ**

### **2.4.1. HyperText Transfer Protocol (HTTP)**

HTTP (HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản) là một trong các giao thức chuẩn về mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client), là giao thức Client/Server dùng cho World Wide Web – WWW HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho Internet).

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. HTTP là một stateless protocol. Hay nói cách khác, request hiện tại không biết những gì đã hoàn thành trong request trước đó. HTTP cho phép tạo các yêu cầu gửi và nhận các kiểu dữ liệu, do đó cho phép xây dựng hệ thống độc lập với dữ liệu được truyền giao.

HTTP Request Method: Là phương thức để chỉ ra hành động mong muốn được thực hiện trên tài nguyên đã xác định.

- GET: được sử dụng để lấy lại thông tin từ Server đã cung cấp bởi sử dụng một URL đã cung cấp. Các yêu cầu sử dụng GET nên chỉ nhận dữ liệu và nên không có ảnh hưởng gì tới dữ liệu.
- HEAD: Tương tự như GET, nhưng nó truyền tải dòng trạng thái và khu vực Header.
- POST: Một yêu cầu POST được sử dụng để gửi dữ liệu tới Server, ví dụ, thông tin khách hàng, file tải lên, ..., bởi sử dụng các mẫu HTML.
- PUT: Thay đổi tất cả các đại diện hiện tại của nguồn mục tiêu với nội dung được tải lên.
- DELETE: Gỡ bỏ các đại diện hiện tại của nguồn mục tiêu bởi URI.
- CONNECT: Thiết lập một tunnel tới Server được xác định bởi URI đã cung cấp.
- OPTIONS: Miêu tả các chức năng giao tiếp cho nguồn mục tiêu.
- TRACE: Trình bày một vòng lặp kiểm tra thông báo song song với path tới nguồn mục tiêu.

#### 2.4.2. Nền tảng NodeJS

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

Các đặc tính của NodeJS:

- Không đồng bộ: Tất cả các API của NodeJS đều không đồng bộ (*non-blocking*), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi

và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).

- Chạy rất nhanh: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
- Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
- Không đệm: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
- Có giấy phép: NodeJS đã được cấp giấy phép bởi MIT License.

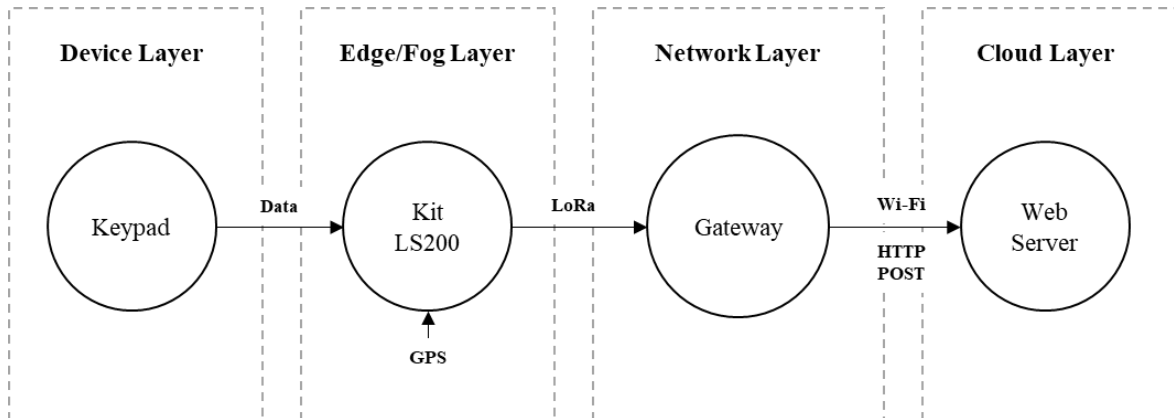
Hai NodeJS framework sử dụng phổ biến:

- ExpressJS là một trong những framework phổ biến dùng để xây dựng API và Website phổ biến nhất của NodeJS. Nó được sử dụng rộng rãi hầu như mọi dự án Web nào đều bắt đầu bằng việc tích hợp Express.
- SocketIO là một web-socket framework có sẵn cho nhiều ngôn ngữ lập trình. Trong NodeJS, SocketIO cho phép xây dựng một các ứng dụng realtime như chatbot, tickers, dashboard APIs, và nhiều ứng dụng khác. SocketIO có lợi ích hơn so với NodeJS thông thường.



## Chương 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 3.1. Sơ đồ hệ thống



Hình 3.1: Sơ đồ theo kiến trúc Device-to-Gateway-to-Cloud-to-Application

Kit LS 200 sẽ thu thập tín hiệu GPS và tín hiệu được nhập vào từ keypad. Sau đó gửi đến UCA Education Board thông qua sóng LoRa. Tại UCA Board chuyển tiếp gói tin qua ESP32-CAM bằng cách truyền nối tiếp UART. ESP32-CAM sẽ tham gia vào mạng nội bộ và gửi gói tin lên cho Server thông qua phương thức POST của giao thức truyền thông HTTP. Cuối cùng dữ liệu được gửi lên Server và được lưu vào cơ sở dữ liệu, sau đó được trình bày một cách trực quan ở giao diện người dùng tại Web Server.

### 3.2. Sử dụng DK Kit để gửi tín hiệu LoRa

#### 3.2.1. Cấu hình chi tiết của DK Kit

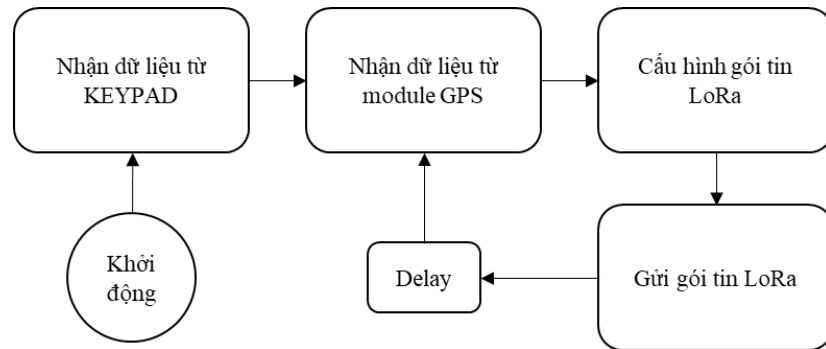


*Hình 3.2: RFThings Design Kit LS200*

Thiết bị đầu cuối giao tiếp tích hợp bộ thu phát LP-WAN (SX1262) . Ứng dụng theo dõi năng lượng thấp với định vị GPS cực nhanh và liên lạc tầm xa. Bo mạch bao gồm một header để kết nối bất kỳ cảm biến và thành phần bổ sung.

- MCU: 256 Kb Flash
- GPS: Quectel L96
- Frequency: 868, 915, 923 MHz

### 3.2.2. Các bước gửi tín hiệu LoRa



*Hình 3.3: Các bước gửi dữ liệu thông qua phương thức LoRa*

Khi khởi động DK, thiết bị sẽ tự động thu thập tín hiệu GPS. Sau đó, người dùng sẽ chọn chức năng phù hợp. Sau khi lấy được kinh độ, vĩ độ và chức năng theo yêu cầu của người dùng, thiết bị sẽ tiến hành xử lý dữ liệu cho đúng chuẩn và cấu hình gói tin LoRa sau đó gửi đi liên tục theo chu kỳ.

## 3.3. Tạo Gateway bằng ESP32-CAM và UCA Education Board

### 3.3.1. UCA Education Board



*Hình 3.4: UCA Education Board*

UCA Education Board phiên bản 3.8 là bo mạch được thiết kế bởi RFThings sử dụng chip ATmega328PB. Board được phát triển cho mục đích giáo dục để hỗ trợ các khóa học về hệ thống nhúng, điện tử số và tương tự, viễn thông và IoT. Trên board có tích hợp sẵn module RFM95 để thu phát tín hiệu LoRa.

### 3.3.2. ESP32-CAM

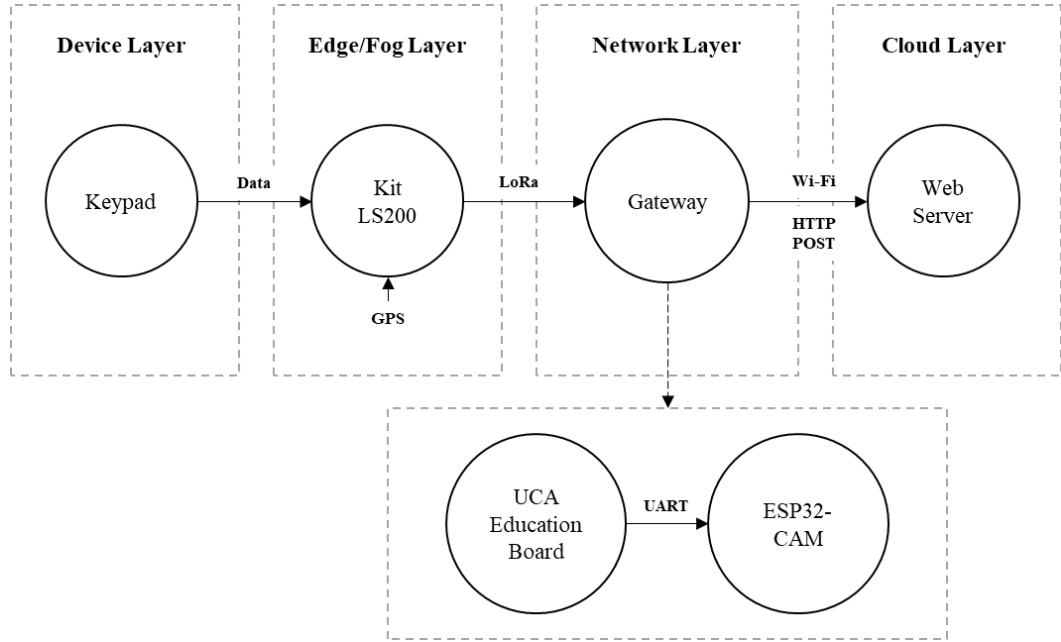


*Hình 3.5: ESP32-CAM*

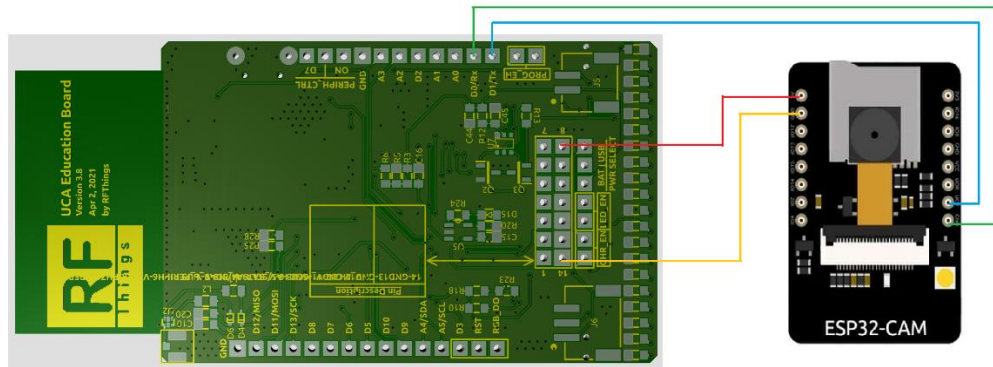
ESP32-CAM là mạch tích hợp với bộ xử lý chính là module ESP32. Ngoài kết nối wifi và bluetooth, ESP32-CAM được thiết kế để có thể gắn Module Camera để thực hiện các dự án với thu thập hình ảnh, stream video. ESP32-CAM không nạp được chương trình trực tiếp mà phải thông qua một board khác.

### 3.3.3. Tạo Gateway bằng ESP32-CAM và UCA Board

ESP32-CAM được kết nối với UCA Board bằng chuẩn giao tiếp UART. Khi nhận được gói tin LoRa từ DK, UCA Board sẽ tiến hành xử lý gói tin để lấy được dữ liệu cần thiết và truyền nó qua cổng UART đến ESP32-CAM.

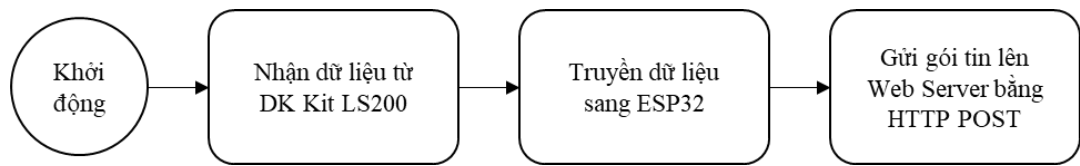


Hình 3.6: Mối liên kết của ESP32-CAM và UCA với hệ thống



Hình 3.7: Ghép nối UART giữa UCA và ESP32-CAM

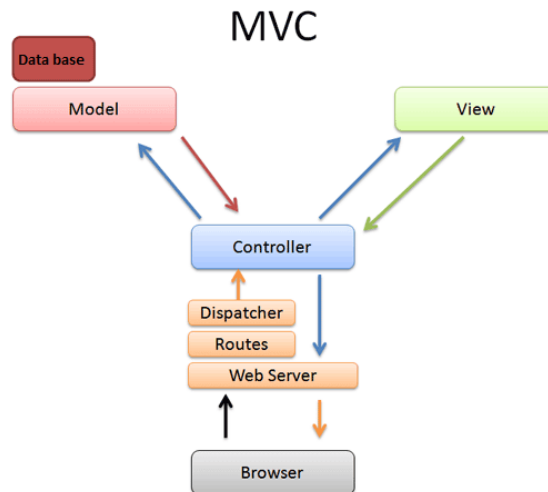
Với khả năng kết nối với Wifi, ESP32-CAM sẽ kết nối với Server bằng địa chỉ IP. Sau khi nhận dữ liệu từ UCA Board, tiến hành xử lý và gửi dữ liệu lên Server theo chuẩn JSON bằng phương thức post của giao thức HTTP.



Hình 3.8: Sơ đồ hoạt động của Gateway

### 3.4. Tạo Web Server cục bộ bằng NodeJS

#### 3.4.1. Mô hình MVC



Hình 3.9: Mô hình MVC

MVC là từ viết tắt của '**Model View Controller**'. Nó đại diện cho các nhà phát triển kiến trúc áp dụng khi xây dựng các ứng dụng. Với kiến trúc MVC, chúng ta xem xét cấu trúc ứng dụng liên quan đến cách luồng dữ liệu của ứng dụng của chúng ta hoạt động như thế nào.

Mô hình MVC được chia làm 3 lớp xử lý gồm Model – View – Controller :

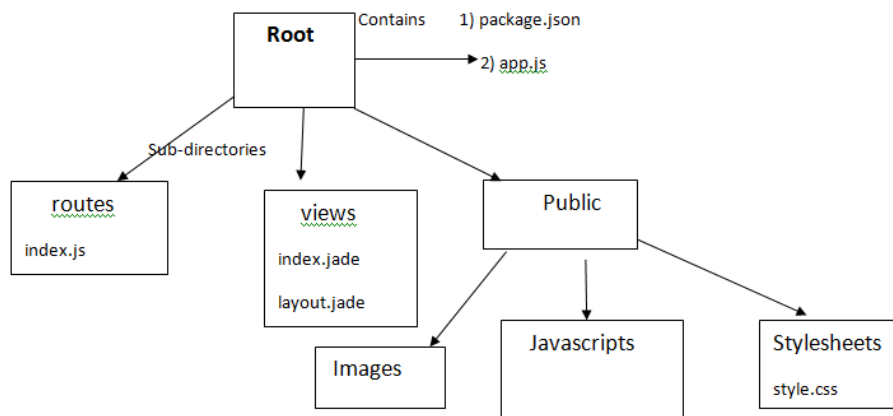
- **Model:** là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu (mysql, mssql... ); nó sẽ bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu...

- **View** : là nơi chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh... nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.
- **Controller** : là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.

Sự tương tác giữa các phần:

- Controller tương tác với qua lại với View.
- Controller tương tác qua lại với Model.
- Model và View không có sự tương tác với nhau mà nó tương tác với nhau thông qua Controller.

### 3.4.2. Express framework



Hình 3.10: Tổ chức thư mục cơ bản

Express là một framework giành cho nodejs. Nó cung cấp cho chúng ta rất nhiều tính năng mạnh mẽ trên nền tảng web cũng như trên các ứng dụng di động. Express hỗ trợ các phương thức HTTP và middleware tạo ra một API vô cùng mạnh mẽ và dễ sử dụng. Có thể tổng hợp một số chức năng chính của express như sau:

- Thiết lập các lớp trung gian để trả về các HTTP request.

- Định nghĩa router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số.

### 3.4.3. Cơ sở dữ liệu - MongoDB

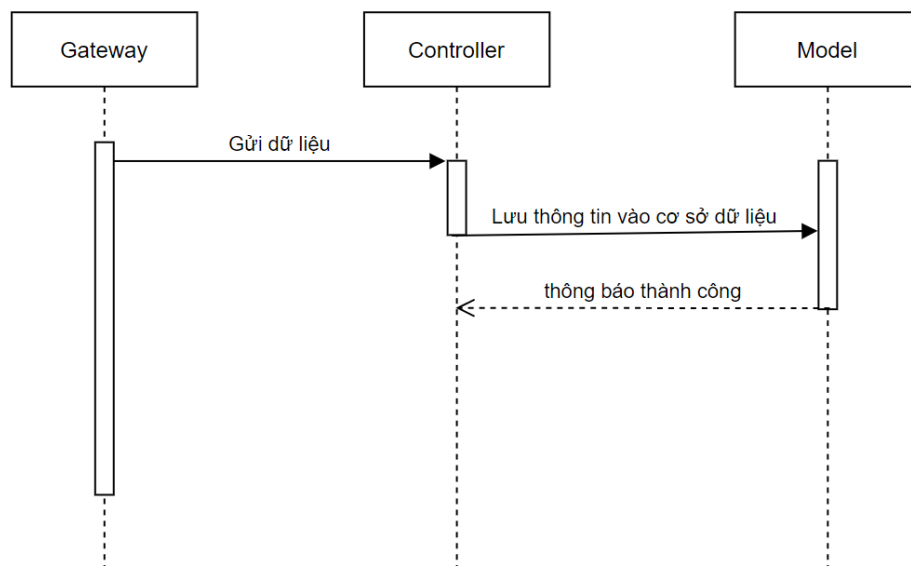
MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

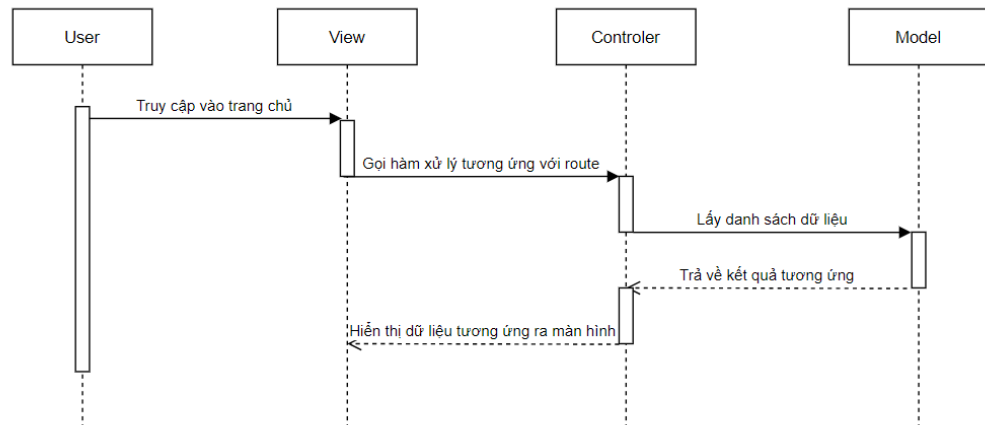
### 3.4.4. Sơ đồ hệ thống Web Server



Hình 3.11: Quá trình gửi dữ liệu lên web server



Gateway bắt đầu dữ liệu lên server theo một địa chỉ nhất định của server. Khi nhận được request server sẽ tiến hành gọi đến hàm tạo đối tượng location. Trong hàm tạo location, tạo một thể hiện của model location theo dữ liệu nhận được sau đó lưu vào cơ sở dữ liệu và thông báo thành công.



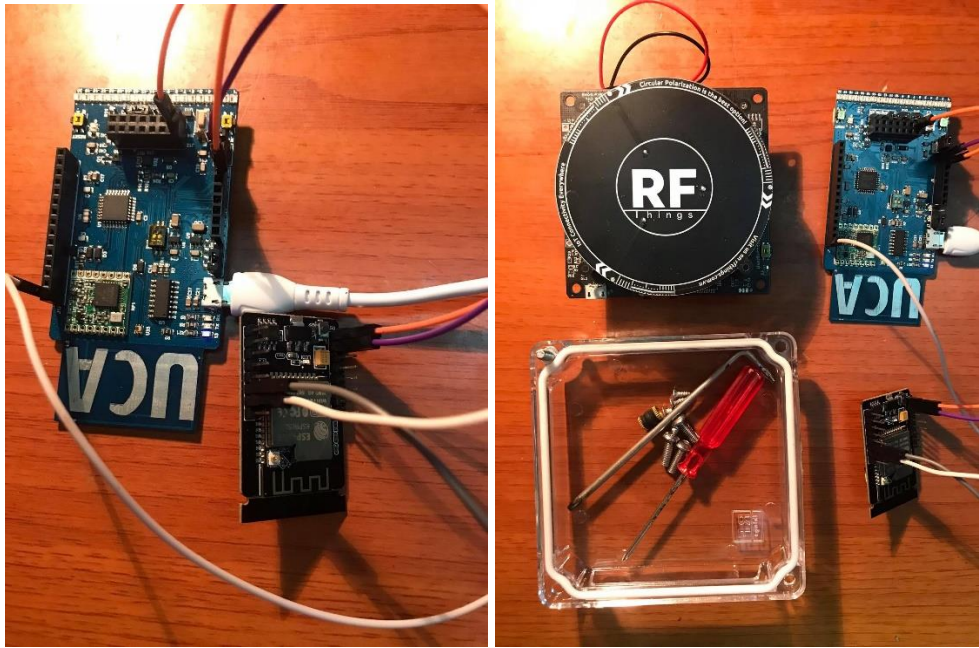
Hình 3.12: Hiện thị giao diện người dùng

Khi người dùng truy cập vào trang chủ, server gọi đến hàm index ở siteController. Tại đây, Controller lấy danh sách dữ liệu thông qua Model và trả về view kèm theo dữ liệu để hiện thị trên giao diện người dùng.

### 3.5. Ghép nối hệ thống

Sau khi nạp chương trình, RFThings Design Kit LS200 sẽ được hoạt động độc lập không cần nối dây.

UCA Design Kit sẽ được kết nối UART với ESP32-CAM sau khi đã được nạp chương trình. Đóng vai trò như một Gateway, 2 thiết bị được cấp nguồn, ESP32-CAM được kết nối với Wi-Fi để truyền dữ liệu lên Web Server và giữ cố định tại 1 vị trí nhất định, để bắt tín hiệu.



Hình 3.13: Gateway và toàn bộ thiết bị sau khi ghép nối hoàn tất

## Chương 4. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ

### 4.1. Phạm vi hoạt động

Thiết bị khởi tạo và chạy với các thông số sau:

- Spreading Factor 7 (SF7)
- Bandwidth 500kHz
- Coding Rate 2

Với thử nghiệm thực tế, do tình hình dịch bệnh nên nhóm không có điều kiện kiểm tra truyền dữ liệu ở khoảng cách xa hơn mà chỉ truyền được trong phạm vi xa nhất có thể là 100m.

### 4.2. Giám sát thời gian truyền gói tin

Thời gian truyền gói tin phụ thuộc vào khoảng cách của 2 thiết bị đầu cuối và thời gian để lấy dữ liệu GPS.

Đối với thời gian để lấy dữ liệu từ GPS:

Khoảng thời gian của lần lấy dữ liệu GPS đầu tiên (sau khi khởi động thiết bị lên) rơi vào khoảng từ 2 phút đến 2 phút 30 giây.

Các lần tiếp theo thời gian lấy dữ liệu GPS nhanh hơn, từ 10 giây trở lên.

Đối với khoảng cách của 2 thiết bị đầu cuối:

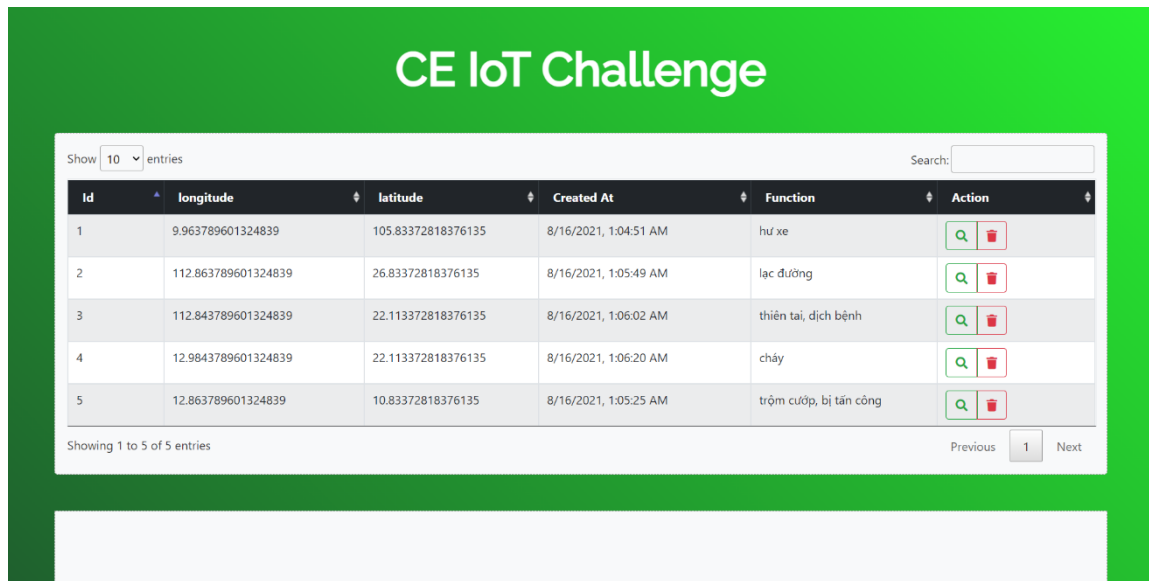
Nếu để 2 thiết bị đầu cuối trong phạm vi 1m, thời gian từ lúc gửi gói tin cho đến khi dữ liệu được hiển thị trên Web Server gần như tức thời, nằm trong khoảng từ 1 đến 10 giây. Rất ít khi có xảy ra hiện tượng mất gói tin.

Nếu để 2 thiết bị đầu cuối trong phạm vi 5-10m, thời gian từ lúc gửi gói tin cho đến khi dữ liệu được hiển thị trên Web Server là từ 2 đến 5 phút. Có thể thấy hiện tượng mất gói tin xảy ra thường xuyên hơn dẫn đến thời gian nhận được tăng lên.











Nếu để 2 thiết bị đầu cuối trong phạm vi 50-100m, thời gian từ lúc gửi gói tin cho đến khi dữ liệu được hiển thị trên Web Server là từ 10 đến 20 phút.

Vậy nếu khoảng cách càng xa thì thời gian nhận được gói tin càng tăng.

### 4.3. Giao diện người dùng

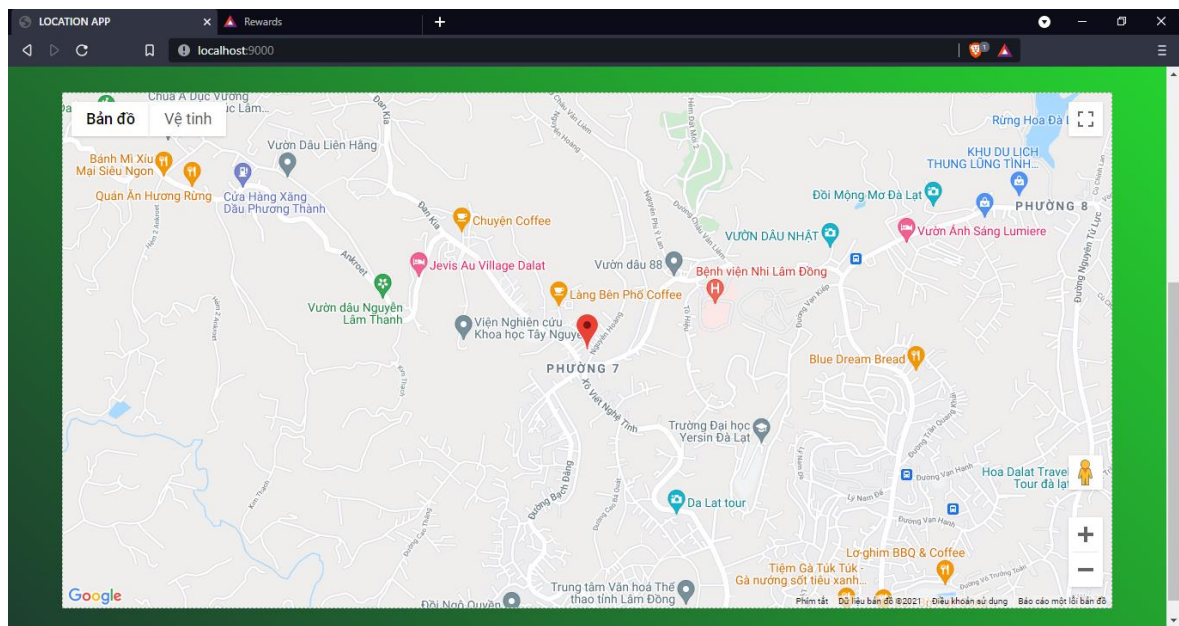


The screenshot shows a web application titled "CE IoT Challenge" with a green header. Below the header is a table displaying emergency cases. The table has columns for Id, longitude, latitude, Created At, Function, and Action. There are 5 entries shown. Above the table, there is a "Show 10 entries" dropdown and a "Search:" input field. Below the table, there is a "Showing 1 to 5 of 5 entries" message and pagination controls with "Previous", "1", and "Next" buttons.

Id	longitude	latitude	Created At	Function	Action
1	9.963789601324839	105.83372818376135	8/16/2021, 1:04:51 AM	hư xe	 
2	112.863789601324839	26.83372818376135	8/16/2021, 1:05:49 AM	lạc đường	 
3	112.843789601324839	22.113372818376135	8/16/2021, 1:06:02 AM	thiên tai, dịch bệnh	 
4	12.9843789601324839	22.113372818376135	8/16/2021, 1:06:20 AM	cháy	 
5	12.863789601324839	10.83372818376135	8/16/2021, 1:05:25 AM	trộm cướp, bị tấn công	 

Hình 4.1: Bảng quản lý các trường hợp cần cứu hộ

- Sử dụng cơ chế phân trang để quản lý dữ liệu một cách chính xác và trực quan hơn với các trường:
  - Id: dùng để thể hiện Id của thiết bị.
  - longitude, latitude: dùng để thể hiện kinh độ và vĩ độ.
  - Created At: hiển thị thời gian dữ liệu được gửi lên server.
  - Function: dùng để xác định trường hợp cần cứu hộ.
  - Action: gồm 2 button là View để hiển thị vị trí cụ thể của người cần cứu hộ trong bản đồ của google phía dưới. Delete dùng để xóa dữ liệu của thiết bị khỏi cơ sở dữ liệu khi đã cứu hộ.
- Tính năng Search giúp tìm kiếm được loại hình cần cứu hộ hoặc thiết bị cần cứu hộ.
- Tính năng show entries có chức năng hiển thị số dòng đã chọn.



## Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

### 5.1. Kết luận

Do tình hình dịch bệnh nên nhóm phải làm trực tuyến và gặp phải một số khó khăn. Cuối cùng sau quá trình nghiên cứu, tìm tòi, học hỏi, nhóm cơ bản đã demo thành công được đề tài “*Hệ thống xác định yêu cầu cứu hộ trên diện rộng sử dụng giao thức LoRa*”. Hệ thống đã hoạt động như kế hoạch đưa ra ban đầu, dữ liệu đã được truyền bằng LoRa và hiển thị lên trên Server.

Tuy nhiên do một số khó khăn về khoảng cách cũng như thời gian, đề tài của nhóm vẫn có một số hạn chế chưa kịp hoàn thành và cải thiện như chưa tạo được keypad và phải sử dụng dữ liệu ảo tự nhập vào từ cổng Serial, vẫn xảy ra trường hợp mất gói tin LoRa khi truyền.

### 5.2. Hướng phát triển đề tài

Hiện thực hóa thiết bị nhỏ gọn có thể luôn mang theo bên mình để thực hiện đúng chức năng của nó là cứu hộ.

Xây dựng hệ thống web, ứng dụng di động bằng việc tối ưu hóa hệ thống quản lý dữ liệu. Bên cạnh đó làm cho giao diện bắt mắt, thân thiện hơn và thêm các tính năng để phục vụ tối đa nhu cầu của người dùng.

Phát triển các cơ chế bảo mật, bảo vệ người dùng tránh việc theo dõi.

Cập nhật phương thức truyền dữ liệu theo đúng chuẩn LoRaWAN thay vì chỉ sử dụng LoRa thường như hiện tại.

## TÀI LIỆU THAM KHẢO

- [1] “Công nghệ LoRa và LoRaWAN truyền dữ liệu xa, công suất thấp trong IoT,” [Trực tuyến]. Available: <https://cnttshop.vn/blogs/tin-tuc/cong-nghe-lora-va-lorawan>. [Đã truy cập 21 7 2021].
- [2] “Công nghệ LoRaWAN, thời đại mới của IoT,” [Trực tuyến]. Available: <https://www.daviteq.com/blog/vi/cong-nghe-lorawan-thoi-dai-moi-cua-iot/>. [Đã truy cập 28 7 2021].
- [3] “Tìm hiểu về LoRa và cách hoạt động,” [Trực tuyến]. Available: <https://vidieukhien.xyz/2018/07/03/tim-hieu-ve-lora-va-cach-hoat-dong/>. [Đã truy cập 21 7 2021].
- [4] “Kiến trúc IoT,” [Trực tuyến]. Available: <http://uommamcongnghe.org/cong-nghe/kien-truc-iot.html>.
- [5] T. VINSYS, “Phân loại các kiểu kiến trúc hệ thống IoT,” [Trực tuyến]. Available: <https://youtu.be/i4BZXT2MySM>.
- [6] S. Kulkarni, “Communication Models in Internet of Things: A Survey,” [Trực tuyến]. Available: <http://www.ijste.org/articles/IJSTEV3I11049.pdf>.
- [7] T. Holowaychuk, “Express - NPM,” [Trực tuyến]. Available: <https://www.npmjs.com/package/express>. [Đã truy cập 12 8 2021].
- [8] TAPIT, “Tổng quan về giao thức HTTP, phương thức truyền tải dữ liệu giữa client-server,” [Trực tuyến]. Available: <https://tapit.vn/http-request-va-http-response-phuong-thuc-giao-tiep-giua-server-client/>. [Đã truy cập 12 8 2021].
- [9] Alius, “Lập trình MVC với Razor view engine,” [Trực tuyến]. Available: <https://m21love.blogspot.com/2012/07/lap-trinh-mvc-voi-razor-view-engine.html>. [Đã truy cập 13 8 2021].

- [10] “Expressjs Là Gì? Tất tần tật về khái niệm Express,” [Trực tuyến]. Available: <https://vccidata.com.vn/expressjs-la-gi/>. [Đã truy cập 13 8 2021].
- [11] thanhtv.96, “MongoDB là gì? Cơ sở dữ liệu phi quan hệ,” [Trực tuyến]. Available: <https://viblo.asia/p/mongodb-la-gi-co-so-du-lieu-phi-quan-he-bJzKmgol9N>. [Đã truy cập 14 8 2021].