# **Báo cáo Chuyên gia: Đề xuất Nâng cao Chất lượng và Tính Thực tiễn cho Đồ án Giám sát Khí độc từ xa**

## **Giới thiệu**

### **Mục đích của Báo cáo**

Báo cáo này được biên soạn nhằm cung cấp một bộ đề xuất chuyên sâu, mang tính định hướng kỹ thuật để nâng cao chất lượng, độ tin cậy và tính thực tiễn cho đồ án "Hệ thống giám sát khí độc từ xa theo thời gian thực".1 Các khuyến nghị trong tài liệu này không chỉ giải quyết các thách thức kỹ thuật cơ bản mà còn hướng đến việc chuyển đổi một nguyên mẫu (prototype) cấp độ sinh viên thành một giải pháp IoT (Internet of Things) hoàn chỉnh, mạnh mẽ và có khả năng ứng dụng cao trong thực tế.

### **Cấu trúc Báo cáo**

Báo cáo được cấu trúc thành ba phần chính, đi theo một lộ trình phát triển logic và chuyên nghiệp:

1. **Phần I: Tối ưu hóa Phần cứng để Đảm bảo Độ tin cậy và Khả năng Triển khai Thực địa.** Phần này tập trung vào việc xây dựng một nền tảng phần cứng vững chắc, từ việc lựa chọn cảm biến chính xác đến thiết kế mạch in (PCB) chuyên nghiệp và quản lý năng lượng thông minh.
2. **Phần II: Xây dựng Firmware Mạnh mẽ và Có khả năng Mở rộng.** Phần này chuyển trọng tâm sang phần mềm lõi của thiết bị, đề xuất các quy trình làm việc hiện đại, kiến trúc giao tiếp hiệu quả và các phương pháp xử lý dữ liệu thông minh ngay tại biên (edge).
3. **Phần III: Hoàn thiện Sản phẩm và Giao diện Giám sát Trực quan.** Phần cuối cùng đề cập đến việc đưa dữ liệu đến người dùng một cách ý nghĩa, đồng thời vạch ra các bước cần thiết để hoàn thiện sản phẩm, từ việc chế tạo vỏ bảo vệ đến chuẩn bị cho sản xuất.

### **Triết lý Cốt lõi**

Toàn bộ các đề xuất được xây dựng dựa trên các nguyên tắc kỹ thuật cốt lõi: độ tin cậy (reliability), độ chính xác (accuracy), khả năng mở rộng (scalability) và tính thực tiễn (practicality). Mục tiêu là trang bị cho đồ án những tiêu chuẩn kỹ thuật của một sản phẩm công nghiệp, giúp hệ thống không chỉ hoạt động ổn định trong môi trường phòng thí nghiệm mà còn có thể đứng vững trước các thách thức của môi trường công nghiệp thực tế như hầm mỏ hay nhà máy hóa chất.1

## **Phần I: Tối ưu hóa Phần cứng để Đảm bảo Độ tin cậy và Khả năng Triển khai Thực địa**

Nền tảng của một hệ thống IoT đáng tin cậy là phần cứng được thiết kế và chế tạo một cách chuyên nghiệp. Phần này sẽ đưa ra các khuyến nghị nhằm cải thiện đáng kể độ chính xác, độ bền và hiệu quả năng lượng của thiết bị cảm biến.

### **1.1. Lựa chọn và Tích hợp Cảm biến Nâng cao**

Chất lượng của toàn bộ hệ thống phụ thuộc trực tiếp vào độ chính xác và độ tin cậy của dữ liệu đầu vào từ các cảm biến. Việc lựa chọn công nghệ cảm biến phù hợp là bước đi chiến lược quan trọng nhất để nâng tầm đồ án.

#### **Thách thức về Độ chính xác của Cảm biến**

Đề xuất ban đầu liệt kê việc giám sát các khí CO2, CO, SO2, NO2, và CH4.1 Các cảm biến bán dẫn oxit kim loại (MOS) thuộc dòng MQ, thường được sử dụng trong các dự án điện tử không chuyên, có ưu điểm là chi phí thấp. Tuy nhiên, chúng tồn tại những nhược điểm cố hữu nghiêm trọng: độ nhạy chéo cao (một cảm biến có thể phản ứng với nhiều loại khí khác nhau), độ trôi (drift) theo thời gian và điều kiện môi trường, và yêu cầu quy trình hiệu chuẩn phức tạp với khí mẫu đã biết nồng độ để có thể đưa ra ước tính PPM (parts per million - phần triệu).2 Cảm biến MiCS-6814, mặc dù tích hợp nhiều thành phần, về cơ bản vẫn là cảm biến MOS và chủ yếu dùng để đo lường định tính (phát hiện sự tăng/giảm của nồng độ khí) thay vì định lượng chính xác nếu không có hiệu chuẩn trong phòng thí nghiệm.4

Việc sử dụng các cảm biến này cho một hệ thống "giám sát khí độc" tạo ra một mâu thuẫn: mục đích của hệ thống đòi hỏi độ tin cậy và an toàn cao, trong khi công nghệ cảm biến được chọn lại không đáp ứng được yêu cầu đó. Để giải quyết vấn đề này, một chiến lược cảm biến lai (hybrid) được đề xuất.

#### **Chiến lược Cảm biến Lai được Khuyến nghị**

1. **Đối với CO2:** Tiếp tục sử dụng cảm biến hồng ngoại không phân tán (NDIR) như **MH-Z19B**. Công nghệ NDIR có độ chọn lọc rất cao đối với CO2, ổn định trong thời gian dài và cung cấp đầu ra kỹ thuật số (UART) hoặc PWM trực tiếp theo đơn vị PPM. Điều này giúp loại bỏ nhiễu và sai số từ quá trình chuyển đổi analog-số (ADC).7
2. **Đối với Khí độc (CO, SO2, NO2):** Đây là nâng cấp quan trọng nhất. Cần thay thế các cảm biến MOS bằng **cảm biến điện hóa (electrochemical sensors)**. Mặc dù có chi phí cao hơn, cảm biến điện hóa cung cấp độ chính xác vượt trội, độ nhạy chéo thấp và là tiêu chuẩn được sử dụng trong các thiết bị an toàn công nghiệp. Sự thay đổi này sẽ nâng cao đáng kể tính thực tiễn và độ tin cậy của đồ án.
3. **Đối với CH4 (Methane):** Có thể tiếp tục sử dụng cảm biến xúc tác hoặc MOS (như MQ-4), vì mục đích chính thường là phát hiện rò rỉ (có/không có khí) thay vì đo lường nồng độ thấp một cách chính xác.

#### **Chiến lược Tích hợp Đa cảm biến (Bus I2C)**

Để đơn giản hóa việc đi dây và tiết kiệm chân GPIO của vi điều khiển, nên ưu tiên tích hợp các cảm biến sử dụng giao thức I2C. ESP32 hỗ trợ hai bus I2C phần cứng, mang lại sự linh hoạt trong thiết kế.10

* **Thiết kế Sơ đồ Nguyên lý:** Tất cả các thiết bị I2C (cảm biến, màn hình OLED, v.v.) được kết nối song song vào cùng hai đường tín hiệu SDA (dữ liệu) và SCL (xung nhịp). Cần có hai điện trở kéo lên (pull-up) với giá trị khoảng từ 2.2kΩ đến 4.7kΩ trên cả hai đường này, nối với nguồn 3.3V.11
* **Xử lý Xung đột Địa chỉ:** Mỗi thiết bị trên bus I2C phải có một địa chỉ duy nhất. Nếu hai cảm biến có cùng địa chỉ được lập trình cứng, chúng không thể hoạt động trên cùng một bus. Các giải pháp bao gồm:
  + Sử dụng một bộ ghép kênh I2C (I2C multiplexer) như TCA9548A. Thiết bị này hoạt động như một công tắc, cho phép vi điều khiển chọn và giao tiếp với một trong nhiều bus I2C con.12
  + Tận dụng hai bộ điều khiển I2C riêng biệt của ESP32 để cách ly các thiết bị có địa chỉ trùng nhau.10

Dưới đây là bảng so sánh các công nghệ cảm biến để làm rõ hơn cho lựa chọn chiến lược lai.

**Table 1: Comparative Analysis of Gas Sensor Technologies**

| Sensor Model/Type | Technology | Target Gas(es) | Interface | Typical Accuracy | Power Consumption | Relative Cost |
| --- | --- | --- | --- | --- | --- | --- |
| **MH-Z19B** | NDIR | CO2 | UART, PWM | ±(50ppm + 5% reading) 7 | Trung bình < 20mA 7 | Trung bình |
| **MiCS-6814** | MOS | CO, NO2, NH3, VOCs | Analog | Định tính, cần hiệu chuẩn 5 | Cao (do nung nóng) | Thấp |
| **Electrochemical Sensor** | Electrochemical | CO, SO2, NO2 (chuyên dụng) | Analog, I2C | Cao, độ chọn lọc tốt | Thấp | Cao |
| **MQ-4** | MOS | CH4, Khí tự nhiên | Analog | Định tính, phát hiện rò rỉ | Cao (do nung nóng) | Rất thấp |

### **1.2. Thiết kế Mạch in (PCB) Chuyên nghiệp trong Altium Designer**

Một thiết kế PCB tốt không chỉ đơn thuần là kết nối các linh kiện, mà nó còn là một thành phần kỹ thuật quan trọng quyết định đến sự ổn định, hiệu năng và khả năng chống nhiễu của toàn bộ hệ thống.

#### **Triết lý Thiết kế Module**

Một hệ thống phức tạp nên được chia thành các khối chức năng nhỏ hơn, dễ quản lý và kiểm tra.

* **Sơ đồ Nguyên lý Phân cấp (Hierarchical Schematics):** Sử dụng tính năng thiết kế đa tờ (multi-sheet) hoặc phân cấp của Altium Designer để tạo các tệp sơ đồ nguyên lý riêng biệt cho từng khối chức năng, ví dụ: Power\_Management.SchDoc, MCU\_and\_LoRa.SchDoc, và Sensor\_Array.SchDoc.14 Cách làm này giúp đơn giản hóa việc gỡ lỗi và cho phép tái sử dụng các khối thiết kế trong tương lai.
* **Module hóa Vật lý:** Đối với phiên bản nguyên mẫu, nên cân nhắc thiết kế mảng cảm biến trên một PCB riêng, kết nối với bo mạch chính thông qua các đầu nối board-to-board (ví dụ: dòng FINEPITCH của Phoenix Contact hoặc SlimStack của Molex).16 Điều này cho phép thay đổi, thử nghiệm các loại cảm biến khác nhau một cách dễ dàng.

#### **Nguyên tắc Thiết kế Mạch Tín hiệu Hỗn hợp (Mixed-Signal)**

Bo mạch chứa các mạch analog nhạy cảm (cảm biến), một lõi kỹ thuật số nhiễu (ESP32), và một phần RF (LoRa). Việc ngăn chặn nhiễu xuyên kênh là tối quan trọng.

* **Bố trí và Phân vùng Linh kiện:** Phân chia PCB thành các khu vực vật lý riêng biệt. Nhóm tất cả các thành phần analog vào một vùng và các thành phần kỹ thuật số vào một vùng khác. ADC, là cầu nối giữa hai thế giới này, nên được đặt ở ranh giới giữa hai vùng.15
* **Mặt phẳng Đất (Ground Plane) Hợp nhất:** **Không chia mặt phẳng đất thành các vùng "AGND" và "DGND" riêng biệt.** Đây là một phương pháp cũ và thường gây ra nhiều vấn đề hơn là giải quyết chúng (ví dụ: vòng lặp đất, anten phát xạ nhiễu).23 Thay vào đó, hãy sử dụng một mặt phẳng đất liền lạc, duy nhất trên một lớp bên trong (ví dụ: Lớp 2 trên PCB 4 lớp). Việc phân vùng linh kiện vật lý sẽ tự động định hướng các dòng điện trở về (return currents), giữ cho nhiễu kỹ thuật số không lan sang vùng analog.20
* **Chiến lược Mặt phẳng Nguồn (Power Plane):** Nếu sử dụng PCB 4 lớp, hãy dành Lớp 3 cho mặt phẳng nguồn. Mặt phẳng này có thể được chia thành các vùng điện áp khác nhau (ví dụ: 3.3V\_Digital, 3.3V\_Analog) nếu cần, nhưng các vùng này không nên chồng chéo lên nhau trên các lớp liền kề để tránh khớp nối điện dung.27

#### **Thiết kế RF cho Giao tiếp LoRa (433MHz)**

* **Kiểm soát Trở kháng (Impedance Control):** Đường mạch (trace) từ chân RF của module LoRa đến đầu nối anten phải là một đường truyền (transmission line) có trở kháng 50-ohm để đảm bảo truyền công suất tối đa và giảm thiểu sóng phản xạ.
* **Hướng dẫn trong Altium Designer:**
  1. **Layer Stack Manager:** Xác định cấu trúc 4 lớp của PCB (ví dụ: Signal-GND-Power-Signal) với các thông số vật liệu (FR-4) và độ dày chính xác.28
  2. **Impedance Profile:** Trong Layer Stack Manager, tạo một "Impedance Profile" mới cho đường mạch đơn (single-ended) với trở kháng mục tiêu là 50-ohm. Bộ giải trường điện từ tích hợp của Altium sẽ tự động tính toán độ rộng đường mạch cần thiết cho lớp trên cùng dựa trên cấu trúc lớp đã định nghĩa.29 Đối với PCB 4 lớp 1.6mm tiêu chuẩn, độ rộng này sẽ hẹp và thực tế hơn nhiều so với trên PCB 2 lớp.32
  3. **Luật Thiết kế (Routing Rule):** Tạo một luật thiết kế PCB mới áp dụng "Impedance Profile" này cho net RF, đảm bảo rằng công cụ đi dây tương tác (interactive router) sẽ tự động sử dụng độ rộng chính xác.30
* **Nguyên tắc Bố trí:** Giữ đường mạch RF ngắn và thẳng nhất có thể. Hạn chế sử dụng via. Bao quanh đường mạch RF bằng một lớp phủ đất (ground pour) và sử dụng kỹ thuật "via stitching" (đặt nhiều via kết nối các mặt phẳng đất ở lớp trên và dưới) dọc theo đường mạch để tạo ra một đường dẫn được che chắn và ngăn chặn phát xạ nhiễu.33

#### **Bố trí Tụ lọc Nhiễu (Decoupling) Chiến lược**

* Đặt một tụ gốm 100nF (0.1µF) gần nhất có thể với *mọi* chân cấp nguồn của ESP32 và các vi mạch khác.35
* Đối với các thành phần analog nhạy cảm như ADC hoặc bộ tạo điện áp tham chiếu ngoài, hãy đặt tụ lọc nhiễu cho cả chân nguồn analog và digital.37
* Đặt các tụ lọc dung lượng lớn hơn (ví dụ: 10µF - 100µF), gọi là tụ bulk, gần các điểm cấp nguồn vào bo mạch và cho các khối chức năng chính.35

### **1.3. Quản lý Năng lượng Thông minh và Bền vững**

Đối với một thiết bị triển khai thực địa, việc quản lý năng lượng hiệu quả là yếu tố quyết định đến tuổi thọ và tính khả thi của hệ thống.

#### **Mạch sạc Năng lượng Mặt trời**

* IC sạc TP4056 là một lựa chọn phù hợp và tiết kiệm chi phí cho việc sạc pin Li-ion đơn cell.39
* **Bảo vệ Tấm pin Mặt trời:** Khi sử dụng tấm pin mặt trời làm nguồn đầu vào, cần phải đặt một diode Schottky nối tiếp với cực dương của tấm pin. Điều này ngăn chặn pin xả ngược dòng điện qua tấm pin vào ban đêm hoặc khi ánh sáng yếu, bảo vệ cả pin và tấm pin.40

#### **TÍNH NĂNG QUAN TRỌNG - Mạch Chia sẻ Tải (Load Sharing)**

* **Vấn đề:** Các module TP4056 thông thường kết nối tải trực tiếp vào hai cực của pin. Nếu hệ thống đang tiêu thụ dòng điện trong khi sạc, thuật toán kết thúc sạc của TP4056 (dựa trên việc phát hiện dòng sạc giảm xuống dưới một ngưỡng nhất định) sẽ bị đánh lừa. Nó có thể không bao giờ ngắt sạc, dẫn đến sạc nhồi quá mức và làm hỏng pin, hoặc có thể đóng ngắt liên tục, làm giảm tuổi thọ pin.43
* **Giải pháp:** Triển khai một mạch chia sẻ tải. Mạch này ưu tiên cấp nguồn cho tải từ nguồn đầu vào (USB/Năng lượng mặt trời) khi có sẵn, và chỉ lấy nguồn từ pin khi nguồn đầu vào bị ngắt. Một mạch đơn giản và hiệu quả sử dụng một P-channel MOSFET và một diode Schottky khác.44 Đây là một thực hành thiết kế chuyên nghiệp đảm bảo sức khỏe của pin và độ tin cậy của hệ thống. Một sơ đồ nguyên lý hoàn chỉnh cho mạch này cần được tích hợp vào thiết kế.

#### **Tối ưu hóa Năng lượng bằng Firmware**

* **Chế độ Ngủ sâu (Deep Sleep) của ESP32:** Phương pháp hiệu quả nhất để tiết kiệm năng lượng là đưa ESP32 vào chế độ ngủ sâu giữa các lần đo. Ở chế độ này, CPU chính bị tắt và mức tiêu thụ điện năng giảm xuống chỉ còn vài microamp.
* **Đánh thức bằng RTC Timer:** Sử dụng bộ đồng xử lý Năng lượng Cực thấp (ULP) hoặc bộ định thời RTC của ESP32 để đánh thức thiết bị theo các khoảng thời gian đã lập trình (ví dụ: 15 phút một lần) để thực hiện đo đạc và truyền dữ liệu.
* **Cấp nguồn có kiểm soát cho Linh kiện (Power Gating):** Sử dụng các MOSFET được điều khiển bởi chân GPIO của ESP32 để ngắt hoàn toàn nguồn điện cấp cho các cảm biến tiêu thụ nhiều năng lượng (đặc biệt là các cảm biến MOS có bộ nung) khi chúng không được sử dụng.

## **Phần II: Xây dựng Firmware Mạnh mẽ và Có khả năng Mở rộng**

Phần này chuyển từ phần cứng vật lý sang phần mềm điều khiển. Trọng tâm là áp dụng các công cụ và phương pháp chuyên nghiệp để tạo ra mã nguồn đáng tin cậy, dễ bảo trì và hiệu quả.

### **2.1. Quy trình Phát triển Chuyên nghiệp**

#### **Môi trường Phát triển Tích hợp (IDE) - Vượt ra ngoài Arduino IDE**

Mặc dù Arduino IDE rất tốt cho người mới bắt đầu, việc phát triển nhúng chuyên nghiệp đòi hỏi các công cụ mạnh mẽ hơn.

* **Khuyến nghị: VS Code + PlatformIO:** PlatformIO là một tiện ích mở rộng cho Visual Studio Code, cung cấp một môi trường phát triển chuyên nghiệp với các tính năng như tự động hoàn thành mã thông minh (IntelliSense), quản lý thư viện theo từng dự án, gỡ lỗi dễ dàng và tích hợp với hệ thống quản lý phiên bản.45 Đây là một bước tiến đáng kể so với Arduino IDE tiêu chuẩn.

#### **Quản lý Phiên bản với Git**

* **Thực hành Thiết yếu:** Toàn bộ mã nguồn phải được quản lý bằng Git. Điều này cho phép theo dõi các thay đổi, cộng tác và quay lại các phiên bản trước đó nếu có lỗi xảy ra.
* **Kho chứa GitHub:** Tạo một kho chứa (repository) miễn phí trên GitHub để lưu trữ mã nguồn của dự án.50
* **Tệp .gitignore:** Việc tạo một tệp .gitignore là cực kỳ quan trọng để ngăn các tệp tạm thời được tạo ra trong quá trình biên dịch, các cấu hình cục bộ và các thư mục chứa thư viện phụ thuộc bị đưa vào kho chứa. Điều này giữ cho mã nguồn sạch sẽ và tập trung. Nên sử dụng một mẫu .gitignore được thiết kế riêng cho các dự án PlatformIO.54

### **2.2. Giao tiếp và Xử lý Dữ liệu Nâng cao**

#### **Triển khai LoRa Mạnh mẽ**

* **Lựa chọn Thư viện:** Thay vì thư viện arduino-LoRa cơ bản, nên sử dụng **RadioLib**.57 Thư viện này hỗ trợ nhiều loại phần cứng hơn, có khả năng cấu hình cao hơn và cung cấp cơ chế xử lý lỗi mạnh mẽ hơn, điều này rất quan trọng để có một liên kết truyền thông đáng tin cậy.
* **Đóng gói Dữ liệu Hiệu quả:** Tránh gửi dữ liệu dưới dạng chuỗi văn bản thuần túy (ví dụ: "CO2:450,CO:10"). Cách làm này rất không hiệu quả. Thay vào đó, hãy đóng gói tất cả dữ liệu cảm biến vào một cấu trúc C (struct). Điều này tạo ra một gói dữ liệu nhị phân nhỏ gọn, giúp truyền đi nhanh hơn và phân tích cú pháp ở phía nhận dễ dàng hơn.
* C

// Ví dụ về Struct C cho gói tin LoRa

typedef struct {

uint16\_t co2\_ppm;

uint16\_t co\_ppm;

uint16\_t no2\_ppb;

uint16\_t so2\_ppb;

float temperature;

float humidity;

float battery\_voltage;

} SensorData;

* Cấu trúc này có thể được gửi đi dưới dạng một mảng byte và dễ dàng được tái tạo lại ở phía máy thu.61

#### **Xử lý Dữ liệu trên Thiết bị (Edge Computing)**

* **Chuyển đổi ADC sang PPM:** Đối với các cảm biến analog, firmware phải chuyển đổi giá trị ADC thô thành nồng độ PPM. Điều này đòi hỏi việc sử dụng các công thức logarit và các đường cong đặc tính được cung cấp trong biểu dữ liệu (datasheet) của cảm biến.2 Quá trình xử lý này nên diễn ra trên nút cảm biến  
   *trước khi* truyền đi để tiết kiệm băng thông.
* **Lọc Dữ liệu:** Các giá trị đọc thô từ cảm biến có thể bị nhiễu. Cần triển khai một bộ lọc kỹ thuật số đơn giản, chẳng hạn như bộ lọc trung bình trượt (moving average filter), để làm mịn các giá trị đọc và truyền đi một giá trị ổn định hơn.

#### **Cảnh báo theo Ngữ cảnh**

* **Vấn đề với Ngưỡng tùy tiện:** Đặt cảnh báo ở mức "1000 ppm" là vô nghĩa nếu không có ngữ cảnh. Mức đó có cao không? Có nguy hiểm không?
* **Giải pháp:** Dựa vào các ngưỡng cảnh báo theo các tiêu chuẩn môi trường chính thức của Việt Nam. Điều này làm cho các cảnh báo của dự án có giá trị pháp lý và khoa học.
  + **QCVN 05:2013/BTNMT:** Cung cấp các giới hạn cho chất lượng không khí xung quanh đối với các khí như SO2, NO2 và CO.67
  + **TCVN 5508:2009:** Cung cấp các tiêu chuẩn về chất lượng không khí tại nơi làm việc.71
* Firmware nên có nhiều mức cảnh báo (ví dụ: CẢNH BÁO, NGUY HIỂM) dựa trên các văn bản chính thức này.

**Table 2: Air Quality Thresholds based on Vietnamese Standards (QCVN 05:2013/BTNMT)**

| Gas | Standard (QCVN 05:2013/BTNMT) | Averaging Time | Permissible Concentration (μg/m3) | Approx. Concentration (ppm/ppb) | Recommended Alert Level |
| --- | --- | --- | --- | --- | --- |
| **SO2** | QCVN 05:2013/BTNMT | 1 hour | 350 | ~134 ppb | Warning |
| **CO** | QCVN 05:2013/BTNMT | 1 hour | 30,000 | ~26 ppm | Warning |
| **CO** | QCVN 05:2013/BTNMT | 8 hours | 10,000 | ~8.7 ppm | Information |
| **NO2** | QCVN 05:2013/BTNMT | 1 hour | 200 | ~106 ppb | Warning |

*Lưu ý: Việc chuyển đổi từ μg/m3 sang ppm/ppb là gần đúng và phụ thuộc vào nhiệt độ và áp suất. Các giá trị trên được tính ở điều kiện tiêu chuẩn.*

**Table 3: Key LoRa Configuration Parameters in RadioLib**

| Parameter | Function in RadioLib | Recommended Value (433MHz) | Description |
| --- | --- | --- | --- |
| Carrier Frequency | radio.begin(freq,...) | 433.0 | Tần số sóng mang trung tâm (MHz). Phải giống nhau ở cả bên gửi và bên nhận. |
| Bandwidth | radio.begin(..., bw,...) | 125.0 | Độ rộng băng thông tín hiệu (kHz). Giá trị lớn hơn cho tốc độ dữ liệu cao hơn nhưng giảm độ nhạy. |
| Spreading Factor | radio.begin(..., sf,...) | 9 | Hệ số trải phổ. Giá trị cao hơn (7-12) tăng phạm vi và khả năng chống nhiễu nhưng giảm tốc độ dữ liệu. |
| Coding Rate | radio.begin(..., cr,...) | 7 | Tỷ lệ mã hóa sửa lỗi (4/5, 4/6, 4/7, 4/8). Giá trị cao hơn tăng độ tin cậy nhưng cũng tăng thời gian phát sóng. |
| Sync Word | radio.begin(..., syncWord,...) | 0x12 | Một byte định danh để phân biệt các mạng LoRa khác nhau. |
| Output Power | radio.begin(..., pwr,...) | 17 | Công suất phát (dBm). Giá trị cao hơn tăng phạm vi nhưng cũng tăng mức tiêu thụ năng lượng. |

### **2.3. Quản lý và Bảo trì Thiết bị từ xa**

#### **Cập nhật Firmware qua mạng (Over-the-Air - OTA)**

* **Đối với Gateway (có WiFi):** Việc triển khai cập nhật OTA là rất quan trọng để bảo trì. Thư viện **ArduinoOTA** cho phép cập nhật firmware qua WiFi mà không cần truy cập vật lý.75 Đây nên là một tính năng tiêu chuẩn trong mã nguồn của gateway.
* **Đối với Nút Cảm biến (chỉ có LoRa):** Cập nhật OTA qua LoRa là **cực kỳ không thực tế và không nên thử nghiệm**. Tốc độ dữ liệu của LoRa rất thấp (tốt nhất cũng chỉ vài kbps). Một tệp nhị phân firmware điển hình có dung lượng hơn 1MB. Việc truyền tệp này sẽ mất hàng giờ hoặc hàng ngày, làm cạn kiệt toàn bộ pin và làm bão hòa mạng lưới.78 Đây là một điểm hiểu biết thực tế quan trọng phân biệt giữa một khái niệm học thuật và một ràng buộc trong thế giới thực.
* **Giải pháp thay thế cho Cập nhật tại thực địa:** Đối với các nút cảm biến từ xa, việc cập nhật firmware sẽ yêu cầu truy cập vật lý (ví dụ: qua cổng USB hoặc một chế độ cập nhật cục bộ qua Bluetooth/WiFi được kích hoạt bằng một nút nhấn vật lý). Hạn chế thực tế này cần được thừa nhận trong báo cáo đồ án.

## **Phần III: Hoàn thiện Sản phẩm và Giao diện Giám sát Trực quan**

Phần cuối cùng này tập trung vào việc đưa dữ liệu đến người dùng một cách có ý nghĩa và xem xét các bước cuối cùng để biến nguyên mẫu thành một sản phẩm hoàn chỉnh.

### **3.1. Gateway và Kết nối Đám mây**

#### **Firmware của Gateway**

Vai trò của nút gateway đơn giản nhưng rất quan trọng:

1. Khởi tạo WiFi và kết nối vào mạng cục bộ.
2. Khởi tạo MQTT client và kết nối đến broker.
3. Vào chế độ lắng nghe LoRa liên tục.
4. Khi nhận được một gói tin SensorData qua LoRa, tiến hành phân tích cú pháp.
5. Xuất bản (publish) dữ liệu lên các chủ đề (topic) MQTT riêng biệt (ví dụ: gas\_monitor/node\_01/co2, gas\_monitor/node\_01/temperature).

#### **MQTT Broker để Thử nghiệm**

Để phát triển, một MQTT broker công cộng miễn phí là lý tưởng. **HiveMQ** cung cấp một broker công cộng tuyệt vời, dễ sử dụng, hỗ trợ cả MQTT tiêu chuẩn và MQTT qua WebSockets.80 Firmware sẽ sử dụng thư viện

**PubSubClient** để kết nối và xuất bản dữ liệu.84

### **3.2. Bảng điều khiển (Dashboard) trên nền Web thời gian thực**

#### **Lưu trữ Miễn phí với GitHub Pages**

Một bảng điều khiển đơn giản nhưng mạnh mẽ có thể được xây dựng bằng HTML, CSS và JavaScript tiêu chuẩn. Toàn bộ ứng dụng front-end này có thể được lưu trữ miễn phí trực tiếp từ kho chứa GitHub của dự án bằng cách sử dụng GitHub Pages.88 Điều này thể hiện một chiến lược triển khai toàn diện và tiết kiệm chi phí.

#### **Dữ liệu Thời gian thực với MQTT qua WebSockets**

* Các trình duyệt web không thể tạo kết nối MQTT trực tiếp. Chúng phải sử dụng MQTT qua WebSockets.
* Thư viện **Paho.js** hoặc **MQTT.js** sẽ được sử dụng trong mã JavaScript của bảng điều khiển để kết nối với broker HiveMQ thông qua cổng WebSocket của nó (ví dụ: 8000 hoặc 8884 cho SSL) và đăng ký (subscribe) các chủ đề dữ liệu.83

#### **Trực quan hóa Dữ liệu Nâng cao**

* **Xu hướng Lịch sử:** Sử dụng **Chart.js** để tạo các biểu đồ đường hiển thị dữ liệu lịch sử cho từng loại khí, cập nhật theo thời gian thực khi có tin nhắn mới từ MQTT broker.97
* **Đồng hồ đo Trực quan:** Đối với các giá trị đọc hiện tại, sử dụng một thư viện như **Gauge.js** hoặc tương tự 102 để tạo các đồng hồ đo kiểu "đồng hồ tốc độ". Chúng trực quan hơn nhiều cho việc nắm bắt thông tin nhanh chóng so với chỉ một con số. Các dải màu của đồng hồ (xanh, vàng, đỏ) có thể được thiết lập theo các tiêu chuẩn Việt Nam đã xác định trước đó.
* **Bản đồ Giám sát Không gian địa lý:** Sử dụng thư viện **Leaflet.js** với các lớp bản đồ (tile layers) miễn phí từ OpenStreetMap để hiển thị một bản đồ.107 Mỗi nút cảm biến sẽ được biểu diễn bằng một điểm đánh dấu (marker). Tính năng cốt lõi là tự động cập nhật màu sắc của điểm đánh dấu dựa trên dữ liệu mới nhất nhận được qua MQTT (ví dụ:  
   marker.setIcon(greenIcon) hoặc marker.setIcon(redIcon)), cung cấp phản hồi trực quan ngay lập tức về tình trạng của tất cả các địa điểm được giám sát.112

### **3.3. Từ Nguyên mẫu đến Sản phẩm**

#### **Thiết kế Vỏ bảo vệ**

* Cần xem xét việc thiết kế một vỏ bảo vệ bằng công nghệ in 3D hoặc sử dụng vỏ hộp có sẵn. Các yếu tố chính là khả năng chống chịu thời tiết (hướng tới tiêu chuẩn IP65) và đảm bảo thông gió thích hợp để các cảm biến khí có thể đo lường chính xác môi trường xung quanh.

#### **Chuẩn bị cho Sản xuất**

* **Tạo Tệp Sản xuất:** Cung cấp hướng dẫn từng bước về cách xuất các tệp Gerber (RS-274X) và tệp khoan NC Drill từ Altium Designer. Đây là các tệp tiêu chuẩn được yêu cầu bởi tất cả các nhà sản xuất PCB.117
* **Dịch vụ Gia công PCB:**
  + **Trong nước (Việt Nam):** Liệt kê các dịch vụ địa phương được đề xuất để làm mạch mẫu nhanh chóng, như Hatakey, Datmachin (Hptech), và Thegioichip, ghi chú về khả năng và thông tin liên hệ của họ.121 Điều này bổ sung giá trị thực tiễn to lớn.
  + **Quốc tế:** Thảo luận về các lựa chọn quốc tế phổ biến và tiết kiệm chi phí như **JLCPCB** và **PCBgogo**, so sánh chúng về chi phí, tốc độ và chất lượng cho các đơn hàng làm mạch mẫu.125

#### **Tìm nguồn cung ứng Linh kiện**

* Cung cấp danh sách các nhà cung cấp linh kiện điện tử uy tín tại Việt Nam (Hà Nội và TP.HCM), chẳng hạn như Hshop.vn, Linh Kiện Minh Hà, và Điện Tử Tương Lai, để hỗ trợ việc tìm mua các bộ phận cần thiết cho nguyên mẫu.129 Điều này giải quyết thách thức thực tế của việc mua sắm.

## **Kết luận và Khuyến nghị**

Đồ án "Hệ thống giám sát khí độc từ xa theo thời gian thực" có một nền tảng vững chắc và tiềm năng ứng dụng thực tế cao.1 Bằng cách áp dụng các đề xuất được trình bày trong báo cáo này, đồ án có thể được nâng cấp đáng kể về mọi mặt, từ độ chính xác của dữ liệu, độ tin cậy của phần cứng, đến tính hiệu quả của phần mềm và trải nghiệm người dùng.

Các khuyến nghị chính có thể được tóm tắt như sau:

1. **Ưu tiên Độ chính xác Dữ liệu:** Chuyển đổi sang sử dụng cảm biến điện hóa cho các khí độc quan trọng là nâng cấp có tác động lớn nhất, giúp hệ thống trở nên đáng tin cậy cho các ứng dụng an toàn.
2. **Thiết kế PCB Chuyên nghiệp:** Áp dụng các nguyên tắc thiết kế module, bố trí mạch tín hiệu hỗn hợp với mặt phẳng đất hợp nhất, và kiểm soát trở kháng cho đường truyền RF. Một PCB được thiết kế tốt là xương sống của một thiết bị ổn định.
3. **Xây dựng Hệ thống Năng lượng Bền vững:** Triển khai mạch chia sẻ tải (load sharing) để bảo vệ pin và kết hợp với các chiến lược tiết kiệm năng lượng trong firmware (deep sleep) để tối đa hóa thời gian hoạt động của thiết bị tại thực địa.
4. **Áp dụng Quy trình Phát triển Hiện đại:** Chuyển sang sử dụng VS Code với PlatformIO và quản lý mã nguồn bằng Git. Điều này không chỉ cải thiện hiệu suất làm việc mà còn là một kỹ năng chuyên nghiệp quan trọng.
5. **Tạo ra một Bảng điều khiển Cung cấp Thông tin Hữu ích:** Xây dựng một giao diện người dùng không chỉ hiển thị dữ liệu mà còn cung cấp ngữ cảnh thông qua việc trực quan hóa (đồng hồ đo, bản đồ) và các ngưỡng cảnh báo dựa trên tiêu chuẩn quốc gia.

Việc thực hiện các đề xuất này sẽ đòi hỏi nỗ lực nghiên cứu và triển khai bổ sung, nhưng kết quả sẽ là một sản phẩm không chỉ đáp ứng yêu cầu của một đồ án tốt nghiệp xuất sắc mà còn đặt nền móng cho một giải pháp IoT có giá trị thương mại và xã hội.