### **Hệ thống Giám sát và Bảo vệ Rừng thông minh**

Hệ thống này ứng dụng công nghệ IoT để giám sát và bảo vệ rừng khỏi nguy cơ cháy và nạn chặt phá trái phép. Hệ thống tích hợp các cảm biến môi trường, âm thanh, GPS và khả năng điều khiển tự động, truyền dữ liệu về server để hiển thị trực quan trên giao diện web.

**1. Kiến trúc Hệ thống**

Hệ thống được cấu thành từ 4 thành phần chính:

* Thiết bị phần cứng (ESP32): Thu thập dữ liệu cảm biến (nhiệt độ, độ ẩm, âm thanh, GPS) và điều khiển bơm nước.
* Server Backend (Django REST Framework Channels): Nhận dữ liệu từ ESP32, quản lý cơ sở dữ liệu, xử lý các yêu cầu từ Frontend và Server AI.
* Server AI (Flask): Nhận dữ liệu âm thanh từ Backend, xử lý phân tích và nhận diện tiếng cưa, gửi kết quả về Backend.
* Frontend (ReactJS): Giao diện người dùng trực quan, hiển thị dữ liệu real-time (biểu đồ, bản đồ GPS), hiển thị cảnh báo và cho phép phát lại âm thanh.

**2. Yêu cầu hệ thống**

Để cài đặt và chạy hệ thống, bạn cần chuẩn bị các môi trường sau:

* **Phần cứng**
* ESP32
* Cảm biến nhiệt độ - độ ẩm DHT22
* Cảm biến âm thanh INMP441 I2S Omnidirectional Microphone
* Module GPS NEO 6M
* Anten GPS 1575.42Mhz SMA
* Module relay 5V 1 kênh
* Động cơ bơm nước R385
* Các linh kiện điện tử phụ trợ (điện trở, dây dẫn, breadboard...)
* Nguồn điện
* **Phần mềm**
* Arduino IDE (cho code ESP32)
* Python 3.x (khuyến nghị 3.8+)
* Node.js & npm (khuyến nghị Node 14+, npm 6+)
* pip (Trình quản lý gói của Python)
* Git (Để clone repository)
* MongoDB (Hệ quản trị cơ sở dữ liệu
* Redis (Đối với Django Channels)

**3. Hướng dẫn cài đặt**

Thực hiện theo các bước sau để thiết lập từng thành phần của hệ thống.

**3.1. Cài đặt Code Arduino (ESP32)**

* Cài đặt Arduino IDE
* Tải và cài đặt Arduino IDE từ trang chủ: [arduino.cc](https://www.arduino.cc/).
* Thêm Board ESP32
  + Trong Arduino IDE, vào File > Preferences.
  + Trong trường Additional Boards Manager URLs, thêm đường dẫn sau:

<https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json>

* Vào Tools > Board > Boards Manager. Tìm kiếm và cài đặt esp32.

**3.2 Cài đặt thư viện:**

* Vào Sketch > Include Library > Manage Libraries.
* Tìm kiếm và cài đặt các thư viện sau:
  + DHT sensor library (bởi Adafruit)
  + TinyGPSPlus (bởi Mikal Hart)
  + ArduinoWebsockets (bởi Links\_Dev)

**3.3 Cấu hình Code**

* Mở file `rainforest\_arduino.ino` trong Arduino IDE.
* Cập nhật các thông tin cấu hình mạng và server tại phần đầu file:
  + const char\* ssid = "aaasss"; // Tên mạng WiFi của bạn
  + const char\* password = "asdasd123"; // Mật khẩu WiFi của bạn
  + const char\* server\_host = "192.168.141.28"; // Địa chỉ IP hoặc Domain của Server Backend
  + const uint16\_t server\_port = 8000; // Cổng của Server Backend
* Đảm bảo các chân GPIO cho cảm biến, I2S (cho âm thanh), GPS, relay và động cơ servo được định nghĩa chính xác trong code để khớp với đấu nối phần cứng của bạn.
* Lưu lại file.
* Upload Code
  + Chọn đúng board ESP32 và cổng COM (Tools > Board, Tools > Port).
* Nhấn nút Upload.

### **3.4. Cài đặt Server Backend (Django REST Framework Channels)**

* Tải Source Code

Nếu bạn có Git, hãy clone repository:  
git clone <https://github.com/IloveUhiuhiu/RainForest_BE.git>

Hoặc tải thư mục rainforest\_be từ source code đã được upload.

* Tạo và Kích hoạt Môi trường ảo

python -m venv venv

*Trên Windows:* .\venv\Scripts\activate

*Trên macOS/Linux:*source venv/bin/activate

* Cài đặt Thư viện Python: pip install -r requirements.txt
* Cấu hình Cơ sở dữ liệu (MongoDB):
  + Cài đặt và khởi chạy MongoDB server trên hệ thống của bạn.
  + Đảm bảo cấu hình kết nối MongoDB trong settings.py của Django.
* Cấu hình Redis:
  + Cài đặt và khởi chạy Redis server trên máy tính của bạn.
  + Đảm bảo CHANNEL\_LAYERS trong settings.py trỏ đúng đến Redis server của bạn.
* Chạy migration:  
  python manage.py makemigrations

python manage.py migrate

* Chạy Server Backend:

daphne -b 0.0.0.0 -p 8000 rainforest\_be.asgi:application

### **3.5. Cài đặt Server AI (Flask)**

* Tải Source Code:

Nếu bạn có Git, hãy clone repository:  
git clone https://github.com/IloveUhiuhiu/RainForest\_AI.git

Hoặc tải thư mục rainforest\_ai từ source code của bạn.

* Tạo và Kích hoạt Môi trường ảo:  
  python -m venv venv

*Trên Windows:*.\venv\_ai\Scripts\activate

*Trên macOS/Linux:*source venv\_ai/bin/activate

* Cài đặt Thư viện Python:  
  pip install -r requirements.txt
* Chuẩn bị Mô hình AI:

Tải file best\_model.h5 và đặt vào thư mục weights từ <https://github.com/IloveUhiuhiu/RainForest_AI.git>, trong thư mục weights

Đảm bảo file mô hình AI đã được huấn luyện (best\_model.h5) nằm trong thư mục weight của ứng dụng Flask.

* Chạy Server AI:

python app.py

### **3.6. Cài đặt Frontend (ReactJS)**

* Tải Source Code:

Nếu bạn có Git, hãy clone repository:  
git clone https://github.com/pnhatthanh/RainForest\_FE.git

Hoặc tải từ source code của bạn.

* Cài đặt các thư viện:  
  npm install
* Cấu hình các Endpoint:

Cập nhật URL của Server Backend và Websocket URL cho Channels trong file config

API\_URL=[http://localhost:8000](http://localhost:8000/api/)

WS\_URL=ws://localhost:8000/ws

(Hãy thay localhost bằng địa chỉ IP hoặc domain thực tế của Server Backend nếu nó chạy trên máy khác.)

* Chạy Frontend:  
  npm start

## **4. Chạy và Kiểm tra Hệ thống**

## Đảm bảo Redis server đang chạy.

## Khởi chạy Server Backend (sử dụng daphne).

## Khởi chạy Server AI (sử dụng python [app.py](http://app.py)).

## Khởi chạy Frontend (sử dụng npm start).

## Kết nối và cấp nguồn cho thiết bị ESP32 (đã upload code).

## Kiểm tra giao diện web:

* Xác minh dữ liệu cảm biến (nhiệt độ, độ ẩm) có được cập nhật liên tục không.
* Kiểm tra vị trí GPS có hiển thị trên bản đồ không.
* Thử các kịch bản kiểm thử (tăng nhiệt độ/giảm độ ẩm, tạo tiếng cưa) để xem hệ thống có cảnh báo và kích hoạt bơm nước đúng không.
* Kiểm tra chức năng nghe lại âm thanh.