**KẾ HOẠCH TỔNG THỂ DỰ ÁN:**

**HỆ THỐNG GIÁM SÁT KHÍ ĐỘC TỪ XA (REAL-TIME REMOTE TOXIC GAS MONITORING SYSTEM)**

|  |  |  |
| --- | --- | --- |
| **GIAI ĐOẠN 0: Nền tảng và Lập kế hoạch (Foundation and Planning)** | | Tiến độ |
| *Mục tiêu: Chuẩn bị đầy đủ công cụ, kiến thức và linh kiện cần thiết.* | |
| 0.1. Thiết lập môi trường phát triển | - Cài đặt VS Code. |  |
| - Cài đặt Extention PlatformIO IDE cho VS Code. |  |
| - Cài đặt Git trên máy tính. |  |
| 0.2. Quản lý mã nguồn với Git và GitHub | - Tạo một Repository mới cho đồ án. |  |
| - Thực hiện commit đầu tiên “Initial commit”. |  |
| - Đẩy lên GitHub. |  |
| 0.3. Mua sắm và Kiểm tra linh kiện | - Thống nhất lại danh sách linh kiện. |  |
| - Đặt mua tất cả các linh kiện đã thống nhất. |  |
| - Kiểm tra sơ bộ sau khi mua. |  |

|  |  |  |
| --- | --- | --- |
| **GIAI ĐOẠN 1:** **Lắp ráp Propotype và Phát triển Firmware (Prototyping & Firmware Development)** | | Tiến độ |
| *Mục tiêu: Xây dựng và lập trình cho hệ thống chạy ổn định trên breadboard.* | |
| * 1. Lắp ráp trên Breadboard (Breadboard Assembly) | - Kết nối ESP32 DevKit và Module LoRa SX1278 (qua giao thức SPI). |  |
| - Kết nối ESP32 Censor Node với cảm biến để kiểm tra riêng lẻ. |  |
| - Kết nối khối nguồn. |  |
| * 1. Lập trình Firmware cho Node Cảm biến (Sensor Node Firmware Development) | - Code đọc từng cảm biến (Read each Sensor): Viết các đoạn code nhỏ để đọc thành công dữ liệu từ từng cảm biến một. |  |
| - Code tích hợp LoRa (LoRa Integration): Cài đặt thư viện RadioLib và viết code gửi/nhận các gói tin LoRa đơn giản giữa 2 bộ ESP32-LoRa. |  |
| - Code đóng gói dữ liệu (Data Structuring): Tạo một cấu trúc dữ liệu (struct) trong C để chứa tất cả các thông tin từ cảm biến. |  |
| - Code Hoàn thiện luồng hoạt động (Filnalize Main Loop): Viết chương trình hoàn thiện cho Sensor Node: Thức dậy (Wake) - Đọc Cảm biến (Read all Sensor) - Gửi gói tin qua LoRa (Send packet via LoRa) - Đi vào chế độ Ngủ sâu (Enter Deep Sleep Mode). |  |
| * 1. Lập trình Firmware cho Gateway (Gateway Firmware Development) | - Code Kết nối WiFi (WiFi Connection): Viết code cho ESP32 Gateway để kết nối vào mạng WiFi. |  |
| - Code Nhận LoRa và Kết nối MQTT (Receive LoRa and MQTT Connection): Viết Code cho Gateway: Luôn nghe LoRa (Always listen for LoRa) - Khi nhận được gói tin, giải nén (Unpack) - Kết nối tới một MQTT Broler như HiveMQ (Connect to a public MQTT Broker (e.g., HiveMQ)) - Xuất dữ liệu lên các topic tương ứng (Publish data to corresponding topics (e.g., sensor/node1/CO2)). |  |
| * 1. Kiểm tra Toàn bộ luồng Dữ liệu (End-to-End Data Flow test) | - Sử dụng MQTT Client (như MQTT Explorer) trên máy tính để đăng ký các topic và xác nhận dữ liệu từ node cảm biến. |  |

|  |  |  |
| --- | --- | --- |
| **GIAI ĐOẠN 2:** **Thiết kế Phần cứng (Hardware Design - PCB)** | | Tiến độ |
| *Mục tiêu: Chuyển mô hình từ breadbroad sang bo mạch in PCB.* | |
| 2.1. Chuẩn bị thư viện Altium (Altium Library Preparation) | - Tìm kiếm hoặc tự tạo Symbol và Footprint cho các chân cắm (Socket/Heatder) của ESP32 DevKit, Module LoRa và các module cảm biến. |  |
| 2.2. Vẽ sơ đồ Nguyên lý (Schematic Design) | - Thiết kế sơ đồ theo các khối chức năng (MCU và LoRa, Cảm biến, Nguồn). |  |
| - Kết nối các khối lại với nhau theo sơ đồ đã chạy trên Breadboard. |  |
| 2.3. Thiết kế Layout mạch in (PCB Layout Design) | - Cài đặt định dạng bo mạch và cấu trúc. |  |
| - Bố trí linh kiện: Phân vùng rõ ràng khu vực Analog (Cảm biến), Digital (ESP32) và RF (LoRa). |  |
| - Đi dây. |  |
| 2.4. Xuất file Gia công (Manufacturing File Generation) | - Xuất file Gerber và khoan NC Drill từ Altium. |  |
| - Gửi cho Đơn vị Gia công PCB. |  |
| 2.5. Hàn mạch và Kiểm tra (PCB Assembly and Testing) | - Hàn tất cả các linh kiện lên bo PCB. |  |
| - Cắm các module vào socket để kiểm tra hoạt động. |  |

|  |  |  |
| --- | --- | --- |
| **GIAI ĐOẠN 3:** **Xây dựng Giao diện Giám sát (Monitoring Dashboard Development)** | | Tiến độ |
| *Mục tiêu: Trực quan hóa dữ liệu một cách thân thiện và hữu ích cho người dùng.* | |
| 3.1. Thiết lập Dự án Web FrontEnd (Setup FrontEnd Web Project) | - Tạo cấu trúc file cơ bản: indext.html, style.css, script.js. |  |
| 3.2. Kết nối và Hiển thị Dữ liệu theo Thời gian thực (Real-time Data Connection and Display) | - Code Tích hợp thư viện (Integrate JS Libraries): Thêm các thư viện JavaScript cần thiết (MQTT.js, Chart.js, Gauge.js, Leaflet.js) |  |
| - Code Kết nối MQTT (MQTT Connection via WebSockets): Viết code JavaScript để kết nối đến MQTT Broker qua WebSockets và đăng ký các topic dữ liệu. |  |
| - Code Cập nhật Giao diện (Update UI): Khi nhận được dữ liệu mới, cập nhật các thành phần trên web |  |
| 3.3. Trực quan hóa Nâng cao (Advanced Visualization) | - Code Biểu đồ Lịch sử (Hisorical Chart): Dùng Chart.js để vẽ biểu đồ đường thể hiện sự thay đổi của nồng độ không khí theo thời gian |  |
| - Code Đồng hồ đo Trực quan (Visual Gauges) Dùng Gauge.js để hiển thị giá trị hiện tại một cách trực quan. |  |
| - Code Bản đồ Giám sát (Geospatial Monitoring Map): Dùng Leaflet.js để hiển thị vị trí các node cảm biến, màu sắc của các marker thay đổi theo trạng thái cảnh báo. |  |
| 3.4. Triển khai lên GitHub Pages (Deploy to GitHub Pages) | - Đẩy toàn bộ code lên frontend lên repository GitHub. |  |
| - Kích hoạt tính năng GitHub Pages để đưa trang web lên mạng miễn phí. |  |

|  |  |  |
| --- | --- | --- |
| **GIAI ĐOẠN 4:** **Phát triển và Tích hợp Mô hình AI (AI Model Development and Integration)** | | Tiến độ |
| *Mục tiêu: Xây dựng và tích hợp một mô hình AI để cung cấp các thông tin thông minh, như ngưỡng cảnh báo và dự báo rủi ro.* | |
| 4.1. Thu thập và Tiền Xử lý dữ liệu (Data Acquisition and Preprocessing) | - Chạy hệ thống hoàn thiện (Phần cứng và Dashboard) để thu thập một bộ dữ liệu chuỗi thời gian của tất cả các chỉ số cảm biến |  |
| - Làm sạch và chuẩn bị dữ liệu đã thu thập để huấn luyện mô hình (Xử lý giá trị thiếu, chuẩn hóa) |  |
| 4.2. Lựa chọn và Huấn luyện Mô hình (Model Selection and Training) | - Xác định mục tiêu AI dựa trên đề xuất: “Huấn luyện AI để điều chỉnh các tham số theo mùa”. |  |
| - Chọn một mô hình phù hợp. |  |
| - Huấn luyện mô hình bằng 1 framework như TensorFlow hoặc scikit-learn trên bỗ dữ liệu đã thu thập. |  |
| - Đánh giá hiệu xuất và độ chính xác, lặp lại các bước nếu cần. |  |
| 4.3. Triển khai và Tích hợp Mô hình (Model Development and Integration) | - Lựa chọn phương án triển khai dựa trên tình hình thực tế: Trên Clound hoặc Trực tiếp lên Gateway. |  |
| - Code Tích hợp đầu ra của AI vào hệ thống. Dashboard sẽ hiển thị các ngưỡng hoặc dự báo rủi ro do AI tạo ra. Logic cảnh báo sử dụng các giá trị động. |  |

|  |  |  |
| --- | --- | --- |
| **GIAI ĐOẠN 5:** **Tích hợp hệ thống, Kiểm thử và Tối ưu hóa (System Integration, Testing and Optimization)** | | Tiến độ |
| *Mục tiêu: Đảm bảo toàn bộ hệ thống (Phần cứng và Phần mềm) hoạt động ổn định, chính xác.* | |
| 5.1. Tích hợp Hệ thống hoàn chỉnh (Full System Integration) | - Lắp ráp Node cảm biến và Gateway. |  |
| - Cung cấp nguồn và đảm bảo hệ thống gửi dữ liệu từ cảm biến lên Dashboard thành công. |  |
| 5.2. Kiểm thử chức năng và Hiệu năng | - Kiểm tra độ chính xác của cảm biến. |  |
| - Kiểm tra tầm phát sóng của LoRa. |  |
| - Kiểm tra thời lượng sử dụng và độ hiệu quả của chế độ Deep Sleep. |  |
| - Kiểm tra tính chính xác và hiệu quả của các kết quả từ mô hình AI. |  |
| 5.3. Tối ưu hóa (Optimization) | - Tinh chỉnh firmware để tiết kiệm năng lượng hơn. |  |
| - Tối ưu hóa mô hình AI để suy luận nhanh hơn hoặc sử dụng it tài nguyên hơn. |  |
| - Cải thiện giao diện/trải nghiệm người dung (UI/UX) của Dashboard dựa trên kết quả kiểm thử. |  |

|  |  |  |
| --- | --- | --- |
| **GIAI ĐOẠN 6:** **Hoàn thiện và Báo cáo (Finalization and Reporting)** | | Tiến độ |
| *Mục tiêu: Hoàn thiện sản phẩm cuối cung và chuẩn bị đầy đủ tài liệu theo yêu cầu của Đồ án.* | |
| 6.1. Thiết kế và In 3D vỏ hộp (Enclosure Design and 3D Printing) | - Thiết kê và in 3D vỏ hộp cho Node cảm biến. |  |
| 6.2. Viết báo cáo Đồ án (Thesis Report Writing) |  |  |
| 6.3. Chuẩn bị bảo vệ Đồ án (Thesis Defense Preparation) | - Làm slide thuyết trình. |  |
| - Chuẩn bị trước video demo cho sản phẩm, trình diễn các tính năng AI. |  |
| - Luyện tập kỹ phần Trình bày và Hỏi đáp. |  |
| 6.4. Hoàn thiện Repository trên GitHub (Finalize Github Repository) | - Dọn dẹp và thêm Chú thích cho code (Code Cleanup and Commenting) |  |
| - Viết file README.md |  |