KẾ HOẠCH THỰC HIỆN DỰ ÁN CỘNG ĐỒNG: HỆ THỐNG GIÁM SÁT MÔI TRƯỜNG

# MỤC ĐÍCH

Xây dựng hệ thống giám sát môi trường thời gian thực sử dụng ESP32 và các cảm biến (MQ-135, DHT11, BMP280, AHT20, GP2Y1014AU0F) để đo chất lượng không khí, nhiệt độ, độ ẩm, áp suất, và bụi mịn. Hệ thống gửi dữ liệu lên cloud qua MQTT, hiển thị trên dashboard Grafana, và gửi cảnh báo khi chỉ số vượt ngưỡng (ví dụ: nhiệt độ > 35°C, bụi > 150 µg/m³). Dự án phục vụ trường học hoặc công viên công cộng, nâng cao nhận thức về môi trường và giáo dục IoT cho học sinh.

# THỜI GIAN THỰC HIỆN

Từ 1/5/2025 đến 30/6/2025 (8 tuần).

# THÀNH PHẦN THAM GIA

- Sinh viên: Thái Văn Hòa (vai trò: lập trình, thiết kế phần cứng, triển khai hệ thống).

# THIẾT BỊ

* ESP32 Dev Module: Vi điều khiển chính để đọc dữ liệu cảm biến và gửi qua MQTT.

Cảm biến:

* MQ-135: Đo chất lượng không khí (CO2, VOCs).
* DHT11: Đo nhiệt độ và độ ẩm (giá rẻ, phù hợp giáo dục).
* BMP280: Đo áp suất khí quyển (hỗ trợ dự báo thời tiết).
* AHT20: Đo nhiệt độ và độ ẩm (chính xác hơn DHT11).
* GP2Y1014AU0F: Đo bụi mịn (tương tự PM2.5).
* Raspberry Pi 4 (4GB): Chạy MQTT broker (Mosquitto), Node-RED, InfluxDB, Grafana.
* Phụ kiện: Dây nối, breadboard, tụ điện 150µF và điện trở 220Ω (cho GP2Y1014AU0F), nguồn điện 5V/3.3V.

# CHI PHÍ ƯỚC TÍNH

* ESP32: 100.000 VNĐ.
* MQ-135: 50.000 VNĐ.
* DHT11: 30.000 VNĐ.
* BMP280: 50.000 VNĐ.
* AHT20: 50.000 VNĐ.
* GP2Y1014AU0F: 100.000 VNĐ.
* Raspberry Pi 4 (8GB): 1.800.000 VNĐ.
* Phụ kiện (dây, tụ, điện trở): 50.000 VNĐ.
* Tổng: ~1.880.000 VNĐ.

# KẾ HOẠCH CHI TIẾT THEO TUẦN

## Giai Đoạn 1: Chuẩn Bị và Thiết Kế (Tuần 1-2, 1/5 - 14/5/2025)

Mục tiêu: Chuẩn bị thiết bị, môi trường phát triển, thiết kế sơ đồ hệ thống.

### Tuần 1 (1-7/5/2025):

* Tìm hiểu tài liệu về ESP32, Raspberry Pi, và các cảm biến (MQ-135, DHT11, BMP280, AHT20, GP2Y1014AU0F) qua datasheet, GitHub, YouTube.
* Nghiên cứu giao thức MQTT, Node-RED, InfluxDB, Grafana để hiểu luồng dữ liệu từ cảm biến đến cloud.
* Lập danh sách thiết bị cần mua (ESP32, cảm biến, Raspberry Pi, phụ kiện).
* Tìm nhà cung cấp uy tín (Shopee, Linh Kiện Việt, hoặc cửa hàng điện tử), so sánh giá, đặt hàng online.
* Tải và cài đặt Arduino IDE, thêm hỗ trợ ESP32 qua Board Manager.
* Cài đặt thư viện Arduino: PubSubClient (MQTT), DHT (DHT11), Adafruit\_BMP280 (BMP280), AHTxx (AHT20).
* Xác định địa điểm triển khai (ví dụ: sân trường cấp 2 gần nhà hoặc công viên nhỏ trong khu phố).
* Ghi chú các yếu tố môi trường tại địa điểm (nhiệt độ, độ ẩm, nguồn bụi) để chọn vị trí lắp đặt.

### Tuần 2 (8-14/5/2025):

* Nhận thiết bị từ nhà cung cấp, kiểm tra chất lượng (đảm bảo ESP32, cảm biến, Raspberry Pi không hư hỏng).
* Đọc datasheet của MQ-135, DHT11, BMP280, AHT20, GP2Y1014AU0F để hiểu các chân kết nối, điện áp, và yêu cầu hiệu chuẩn.
* Cài đặt Raspberry Pi OS (dùng Raspberry Pi Imager) trên thẻ SD.
* Cài đặt Mosquitto (sudo apt update && sudo apt install mosquitto mosquitto-clients) và kiểm tra dịch vụ (sudo systemctl status mosquitto).
* Cài đặt Node-RED
* Kiểm tra kết nối mạng WiFi trên ESP32 và Raspberry Pi, ghi lại IP tĩnh của Raspberry Pi để dùng cho MQTT.

## Giai Đoạn 2: Phát Triển Hệ Thống (Tuần 3-5, 15/5 - 4/6/2025)

Mục tiêu: Lập trình ESP32, tích hợp cảm biến, cấu hình server Raspberry Pi.

### Tuần 3 (15-21/5/2025):

* Kết nối phần cứng theo sơ đồ mạch trên breadboard, đảm bảo các chân GPIO không bị xung đột.
* Kiểm tra từng cảm biến bằng code Arduino đơn giản (ví dụ: đọc nhiệt độ DHT11, áp suất BMP280 qua Serial Monitor).
* Viết code ESP32 để đọc dữ liệu từ tất cả cảm biến:
* MQ-135: Đọc analog (GPIO 34).
* DHT11: Nhiệt độ, độ ẩm (GPIO 4).
* BMP280: Áp suất (I2C).
* AHT20: Nhiệt độ, độ ẩm (I2C).
* GP2Y1014AU0F: Đọc bụi mịn (GPIO 35, điều khiển LED GPIO 32).
* In dữ liệu cảm biến ra Serial Monitor để kiểm tra tính chính xác.
* Kiểm tra lỗi cảm biến (ví dụ: giá trị NaN từ DHT11, tín hiệu không ổn định từ GP2Y1014AU0F).
* Điều chỉnh code để xử lý giá trị bất thường (gán -1 nếu lỗi).
* Lưu code ESP32 vào file .ino, ghi chú các thông số (GPIO, thư viện).

### Tuần 4 (22-28/5/2025):

* Tích hợp giao thức MQTT vào code ESP32, sử dụng PubSubClient để gửi dữ liệu đến các topic:
* env/temperature\_dht, env/humidity\_dht (DHT11).
* env/temperature\_aht, env/humidity\_aht (AHT20).
* env/air\_quality (MQ-135).
* env/pressure (BMP280).
* env/dust (GP2Y1014AU0F).
* Cấu hình địa chỉ MQTT server (IP của Raspberry Pi), port 1883.
* Tạo flow Node-RED trên Raspberry Pi:
* Thêm node mqtt in cho mỗi topic, kiểm tra dữ liệu nhận được.
* Lưu dữ liệu tạm vào file CSV (dùng node file) để kiểm tra.
* Kiểm tra kết nối MQTT giữa ESP32 và Raspberry Pi qua mạng WiFi.
* Debug lỗi MQTT (ví dụ: mất kết nối, dữ liệu không đồng bộ).
* Thêm cơ chế reconnect tự động trong code ESP32 nếu WiFi/MQTT ngắt.
* Kiểm tra toàn bộ luồng dữ liệu từ cảm biến → ESP32 → Node-RED, đảm bảo dữ liệu hiển thị đúng trong Node-RED.

### Tuần 5 (29/5-4/6/2025):

* Cài đặt InfluxDB trên Raspberry Pi (sudo apt update && sudo apt install influxdb).
* Tạo database trong InfluxDB (dùng lệnh influx → CREATE DATABASE env\_data).
* Cài node node-red-contrib-influxdb trong Node-RED, cấu hình lưu dữ liệu từ các topic MQTT vào InfluxDB.
* Cài đặt Grafana trên Raspberry Pi (sudo apt install grafana).
* Kết nối Grafana với InfluxDB, tạo datasource.
* Tạo dashboard Grafana với các panel:
* Nhiệt độ (DHT11, AHT20).
* Độ ẩm (DHT11, AHT20).
* Chất lượng không khí (MQ-135).
* Áp suất (BMP280).
* Bụi mịn (GP2Y1014AU0F).
* Kiểm tra dữ liệu hiển thị trên dashboard, điều chỉnh định dạng (đơn vị, màu sắc).

## Giai Đoạn 3: Kiểm Tra và Tối Ưu (Tuần 6-7, 5/6 - 18/6/2025)

Mục tiêu: Hiệu chuẩn cảm biến, thêm cảnh báo, tối ưu hệ thống.

### Tuần 6 (5-11/6/2025):

* Hiệu chuẩn cảm biến:
* MQ-135: Đặt trong môi trường sạch (không khói, hóa chất), so sánh với giá trị tham chiếu từ datasheet.
* GP2Y1014AU0F: Đặt trong môi trường không bụi, tránh ánh sáng mạnh, ghi baseline (0 µg/m³).
* BMP280: Đặt ở nơi thông thoáng, so sánh áp suất với dữ liệu thời tiết địa phương.
* AHT20, DHT11: Đặt trong phòng ổn định (24 giờ), so sánh giá trị để kiểm tra độ chính xác.
* Tạo flow Node-RED để kiểm tra ngưỡng và gửi cảnh báo:
* Nhiệt độ > 35°C (DHT11 hoặc AHT20).
* Độ ẩm > 80%.
* Chất lượng không khí > 1000 ppm (MQ-135).
* Bụi > 150 µg/m³ (GP2Y1014AU0F).
* Cấu hình node email trong Node-RED (dùng Gmail SMTP) hoặc tích hợp bot Telegram để gửi cảnh báo.
* Kiểm tra thông báo bằng cách mô phỏng dữ liệu vượt ngưỡng (ví dụ: dùng khói nhẹ cho MQ-135).

### Tuần 7 (12-18/6/2025):

* Tối ưu code ESP32:
* Thêm kiểm tra lỗi dữ liệu (bỏ qua giá trị không hợp lệ).
* Thêm cơ chế reconnect WiFi/MQTT sau mỗi 5 giây nếu ngắt kết nối.
* Giảm tần suất gửi dữ liệu (mỗi 10 giây) để tiết kiệm năng lượng.
* Tinh chỉnh dashboard Grafana:
* Thêm biểu đồ so sánh nhiệt độ/độ ẩm giữa DHT11 và AHT20.
* Thêm panel dự báo thời tiết dựa trên áp suất (ví dụ: áp suất giảm → khả năng mưa).
* Tùy chỉnh giao diện (màu sắc, đơn vị, kích thước biểu đồ).
* Kiểm tra hệ thống trong môi trường thực tế:
* Đặt ESP32 + cảm biến ngoài trời (sân trường/công viên) trong 1-2 giờ.
* Kiểm tra dữ liệu trên Grafana, ghi nhận lỗi (nếu có).
* Sửa lỗi phát sinh (ví dụ: cảm biến bị ẩm, kết nối WiFi chập chờn).

## Giai Đoạn 4: Triển Khai và Báo Cáo (Tuần 8, 19-30/6/2025)

Mục tiêu: Lắp đặt hệ thống, hướng dẫn cộng đồng, viết báo cáo.

### Tuần 8 (19-30/6/2025):

* Lắp đặt hệ thống tại địa điểm (trường học/công viên):
* Đặt ESP32 + cảm biến ở khu vực thông thoáng, tránh ánh sáng mạnh và độ ẩm cao
* Đặt Raspberry Pi ở phòng máy hoặc khu vực có nguồn điện ổn định, kết nối mạng WiFi.
* Kiểm tra hệ thống sau khi lắp đặt:
* Đảm bảo dữ liệu cảm biến gửi đúng đến Grafana.
* Kiểm tra cảnh báo (mô phỏng vượt ngưỡng bằng khói hoặc nhiệt).
* Tạo tài liệu hướng dẫn sử dụng dashboard :
* Tổ chức buổi giới thiệu dự án:
* Mời học sinh, giáo viên, hoặc cư dân công viên tham gia (10-20 người).
* Trình bày cách hệ thống hoạt động, lợi ích (giám sát ô nhiễm, học IoT).
* Hướng dẫn xem dữ liệu trên Grafana, khuyến khích học sinh phân tích (ví dụ: bụi tăng vào giờ cao điểm).
* Viết báo cáo dự án :
* Mô tả mục tiêu, thiết bị, cách thực hiện.
* Kết quả (ảnh chụp dashboard, dữ liệu mẫu).
* Bài học kinh nghiệm (khó khăn, cách khắc phục).
* Đề xuất cải tiến (thêm cảm biến, ứng dụng di động).
* Lưu trữ code, sơ đồ mạch, tài liệu vào GitHub để tham khảo sau này.

https://github.com/thaivanhoa37/environmental-monitoring-system.git

# KẾT QUẢ MONG ĐỢI

* Hệ thống giám sát môi trường hoạt động ổn định, cung cấp dữ liệu thời gian thực về nhiệt độ, độ ẩm, chất lượng không khí, áp suất, bụi mịn.
* Dashboard Grafana dễ sử dụng, hỗ trợ học sinh phân tích dữ liệu môi trường.
* Cộng đồng (học sinh, giáo viên, dân chúng) hiểu thêm về IoT và ô nhiễm môi trường.
* Tài liệu hướng dẫn và báo cáo dự án để lưu trữ hoặc phát triển thêm.