ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN

ĐỒ ÁN TỐT NGHIỆP
CAPSTONE PROJECT

NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HOÁ

ĐỀ TÀI:

THIẾT KẾ TỔNG ĐÀI TIN NHẮN SMS GATEWAY CHO MẠNG CẢM BIẾN KHÔNG DÂY

Người hướng dẫn: **TS. NGÔ ĐÌNH THANH**
Sinh viên thực hiện: **NGUYỄN TUẤN PHONG**
Số thẻ sinh viên: **105200463**
Lớp: **20TDHCLC3**

Đà Nẵng, 6/2025

TÓM TẮT

Tên đề tài: Thiết kế tổng đài tin nhắn SMS gateway cho mạng cảm biến không dây.

Sinh viên thực hiện: Nguyễn Tuấn Phong

Số thẻ SV: 105200463 Lớp: 20TDHCLC3

Mất mát dữ liệu là một vấn đề nghiêm trọng đối với các hệ thống giám sát và thu thập dữ liệu trong hệ thống IoT. Nguyên nhân chủ yếu dẫn đến tình trạng này thường là do kết nối không ổn định giữa thiết bị và mạng, gây ra sự gián đoạn trong quá trình truyền tải dữ liệu. Điều này có thể gây sai lệch hoặc thậm chí mất hoàn toàn dữ liệu, ảnh hưởng tiêu cực đến khả năng theo dõi và phân tích của hệ thống. Để khắc phục vấn đề này, việc sử dụng thiết bị gateway trở thành một giải pháp hiệu quả. Gateway đóng vai trò là một lớp bảo vệ dự phòng, đảm bảo dữ liệu luôn được truyền tải một cách liên tục và an toàn, ngay cả khi kết nối giữa thiết bị và server gặp sự cố.

Gateway cung cấp khả năng sử dụng các phương thức truyền tải đáng tin cậy, cho phép hệ thống tiếp tục đẩy dữ liệu lên server ngay cả khi mạng chính bị gián đoạn. Khi gặp sự cố trong quá trình kết nối với server, dữ liệu có thể được gửi đến gateway, sau đó gateway sẽ tiếp tục xử lý và đẩy dữ liệu lên server, đảm bảo tính toàn vẹn và liên tục của dữ liệu. Điều này giúp giảm thiểu rủi ro mất mát dữ liệu và đảm bảo hoạt động ổn định của hệ thống IoT.

Với vai trò quan trọng của gateway trong hệ thống, thiết bị này cần được thiết kế để hoạt động liên tục, tự động và có khả năng xử lý khối lượng dữ liệu lớn được gửi đến đồng thời. Việc đảm bảo tính ổn định, tốc độ và tin cậy trong quá trình xử lý và truyền tải dữ liệu qua gateway là yếu tố quyết định để nâng cao hiệu suất và độ tin cậy của toàn bộ hệ thống thu thập dữ liệu.

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Nguyễn Tuấn Phong Số thẻ sinh viên: 105200463

Lớp: 20TDHCLC3 Khoa: Điện Ngành: Kĩ thuật điều khiển và tự động hoá

1. *Tên đề tài đồ án:*

   Thiết kế tổng đài tin nhắn SMS gateway cho mạng cảm biến không dây

2. *Đề tài thuộc diện:* ☐ *Có ký kết thoả thuận sở hữu trí tuệ đối với kết quả thực hiện*
3. *Các số liệu và dữ liệu ban đầu:*

Nội dung các phần thuyết minh và tính toán:

- Tìm hiểu các bài báo, nghiên cứu xây dựng giải pháp
- Xây dựng mô hình hoạt động cho hệ thống
- Xây dựng thiết kế lưu đồ thuật toán cho hệ thống
- Xây dựng các thư viện, chương trình giao tiếp với module SIM
- Xây dựng chương trình xử lý dữ liệu
- Thiết kế chương trình giám sát
- Thiết kế dashboard hệ thống
- Hoàn thiện và vận hành hệ thống
- Viết chương trình mô phỏng kiểm thử hoạt động hệ thống
- Làm slide báo cáo, viết báo cáo thuyết minh đồ án

4. *Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):*

5. Họ tên người hướng dẫn:	Phần/Nội dung:
TS. Ngô Đình Thanh	- Hướng dẫn quy trình thiết kế dự án doanh nghiệp

	- Hướng dẫn, tư vấn giải pháp công nghệ cho dự án - Yêu cầu dựu án - Thảo luận, tư vấn giải pháp - Đánh giá sản phẩm dự án - Hướng dẫn thuyết minh báo cáo

6. *Ngày giao nhiệm vụ đồ án:* *24/02/2025*
7. *Ngày hoàn thành đồ án:* *17/06/2025*

Đà nẵng, ngày tháng 06 năm 2025

Trưởng Bộ môn Tự động hoá **Người hướng dẫn**

TS. Giáp Quang Huy

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
ĐỘC LẬP – TỰ DO – HẠNH PHÚC

PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP

(Phiếu dành cho người hướng dẫn/sinh viên)

Họ và tên sinh viên Nguyễn Tuấn Phong Số thẻ SV: 105200463
Tên đề tài ĐATN: Thiết kế tổng đài tin nhắn SMS gateway cho mạng cảm biến không dây.
Họ tên người HD: TS. Ngô Đình Thanh Đơn vị: Khoa điện

Tuần	Ngày	Khối lượng		GVHD Ký tên
		Đã thực hiện	Tiếp tục thực hiện	
1	24/02/2025	Nhận đề tài	Nhận đề tài, tìm hiểu tổng quát đề tài	
2	10/03/2025	Tìm hiểu bài báo liên quan, xây dựng giải pháp cho vấn đề	Báo cáo giảng viên quá trình nghiên cứu, xin ý kiến về giải pháp thiết kế	
3	17/03/2025	Báo cáo giải pháp, thống nhất phương án thiết kế	Viết báo cáo chương 1	
4	24/03/2025	Duyệt lần 1: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN • Không tiếp tục thực hiện ĐATN •		
5	31/03/2025	Nghiên cứu bài báo giải pháp hiện nay	Thiết kế chi tiết giải pháp	
6	07/04/2025	Hoàn thiện thiết kế, xây dựng thành phần hệ thống	Hoàn thiện chi tiết giải pháp và thành phần hệ thống	
7	14/04/2025	Hoàn thiện thiết kế hệ thống	Viết báo cáo chương 2	
8	21/04/2025	Duyệt lần 2: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN • Không tiếp tục thực hiện ĐATN •		
9	28/04/2025	Thiết kế kiến trúc chương trình	Xây dựng chương trình hoạt động cho thiết bị	
10	05/05/2025	Hoàn thiện kiểm tra chương trình cho thiết bị	Xây dựng chương trình giám sát	
11	12/05/2025	Hoàn thiện chương trình kiểm tra hoạt	Viết báo cáo chương 3	

		động		
12	19/05/2025	Duyệt lần 3: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN • Không tiếp tục thực hiện ĐATN •		
13	26/05/2025	Hoàn thiện thiết bị, tiến hành thực nghiệm	Hiệu chỉnh chương trình Viết báo cáo chương 4	
14	02/06/2025	Tiến hành thực nghiệm hiệu chỉnh hoàn thiện sản phẩm	Tổng hợp tài liệu hoàn thiện báo cáo theo mẫu	
15	17/06/2025	Viết thuyết minh, hoàn thiện đề tài	Hoàn thiện thuyết minh	

LỜI NÓI ĐẦU VÀ CẢM ƠN

Trải qua một thời gian nghiên cứu và thực hành làm đồ án tốt nghiệp, mặc dù còn nhiều hạn chế về kiến thức lẫn kĩ năng chuyên môn, nhưng được sự giúp đỡ của rất nhiều người động viên và giúp đỡ thì cuối cùng em cũng hoàn thành xong đồ án tốt nghiệp.

Cho phép em gửi lời cảm ơn đến TS. Ngô Đình Thanh, người thầy đã hướng dẫn em trong suốt quá trình thực hiện đồ án tốt nghiệp, những điều thầy chia sẻ về kiến thức kỹ thuật và kỹ năng mềm đã giúp đỡ em rất nhiều trong quá trình thực hiện đồ án tốt nghiệp và sẽ là hành trang cho em trong suốt quá trình đi làm sau này.

Cuối cùng, em xin chân thành cảm ơn ban giám hiệu trường Đại học Bách Khoa - Đại học Đà Nẵng và quý thầy cô trong khoa Điện đã truyền đạt những bài học quý báu, những kinh nghiệm bổ ích và tạo điều kiện thuận lợi cho em hoàn thiện đồ án này.

LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT

Em xin cam đoan đề tài tốt nghiệp: “Thiết kế tổng đài tin nhắn SMS gateway cho mạng cảm biến không dây” là một dự án nghiên cứu độc lập của em dưới sự hướng dẫn của giáo viên hướng dẫn TS. Ngô Đình Thanh. Ngoài ra không có bất kì sự sao chép nào trong việc thực hiện. Sản phẩm, nội dung báo cáo tốt nghiệp là quá trình mà em nỗ lực nghiên cứu và thực hiện dưới sự hỗ trợ từ TS. Ngô Đình Thanh. Các số liệu, kết quả trình bày trong báo cáo là hoàn toàn trung thực, em xin chịu toàn bộ trách nhiệm, kỷ luật của bộ môn và nhà trường nếu như có vấn đề về liêm chính học thuật xảy ra.

Sinh viên thực hiện

Nguyễn Tuấn Phong

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC CÁC BẢNG

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

IoT	Internet of Things
4G	Four Generation
HTTP	Hypertext Transfer Protocol I2C Integrated Circuit
UART	Universal Asynchronous Receiver Transmitter
WiFi	Wireless Fidelity
WSN	Wireless Sensor Network
SMS	Short Message Service
MQTT	Message Queuing Telemetry Transport

MỞ ĐẦU

1. LÝ DO CHỌN ĐỀ TÀI

Hiện nay, các hệ thống thu thập dữ liệu đang ngày càng được chú trọng bởi tính liên tục và chính xác của dữ liệu thời gian thực. Những hệ thống có khả năng cập nhật dữ liệu liên tục mà không bị gián đoạn, cũng như hoạt động một cách ổn định, thường được đánh giá rất cao trong các ứng dụng thực tế. Trong bối cảnh đó, các hệ thống cảm biến không dây thu thập dữ liệu lớn điển hình với hệ thống khí tượng thuỷ văn, với mục tiêu thu thập dữ liệu về lượng mưa từ hàng nghìn trạm trên khắp cả nước. Số lượng trạm đo mưa đã lên tới hàng ngàn trạm , được thiết lập và lắp đặt ở nhiều khu vực, bao gồm cả những vùng núi cao và những nơi có kết nối mạng yếu. Những khu vực này không chỉ gặp khó khăn trong việc duy trì kết nối liên tục mà còn phải đối mặt với các điều kiện thời tiết khắc nghiệt như mưa bão, có thể ảnh hưởng nghiêm trọng đến khả năng thu thập dữ liệu và dẫn đến việc mất thông tin quan trọng.

Vấn đề này đặt ra một thách thức lớn cho trong việc đảm bảo tính tin cậy của hệ thống, cũng như duy trì kết nối liên tục để giữ cho dữ liệu luôn được cập nhật. Để giải quyết vấn đề cấp thiết này, nhóm nghiên cứu đã tiến hành thiết kế và phát triển một thiết bị gateway, nhằm bảo đảm rằng việc duy trì kết nối giữa các thiết bị sensor và server luôn được duy trì một cách liên tục. Thiết bị gateway này không chỉ giúp tăng cường khả năng kết nối mà còn cung cấp khả năng giám sát và phân tích cường độ kết nối giữa các thiết bị.

Một trong những ưu điểm nổi bật của thiết bị gateway là khả năng giám sát quá trình hoạt động của hệ thống. Qua việc đánh giá lượng dữ liệu gửi về gateway, nhóm nghiên cứu có thể xác định được tình trạng kết nối của các thiết bị sensor, đặc biệt khi các thiết bị này mất kết nối trực tiếp với server. Thiết bị gateway sẽ lưu trữ tạm thời dữ liệu trong trường hợp mất kết nối, sau đó tự động đẩy dữ liệu lên server ngay khi kết nối được khôi phục. Điều này không chỉ giúp giảm thiểu nguy cơ mất dữ liệu mà còn cải thiện hiệu suất tổng thể của hệ thống, đảm bảo rằng tất cả thông tin quan trọng đều được lưu trữ và có thể truy cập một cách dễ dàng.

2. MỤC TIÊU NGHIÊN CỨU

Mục tiêu của nghiên cứu này là tối ưu hóa quá trình vận hành của hệ thống IoT và đảm bảo rằng dữ liệu được truyền tải liên tục và chính xác lên cơ sở dữ liệu, ngay cả trong những điều kiện kết nối không ổn định. Để đạt được mục tiêu này, nghiên cứu sẽ

tập trung vào việc phân tích các yếu tố ảnh hưởng đến tín hiệu và khả năng kết nối của thiết bị trong môi trường thực tế. Những yếu tố này có thể bao gồm tín hiệu yếu, mất kết nối do thời tiết xấu, hoặc địa hình phức tạp, tất cả đều có thể gây trở ngại cho quá trình thu thập và truyền tải dữ liệu.

Trên cơ sở các phân tích đó, nghiên cứu sẽ đề xuất một thuật toán tối ưu cho thiết bị gateway, nhằm giúp quản lý quá trình truyền tải dữ liệu một cách linh hoạt và an toàn. Việc thiết kế và triển khai thiết bị gateway sẽ không chỉ tập trung vào việc đảm bảo kết nối liên tục mà còn chú trọng đến việc giám sát hiệu quả hoạt động của toàn hệ thống. Điều này bao gồm khả năng theo dõi tình trạng kết nối của các thiết bị sensor, đánh giá cường độ tín hiệu, và xác định kịp thời các sự cố có thể xảy ra.

Sau khi thiết bị gateway được triển khai, quá trình thử nghiệm sẽ diễn ra để đánh giá hiệu quả của giải pháp đề xuất. Qua các thử nghiệm thực tế, nhóm nghiên cứu sẽ xác định tính khả thi và độ tin cậy của thiết bị trong các tình huống cụ thể, từ đó đưa ra các điều chỉnh cần thiết nhằm cải thiện hiệu suất hoạt động của hệ thống. Mục tiêu cuối cùng là xây dựng một hệ thống IoT mạnh mẽ, có khả năng hoạt động ổn định và hiệu quả, đồng thời cung cấp dữ liệu chính xác và kịp thời cho người vận hành, giúp họ có những quyết định chính xác trong công tác quản lý và giám sát.

3. ĐỐI TƯỢNG, PHẠM VI NGHIÊN CỨU

Đề tài nghiên cứu tập trung vào việc phân tích và phát triển hệ thống thu thập dữ liệu từ hệ thống cảm biến không dây, một phần quan trọng trong việc giám sát và thu thập thông tin từ môi trường thực tế. Đối tượng nghiên cứu chính là các thiết bị sensor này và hệ thống dữ liệu liên quan, với nhiệm vụ thu thập, truyền tải, và xử lý dữ liệu cảm biế. Sensor chịu trách nhiệm ghi lại các thông số lượng mưa sau đó truyền dữ liệu thu thập được về hệ thống trung tâm để xử lý và lưu trữ.

Mục tiêu chính của nghiên cứu là đảm bảo tính liên tục và chính xác của dữ liệu thu thập được, bất chấp các yếu tố ngoại cảnh có thể gây ra gián đoạn trong quá trình truyền tải. Điều này đòi hỏi phải tích hợp chặt chẽ giữa thiết bị sensor và hệ thống server. Một trong những thách thức lớn trong việc thu thập dữ liệu từ xa là sự gián đoạn kết nối mạng, điều này có thể xảy ra ở các khu vực có điều kiện mạng không ổn định hoặc khi các thiết bị hoạt động trong môi trường khắc nghiệt. Nghiên cứu này tập trung vào việc tìm kiếm giải pháp cho các vấn đề liên quan đến mất kết nối và gián đoạn dữ liệu, nhằm đảm bảo rằng toàn bộ dữ liệu thu thập được đều được truyền tải một cách liên tục và không bị mất mát.

Một phần quan trọng của nghiên cứu là thiết kế và phát triển một thiết bị gateway, đóng vai trò là một cầu nối trung gian giữa các thiết bị sensor và hệ thống server. Thiết

bị gateway sẽ giúp lưu trữ tạm thời dữ liệu trong trường hợp mất kết nối với server, sau đó tự động đẩy dữ liệu lên khi kết nối được khôi phục. Điều này đảm bảo rằng dù trong điều kiện kết nối mạng kém, dữ liệu vẫn được thu thập và xử lý mà không bị mất mát hay sai lệch. Ngoài ra, thiết bị gateway cũng đóng vai trò trong việc giám sát và báo cáo tình trạng hoạt động của hệ thống, giúp người vận hành dễ dàng theo dõi tình trạng thiết bị và đưa ra các biện pháp khắc phục kịp thời nếu có sự cố xảy ra.

Phạm vi nghiên cứu của đề tài không chỉ dừng lại ở việc tích hợp thiết bị mà còn mở rộng đến việc xây dựng một hệ thống giám sát hiệu quả. Hệ thống này sẽ cung cấp các thông tin chi tiết về tình trạng hoạt động của các thiết bị sensor, dung lượng lưu trữ còn lại, và mức tiêu thụ năng lượng của hệ thống. Qua đó, người vận hành có thể nắm bắt được tình hình thực tế và đánh giá chính xác tình trạng của hệ thống, từ đó đưa ra các quyết định tối ưu trong việc quản lý và vận hành.

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1.1. Thách thức trong wireless sensor network

Mạng cảm biến không dây WSN đang ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực như y tế, môi trường, quân sự, và công nghiệp nhờ khả năng thu thập dữ liệu từ các vùng địa lý rộng lớn một cách linh hoạt và tiết kiệm chi phí, dễ dàng quản lý hoạt động và cho thông tin chính xác. Tuy nhiên, WSN đối mặt với rất nhiều thách thức và vấn đề kỹ thuật cũng như xã hội cần được giải quyết để đảm bảo hoạt động hiệu quả và bền vững. Một trong những vấn đề quan trọng nhất là kiểm soát năng lượng, bởi các cảm biến thường hoạt động bằng pin với dung lượng hạn chế, việc tiêu hao năng lượng quá nhanh sẽ ảnh hưởng trực tiếp đến tuổi thọ mạng và tính liên tục của dữ liệu. Các cơ chế quản lý năng lượng hiệu quả như điều chỉnh công suất truyền, chế độ ngủ đông (sleep mode), và cân bằng tải năng lượng được xem là những giải pháp quan trọng để kéo dài tuổi thọ của mạng được trích dẫn trong bài báo *"A Survey on Power Control Issues in Wireless Sensor Networks, IEEE 2007"* [1].

Ngoài vấn đề năng lượng, bảo mật trong WSN cũng là một thách thức lớn do đặc điểm mạng phân tán, kết nối không dây dễ bị tấn công từ bên ngoài và các vấn đề bảo mật khác. Các giao thức bảo mật truyền thống khó áp dụng trực tiếp vì giới hạn về tài nguyên phần cứng và phần mềm của các nút cảm biến. Do đó, thiết kế các giải pháp bảo mật, hiệu quả, đồng thời đảm bảo tính toàn vẹn và bảo mật dữ liệu trong quá trình truyền tải là rất cần thiết để bảo vệ mạng khỏi các tấn công như giả mạo, tấn công làm nghẽn đường truyền, từ chối dịch vụ, hay truy cập trái phép đã được đề cập ở bài báo *"Secure wireless sensor networks problems"* [2].

Về khía cạnh xã hội, khi WSN được ứng dụng trong lĩnh vực y tế, đặc biệt trong giám sát sức khỏe và chăm sóc bệnh nhân, các vấn đề về quyền riêng tư và bảo mật dữ liệu cá nhân trở nên cấp thiết hơn bao giờ hết. *"Social Issues in Wireless Sensor Networks with Healthcare Perspective"* [3] mô tả việc thu thập, lưu trữ và truyền tải thông tin sức khỏe nhạy cảm đòi hỏi phải có các chính sách và kỹ thuật bảo vệ nghiêm ngặt để tránh lộ lọt dữ liệu hoặc dữ liệu không đến đúng đích, gây mất dữ liệu hoặc dữ liệu bị dùng không đúng cách. Đồng thời, sự chấp nhận và tin tưởng của người dùng cũng phụ thuộc vào khả năng đảm bảo an toàn thông tin cá nhân, làm nổi bật tầm quan trọng của việc thiết kế các hệ thống WSN thân thiện với người dùng và tuân thủ các quy định pháp luật.

Ngoài các vấn đề về năng lượng, bảo mật và xã hội, WSN còn gặp phải các khó khăn về thiết kế mạng như khả năng mở rộng, độ tin cậy trong truyền thông và khả năng tự phục hồi khi xảy ra lỗi hoặc một vài nút sensor gặp vấn đề không thể phục hồi và thực hiện lại hoạt động. Với nghiên cứu của Ibrahim cùng động sự ở *"Challenges and Issues for Wireless Sensor Networks"* [4], các vấn đề về truyền thông thường gặp do các nút cảm biến có thể bị hỏng hoặc cạn kiệt năng lượng bất ngờ, việc xây dựng các giao thức định tuyến thông minh, có khả năng phát hiện và xử lý sự cố để duy trì tính toàn vẹn của mạng là rất quan trọng cùng với đó là việc cần phải biết nút nào gặp sự cố để có phương án khắc phục sự cố. Các yếu tố môi trường như nhiễu sóng, chướng ngại vật vật lý cũng ảnh hưởng đến chất lượng liên lạc và độ chính xác của dữ liệu thu thập.

Bài báo *"Challenges and Issues for Wireless Sensor Networks"* [4] cũng chỉ ra rằng việc tích hợp WSN với các công nghệ mới như IoT (Internet of Things) làm tăng thêm độ phức tạp của hệ thống, đòi hỏi các giải pháp đồng bộ về phần cứng, phần mềm, cũng như các giao thức truyền thông để đảm bảo tương thích và hiệu quả hoạt động trong môi trường đa dạng và phức tạp, việc quản lý hoạt động và giám sát theo dõi hệ thống cũng bị đặt thêm thách thức. Các yêu cầu về khả năng xử lý dữ liệu thời gian thực, khả năng tự động hóa cao và sự ổn định trong môi trường thực tế cũng là những thách thức đáng kể đối với các nhà phát triển WSN hiện nay.

Qua đó, để phát triển và ứng dụng mạng cảm biến không dây một cách hiệu quả, cần có sự phối hợp toàn diện giữa nhiều lĩnh vực, kỹ thuật để tối ưu hóa năng lượng và bảo mật, khoa học xã hội để đảm bảo quyền riêng tư và sự chấp nhận của người dùng, cũng như công nghệ để xây dựng các giải pháp mạng linh hoạt, mở rộng và ổn định, dễ dàng giám sát và thực hiện vận hành. Những thách thức này đòi hỏi nghiên cứu sâu rộng và phát triển các giải pháp sáng tạo nhằm khai thác tối đa tiềm năng của WSN trong tương lai.

1.2. Vấn đề mất kết nối trong wireless sensor network

Trong quá trình triển khai các hệ thống IoT, vấn đề kết nối luôn là vấn đề thiết yếu. Việc các hệ thống IoT hoạt động dựa trên sự phối hợp giữa hàng trăm đến hàng ngàn node cảm biến đòi hỏi phải xây dựng cơ sở hạ tầng kết nối nhằm đảm bảo tính liên tục của hệ thống và hạn chế kết nối gián đoạn. Theo bài viết *"Wireless Sensor Network Challenges and Solutions"* [5] của Analog Devices (2012). Trong số các nguyên nhân gây ra việc mất ổn định kết nối trong hệ thống mạng sensor, nguyên nhân nhiễu sóng đối với các thiết bị wifi, bluetooth làm mất các gói tin trong quá trình truyền tải dữ liệu là thường gặp nhất.

Bên cạnh đó, số lượng phần cứng của các node cảm biến cũng dẫn đến việc thiếu ổn định kết nối mạng. Như được trình bày trong bài báo *"Wireless Sensor Networks (WSNs): Security and Privacy Issues and Solutions"* [6] của IntechOpen (2020), các cảm biến thường chỉ có dung lượng bộ nhớ thấp, khả năng xử lý giới hạn và năng lượng pin rất ít do đó không thể duy trì kết nối liên tục hoặc sử dụng các giao thức mạng phức tạp. Vì thế nó ảnh hưởng đến việc duy trì kết nối dài hạn, đặc biệt là đối với hệ thống triển khai ngoài trời.

Ngoài ra, cấu trúc mạng cũng là yếu tố ảnh hưởng đến mạng cảm biến do sự thay đổi thường xuyên về vị trí vật lý hoặc nhiễu môi trường. Bài viết *"Khảo sát các vấn đề bảo mật trong mạng cảm biến không dây"* [7] đăng trên Tạp chí JSTIC (2020) đã chỉ rõ rằng việc thiếu một cơ chế phát hiện lỗi và tự điều chỉnh mạng (self-healing) là một rào cản lớn cho việc duy trì một hệ thống kết nối bền vững. Cấu trúc mạng phân tán khiến cho bất kỳ một sự cố nhỏ tại một node cũng có thể dẫn đến hiệu ứng lan truyền gây mất liên kết đến nhiều node khác, dẫn đến hiện tượng mất vùng phủ sóng (coverage hole) hoặc mất kênh truyền dữ liệu.

Mạng cảm biến còn gây khó trong khả năng mở rộng và duy trì đồng bộ dữ liệu đặc biệt khi số lượng node tăng lên, lưu lượng dữ liệu trong mạng cũng tăng theo cấp số nhân, làm phát sinh hiện tượng tắc nghẽn, giảm thông lượng và tăng độ trễ. Điều này đặc biệt nguy hiểm đối với các ứng dụng yêu cầu thời gian thực như giám sát sức khỏe hoặc hệ thống cảnh báo sớm. Các cơ chế quản lý luồng dữ liệu và cân bằng tải giữa các node là cần thiết, nhưng lại khó triển khai hiệu quả trên nền tảng phần cứng giới hạn.

1.3. Giải pháp hiện nay cho wireless sensor network

Một trong những thách thức hàng đầu trong quá trình triển khai và vận hành các hệ thống IoT (Internet of Things) là đảm bảo kết nối ổn định và liên tục. Trong đó, khả năng duy trì kết nối là đầu mối quan trọng để đảm bảo mạng lưới các thiết bị có thể truyền tải và xử lý dữ liệu chính xác. Bất kỳ sự gián đoạn nào trong kết nối đều có thể gây ra những hậu quả nghiêm trọng, làm trì trệ các hoạt động, gây mất mát dữ liệu quan trọng và ảnh hưởng trực tiếp đến hiệu suất của hệ thống.

Ngoài ra, vấn đề nổi cộm trong việc đảm bảo kết nối còn là khả năng duy trì sự ổn định trong môi trường hoạt động đa dạng. Các thiết bị IoT thường được triển khai tại các địa điểm khác nhau, từ các khu vực đô thị phát triển với hạ tầng kết nối hiện đại, đến những vùng sâu, vùng xa, nơi mà điều kiện mạng còn nhiều hạn chế. Trong các khu vực này, các thiết bị IoT phải đối mặt với hàng loạt thách thức như tín hiệu yếu, độ trễ cao, hoặc thậm chí mất kết nối hoàn toàn. Đặc biệt, trong những môi trường có điều kiện thời tiết khắc nghiệt như mưa lớn, bão, thời tiết cực đoan cũng sẽ ảnh hưởng trực tiếp

đến tín hiệu cũng như cường độ kết nối, nguy cơ gián đoạn kết nối càng trở nên trầm trọng hơn. Những điều kiện này có thể làm suy giảm nghiêm trọng chất lượng tín hiệu, dẫn đến việc thu thập dữ liệu không đầy đủ hoặc bị gián đoạn, ảnh hưởng đến khả năng đưa ra quyết định kịp thời và chính xác dựa trên dữ liệu.

Trong bài báo *"Connected Coverage Assurance for Sensor Scheduling in Wireless Sensor Networks"* [8] của Attapol Adulyasas (2015), tác giả đã chỉ ra rằng một trong những thách thức quan trọng nhất đối với mạng cảm biến không dây (WSN) là duy trì kết nối mạng trong khi vẫn đảm bảo vùng phủ sóng và kéo dài tuổi thọ năng lượng. Các mạng WSN thường bao gồm nhiều cảm biến hoạt động bằng pin, và việc để toàn bộ cảm biến hoạt động liên tục không chỉ tiêu tốn năng lượng mà còn có thể dẫn đến mất kết nối khi các nút chết dần theo thời gian.

Để giải quyết vấn đề này, bài báo đề xuất một giải pháp lập lịch cảm biến tối ưu dựa trên mô hình "connected set cover" – một tập hợp cảm biến vừa đảm bảo phủ sóng khu vực giám sát, vừa duy trì khả năng kết nối liên tục giữa các nút và đến trạm thu thập dữ liệu. Giải pháp này cho phép hệ thống luân phiên kích hoạt các cảm biến theo từng chu kỳ nhằm tiết kiệm năng lượng, đồng thời vẫn duy trì toàn vẹn kết nối mạng. Tác giả cũng nhấn mạnh rằng việc chỉ kích hoạt các cảm biến cần thiết trong mỗi chu kỳ, trong khi các nút khác ở trạng thái ngủ, sẽ tối ưu hóa mức tiêu thụ năng lượng và kéo dài thời gian hoạt động của mạng.

Bài báo còn phát triển một thuật toán lập lịch cảm biến có tính đến cả hai tiêu chí: vùng phủ đầy đủ và duy trì kết nối. Điều này giúp tránh hiện tượng mất vùng giám sát hoặc các nút bị cô lập do thiếu đường truyền dữ liệu. Giải pháp của Adulyasas được xem là thiết thực trong các ứng dụng giám sát dài hạn không người trực, nơi mà khả năng tự duy trì kết nối và tiết kiệm năng lượng là cực kỳ quan trọng.

Ngoài yếu tố môi trường, sự gia tăng đáng kể về số lượng thiết bị IoT trong các hệ thống lớn cũng tạo ra áp lực lớn đối với hạ tầng kết nối. Trong các hệ thống IoT quy mô lớn, khi hàng nghìn, thậm chí hàng triệu thiết bị hoạt động đồng thời và truyền tải dữ liệu, băng thông mạng có thể bị quá tải. Hiện tượng tắc nghẽn kết nối không chỉ làm giảm tốc độ truyền tải dữ liệu mà còn có thể dẫn đến gián đoạn kết nối hoàn toàn. Điều này đặc biệt nguy hiểm trong các ứng dụng IoT đòi hỏi khả năng hoạt động thời gian thực, nơi mà một khoảng trễ nhỏ trong quá trình truyền tải dữ liệu có thể gây ra những sự cố nghiêm trọng, ảnh hưởng trực tiếp đến các quyết định vận hành hoặc làm mất đi khả năng phản ứng nhanh trước các tình huống khẩn cấp.

Thêm vào đó, vấn đề đảm bảo kết nối còn trở nên phức tạp hơn khi các hệ thống IoT thiếu các giải pháp dự phòng hiệu quả. Trong trường hợp mất mạng viễn thông tạm

thời, nếu không có phương án thay thế, sẽ làm mất dữ liệu cục bộ tại một số trạm trong thời gian xảy ra sự cố . Điều này đặc biệt ảnh hưởng lớn đối với các ứng dụng IoT mang tính chất nhạy cảm, như giám sát sức khỏe từ xa, quản lý an ninh, theo dõi tình trạng thời tiết, các hệ thống dự đoán sự cố hoặc vận hành các thiết bị quan trọng trong môi trường công nghiệp. Khi không có kết nối liên tục, dữ liệu quan trọng không thể được truyền tải kịp thời, dẫn đến mất mát thông tin và có thể gây ra thiệt hại lớn về kinh tế cũng như làm giảm uy tín của tổ chức vận hành hệ thống IoT.

1.4. Tổng kết.

Mạng cảm biến không dây (WSN) đóng vai trò cốt lõi trong hệ sinh thái IoT hiện đại, tuy nhiên quá trình triển khai và vận hành hệ thống này vẫn đang đối mặt với hàng loạt thách thức đáng kể. Trước tiên, giới hạn năng lượng của các node cảm biến là trở ngại lớn nhất ảnh hưởng đến tuổi thọ và độ tin cậy của toàn mạng, đặc biệt trong các môi trường khắc nghiệt và khó tiếp cận. Bên cạnh đó, vấn đề truyền thông không ổn định do nhiễu sóng, mất gói tin và trễ cao gây ảnh hưởng nghiêm trọng đến các ứng dụng thời gian thực. Về mặt bảo mật, hạn chế về phần cứng khiến các node dễ bị tấn công và không thể triển khai các thuật toán mã hóa mạnh. Khả năng mở rộng của mạng cũng bị hạn chế bởi hiện tượng "nghẽn cổ chai" tại các nút trung tâm, trong khi yếu tố môi trường vật lý tiếp tục đe dọa tính bền vững của thiết bị.

Đặc biệt, mất kết nối là một trong những vấn đề trọng yếu trong quá trình vận hành mạng cảm biến. Nguyên nhân chủ yếu đến từ nhiễu tín hiệu, giới hạn phần cứng, cấu trúc mạng phân tán thiếu khả năng tự phục hồi, và sự quá tải khi số lượng node tăng cao. Những yếu tố này không chỉ ảnh hưởng đến chất lượng dịch vụ mà còn gây mất dữ liệu, gián đoạn hoạt động, và làm giảm hiệu quả tổng thể của hệ thống.

Trước thực trạng này, các giải pháp hiện nay tập trung vào việc cải thiện tính ổn định kết nối trong điều kiện môi trường đa dạng, xây dựng kiến trúc mạng linh hoạt, và bổ sung các cơ chế dự phòng dữ liệu để đảm bảo hoạt động liên tục ngay cả khi kết nối bị gián đoạn tạm thời.

CHƯƠNG 2: ĐỀ XUẤT GIẢI PHÁP

2.1 . Đề xuất giải pháp.

Trong kiến trúc của một mạng cảm biến không dây (Wireless Sensor Network - WSN), gateway là thành phần kết nối các node cảm biến với mạng diện rộng hoặc đám mây hay edge server. Trong đó, chức năng chính của gateway bao gồm thu thập, lọc, tổng hợp, và chuyển tiếp dữ liệu vì vậy việc lựa chọn và thiết kế gateway ảnh hưởng trực tiếp đến hiệu năng toàn mạng, đặc biệt là về mặt năng lượng, độ trễ, độ tin cậy và khả năng mở rộng. Do đó, nghiên cứu các giải pháp tối ưu hóa gateway đang là một hướng đi quan trọng trong phát triển mạng WSN hiện đại.

Một trong những xu hướng nổi bật hiện nay là ứng dụng công nghệ blockchain kết hợp học sâu (deep learning) trong thiết kế gateway thông minh. Nghiên cứu của *Saranya và Sudha (2025)* [9] đề xuất một mô hình định tuyến WSN sử dụng thuật toán BCOA để lựa chọn nút cluster head, đồng thời tích hợp blockchain nhằm tăng cường bảo mật dữ liệu giữa các nút và gateway. Giải pháp này không chỉ tối ưu đường truyền mà còn tăng khả năng chống giả mạo và đảm bảo toàn vẹn dữ liệu truyền qua gateway.

Trong lĩnh vực nông nghiệp chính xác, gateway WSN tích hợp điện toán biên (edge computing) giúp giảm tải dữ liệu truyền về server trung tâm và xử lý thông tin tại chỗ nhanh hơn. *Hoque et al. (2024)* [10] triển khai một hệ thống gateway có khả năng tổng hợp và phân tích dữ liệu tại vườn cây hoặc ruộng lúa, giúp nông dân điều chỉnh hệ thống tưới tiêu theo thời gian thực. Việc tích hợp edge computing vào gateway giúp giảm độ trễ, tiết kiệm năng lượng và đảm bảo khả năng hoạt động độc lập ngay cả khi mất kết nối mạng.

Một loạt nghiên cứu gần đây khai thác công nghệ LoRa/LoRaWAN như một giải pháp gateway lý tưởng cho WSN ở vùng sâu, vùng xa. Ví dụ, *Pires và Gomes (2024)* [11] đã triển khai gateway LoRaWAN cho hệ thống giám sát chất lượng nước sông, cho phép thu thập dữ liệu từ khoảng cách xa (hàng km) với mức tiêu thụ năng lượng cực thấp. Các thiết bị cảm biến được truyền về một gateway trung tâm rồi lên đám mây phân tích. LoRaWAN đặc biệt phù hợp với các hệ thống WSN quy mô lớn, cần hoạt động lâu dài nhờ pin.

Ngoài ra, chi phí vận hành và duy trì các giải pháp dự phòng cũng là thách thức quan trọng khác liên quan đến việc đảm bảo kết nối. Trên thực tế hiện nay, tại nhiều khu vực, chất lượng kết nối không được đảm bảo, đặc biệt ở những nơi cơ sở hạ tầng mạng chưa phát triển. Để khắc phục tình trạng này, cần phải áp dụng nhiều phương thức dự

phòng nhằm duy trì kết nối liên tục trong đó triển khai dự phòng để đảm bảo quá trình kết nối là một phương pháp được các tổng đài gateway SMS của các nhà mạng sử dụng. Đây là một phương thức dự phòng hiệu quả, giúp đảm bảo quá trình truyền và thu thập dữ liệu ngay cả khi kết nối chính gặp sự cố. Bằng việc sử dụng gateway SMS dự phòng, hệ thống có thể khắc phục được gần như triệt để việc mất dữ liệu khi kết nối 3G/4G hoạt động không hiệu quả. Tuy vậy, vấn đề được đặt ra sau đó còn liên quan đến chi phí duy trì khi vận hành hệ thống.

Qua quá trình nghiên cứu và khảo sát thực tế, nhóm nhận thấy rằng việc phát triển một thiết bị gateway độc lập sử dụng phương thức SMS làm giải pháp kết nối dự phòng là một hướng đi tiềm năng. Thiết bị này có thể được thiết kế chuyên biệt để tối ưu hóa hiệu suất, phù hợp với nhu cầu thực tế của hệ thống. Giải pháp này không chỉ giúp đảm bảo kết nối liên tục ngay cả trong những điều kiện khắc nghiệt mà còn giảm thiểu sự phụ thuộc vào các dịch vụ nhà mạng, từ đó giúp tiết kiệm đáng kể chi phí vận hành và duy trì hệ thống trong dài hạn.

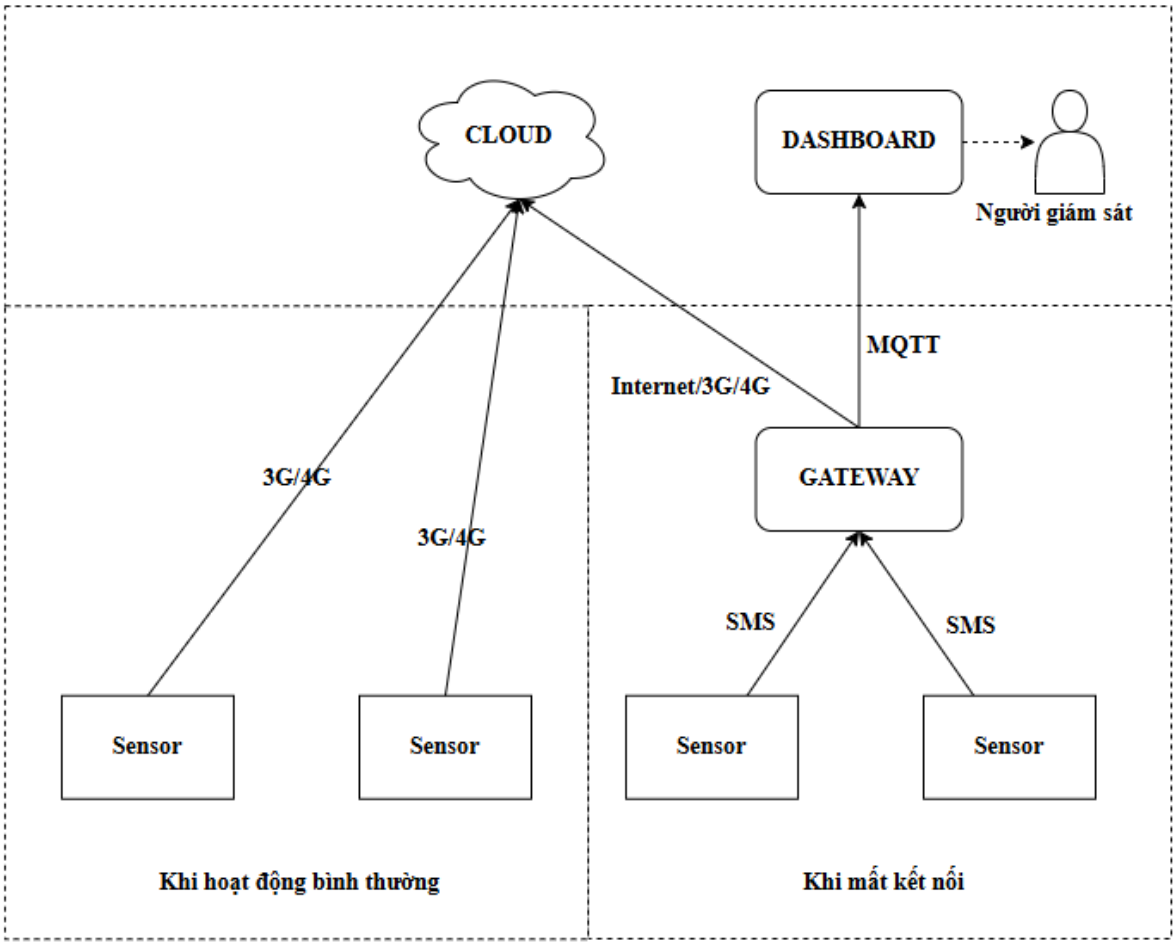
2.2 Phương án thiết kế

Để đảm bảo rằng quá trình vận hành của hệ thống hiện tại không bị ảnh hưởng, việc thiết kế hệ thống gateway mới cần được thực hiện một cách cẩn trọng và có tính toán kỹ lưỡng. Hệ thống gateway này sẽ được tích hợp vào hệ thống hiện tại với vai trò như một phương thức dự phòng, hỗ trợ hoạt động mà không làm gián đoạn hoặc thay đổi các quy trình hiện có.

Mục tiêu chính của việc thiết kế gateway là bổ sung tính năng đảm bảo kết nối liên tục và ổn định cho hệ thống, đặc biệt trong các trường hợp xảy ra sự cố kết nối chính. Gateway sẽ hoạt động song song với hệ thống hiện tại mà không can thiệp hoặc làm thay đổi cấu trúc và chức năng cốt lõi của hệ thống. Việc này không chỉ giúp duy trì sự ổn định trong quá trình vận hành mà còn giảm thiểu rủi ro gây ra bởi bất kỳ sự thay đổi nào đối với hệ thống hiện hành.

Hơn nữa, gateway được thiết kế sao cho có thể tích hợp dễ dàng mà không cần điều chỉnh quá nhiều các thành phần đã có trong hệ thống. Điều này nhằm tránh việc ảnh hưởng đến hiệu suất và độ tin cậy của hệ thống cũ. Đồng thời, các tính năng của gateway sẽ được tối ưu hóa để phù hợp với nhu cầu dự phòng, hỗ trợ quản lý dữ liệu hiệu quả hơn trong những trường hợp mạng chính gặp sự cố.

Dựa trên các yêu cầu và mục tiêu trên, nhóm đã đề xuất một kiến trúc vận hành cho hệ thống mới. Kiến trúc này không chỉ đảm bảo sự tích hợp mượt mà giữa gateway và hệ thống hiện tại mà còn nâng cao khả năng hoạt động tổng thể, giúp hệ thống đạt được độ tin cậy cao hơn trong quá trình thu thập và truyền tải dữ liệu.



Hình 2.1 Kiến trúc vận hành hệ thống

Quá trình hoạt động của hệ thống diễn ra theo các bước tuần tự và được thiết kế để đảm bảo tính ổn định, hiệu quả trong việc thu thập, xử lý và truyền tải dữ liệu.

Trước tiên, các thiết bị sensor sẽ thực hiện nhiệm vụ thu thập dữ liệu. Quá trình thu thập dữ liệu của sensor sẽ không bị thay đổi so với hệ thống hiện tại, với dữ liệu thu được, thay vì gửi dữ liệu đến tổng đài SMS của nhà mạng, dữ liệu được gửi đến gateway thông qua tin nhắn SMS. Phương thức SMS được lựa chọn vì tính ổn định cao và khả năng hoạt động tốt ngay cả ở những khu vực mà sóng di động yếu, đảm bảo dữ liệu có thể được truyền tải mà không bị gián đoạn cùng với đó việc sử dụng phương pháp này sẽ đảm bảo thay thế được gateway SMS của nhà mạng. Gateway sẽ nhận các tin nhắn SMS này từ các thiết bị sensor, đóng vai trò là trung gian kết nối dữ liệu từ các điểm thu thập tới hệ thống quản lý.

Khi tin nhắn SMS được gửi đến gateway, bước tiếp theo là giải mã dữ liệu từ các gói tin SMS đó. Mỗi tin nhắn SMS chứa các thông tin quan trọng như giá trị đo lường, thời gian thu thập, cùng các thông tin khác, tất cả đều được mã hóa theo một định dạng

cụ thể. Gateway sẽ tiến hành giải mã những gói tin này, xử lý và đảm bảo rằng dữ liệu được hiểu đúng và đầy đủ, chuẩn bị cho các bước xử lý tiếp theo.

Sau khi giải mã thành công, dữ liệu sẽ được truyền từ gateway lên hệ thống quản lý đám mây thông qua giao thức HTTPs. Quá trình này giúp lưu trữ và phân tích dữ liệu trên nền tảng đám mây, cho phép người dùng truy cập thông tin từ bất kỳ đâu với điều kiện có kết nối internet. Hệ thống đám mây không chỉ đóng vai trò là nơi lưu trữ mà còn là trung tâm phân tích và cung cấp các công cụ quản lý thông minh cho người vận hành.

Toàn bộ quá trình từ thu thập, giải mã cho đến truyền tải dữ liệu đều được thực hiện trong thời gian thực. Điều này đảm bảo rằng dữ liệu từ các thiết bị sensor được cập nhật liên tục, giúp hệ thống có thể phản hồi kịp thời với bất kỳ thay đổi nào từ môi trường. Người vận hành có thể theo dõi toàn bộ quá trình hoạt động của gateway, bao gồm việc giám sát số lượng gói tin và dữ liệu mà gateway đang xử lý.

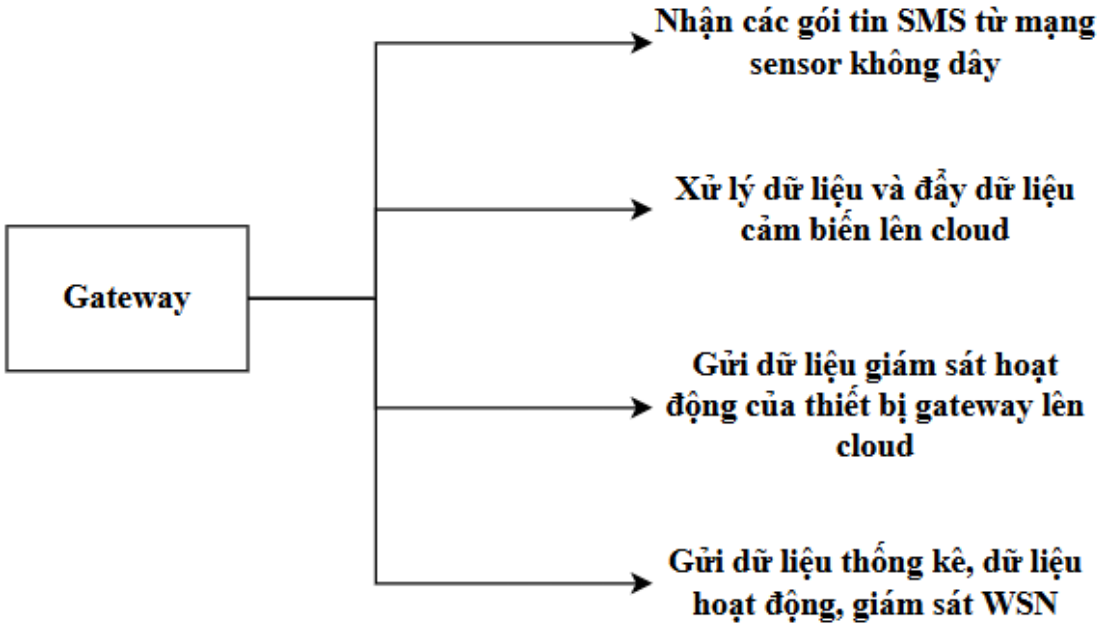
Hơn nữa, hệ thống giám sát cũng hỗ trợ quản lý toàn diện hoạt động của gateway. Người vận hành có thể kiểm tra trạng thái kết nối, theo dõi số lượng tin nhắn SMS đã nhận và xử lý, cùng lượng dữ liệu đã được gửi lên đám mây. Nhờ đó, họ có thể phát hiện sớm các sự cố trong quá trình truyền tải dữ liệu và áp dụng các biện pháp khắc phục kịp thời. Tất cả các yếu tố này phối hợp với nhau để đảm bảo tính ổn định và hiệu quả cao nhất cho toàn bộ hệ thống.

2.3 Xây dựng thành phần hệ thống

Với yêu cầu và chức năng hoạt động như một gateway độc lập có khả năng thay thế hoàn toàn chức năng của SMS gateway truyền thống, việc thiết kế và triển khai thiết bị đóng vai trò trung gian trong mạng cảm biến không dây (sensor network) đòi hỏi một quá trình nghiên cứu và xây dựng hệ thống bài bản. Thiết bị không chỉ đảm nhiệm vai trò thu thập dữ liệu từ các nút cảm biến phân tán, mà còn phải xử lý, định tuyến, và truyền tải dữ liệu đến hệ thống lưu trữ trung tâm hoặc máy chủ phân tích theo cách hiệu quả, ổn định và an toàn.

Qua đó xây dựng chức năng chính của gateway để phù hợp cho việc nhận và xử lý dữ liệu SMS nhận được từ các bộ sensor khi kết nối 3G/4G là kết nối chính của sensor gặp sự cố trong quá trình hoạt động. Để phù hợp với yêu cầu này, gateway cần có khả năng nhận đồng thời nhiều gói tin SMS từ sensor và có thể xử lý gói tin một cách chính xác và nhanh chóng. Ngoài ra việc kết hợp thêm hệ thống giám sát hoạt động của gateway sẽ cho phép người vận hành nắm được tính trạng hoạt động của thiết bị hiện tại. Cuối cùng tính năng thống kê và giám sát số lượng gói tin của gateway nhận được sẽ giúp góp phần phân tích và đánh giá được tình trạng hoạt động và kết nối của hệ thống. Khi có sự tăng đột biến về số lượng gói tin đến gateway, người vận hành có thể

đánh giá rằng việc kết nối chính của các sensor trong hệ thống đang gặp sự cố và cần phải đưa ra các giải pháp khắc phục kịp thời.



Hình 2.2 Chức năng chính của thiết bị gateway

Một điểm quan trọng trong thiết kế là đảm bảo tính tự động hóa cao. Khi được khởi động, gateway có thể khởi động, kết nối và bắt đầu thu thập, xử lý, truyền dữ liệu mà không cần bất kỳ thao tác cấu hình thủ công nào. Điều này đặc biệt hữu ích trong các hệ thống phân tán hoặc triển khai ở những nơi khó tiếp cận, hạn chế sự can thiệp của con người.

Ngoài ra, để nâng cao khả năng quản lý và tối ưu hóa hoạt động của toàn hệ thống, mỗi gateway cần được thiết kế như một thiết bị hoạt động độc lập, có thể giám sát trạng thái của chính nó và gửi dữ liệu lên nền tảng cloud theo định kỳ hoặc theo sự kiện. Việc này giúp đảm bảo tính toàn vẹn dữ liệu. Hạn chế tình trạng mất mát dữ liệu do gián đoạn kết nối. Tối ưu truyền tải, chỉ gửi các gói dữ liệu cần thiết, tránh dư thừa. Khả năng dự phòng, khi đường truyền chính gặp sự cố, thiết bị có thể chuyển sang truyền SMS như một phương thức thay thế.

Với chức năng quan trọng trong việc trung chuyển dữ liệu và điều phối hoạt động giữa các thiết bị đầu cuối và hệ thống cloud, gateway cần được xây dựng trên nền tảng phần cứng đủ mạnh để đáp ứng cả yêu cầu về xử lý dữ liệu và khả năng kết nối ổn định trong môi trường triển khai thực tế. Trong bối cảnh đó, việc lựa chọn Raspberry Pi làm bộ điều khiển trung tâm (controller) cho gateway là một giải pháp phù hợp và hiệu quả.

Bên cạnh phần xử lý, vấn đề kết nối từ gateway đến hệ thống cloud là yếu tố then chốt để đảm bảo luồng dữ liệu được truyền liên tục và ổn định. Để giải quyết bài toán này, việc tích hợp module SIM7600 – một module truyền thông 4G LTE – là lựa chọn hợp lý. SIM7600 hỗ trợ kết nối internet tốc độ cao qua mạng di động, giúp gateway có thể truyền dữ liệu trong thời gian thực ngay cả ở những khu vực không có sẵn kết nối Wi-Fi hoặc Ethernet, chẳng hạn như trong môi trường triển khai ngoài trời, vùng nông thôn hoặc các hệ thống di động. Ngoài ra, SIM7600 còn hỗ trợ nhắn tin SMS và định vị GNSS, mở ra khả năng tích hợp thêm các tính năng mở rộng như gửi cảnh báo qua tin nhắn hoặc theo dõi vị trí thiết bị.

Qua đó thông số phần cứng gateway:

Bộ Vi Xử Lý (CPU):	Quad-core Cortex-A76 với xung nhịp 2.4 GHz, một cải tiến lớn so với các phiên bản trước. Đây là bộ vi xử lý 64-bit mạnh mẽ, cung cấp hiệu năng vượt trội để thực hiện các phép toán phức tạp, hỗ trợ tốt các ứng dụng yêu cầu tính toán cao và khả năng xử lý đa nhiệm. Cortex-A76 đảm bảo khả năng đáp ứng nhanh, đặc biệt phù hợp với các hệ thống yêu cầu xử lý đồng thời nhiều dữ liệu trong thời gian ngắn.
Bộ Nhớ (RAM)	Gateway được trang bị RAM LPDDR4 với các tùy chọn 2GB, 4GB, và 8GB. RAM LPDDR4 cho tốc độ truy cập dữ liệu nhanh hơn, giảm thiểu tình trạng nghẽn khi chạy nhiều ứng dụng cùng lúc. Với lượng RAM lớn, thiết bị có thể thực hiện các tác vụ phức tạp, xử lý dữ liệu lớn, và đảm bảo khả năng vận hành mượt mà trong các ứng dụng thời gian thực và IoT.
Kết Nối Mạng	Cổng Ethernet Gigabit: Tích hợp cổng Ethernet tốc độ cao, hỗ trợ truyền dữ liệu lên đến 1000 Mbps. Điều này đảm bảo kết nối có dây ổn định, phù hợp với các ứng dụng yêu cầu độ trễ thấp và truyền tải dữ liệu lớn. Wi-Fi Băng Tần Kép: Thiết bị hỗ trợ Wi-Fi 2.4 GHz và 5 GHz, cung cấp kết nối không dây linh hoạt. Đặc biệt, Wi-Fi 5 GHz ít bị nhiễu sóng hơn và có băng thông cao hơn, phù hợp với các hệ thống IoT yêu cầu tốc độ truyền tải nhanh
Lưu Trữ	Thẻ microSD: Thiết bị sử dụng thẻ microSD cho hệ điều hành và lưu trữ dữ liệu chính. Với khả năng tùy chỉnh dung lượng linh hoạt, người dùng có thể mở rộng lưu trữ dễ dàng.

	Ổ Đĩa USB: Hỗ trợ kết nối với các ổ đĩa USB, cho phép mở rộng lưu trữ lớn hơn. Điều này rất cần thiết cho các hệ thống yêu cầu lưu trữ lâu dài hoặc xử lý dữ liệu lớn.
Tính Năng Nhận và Gửi SMS	Nhận SMS: Gateway tích hợp cho phép nhận tin nhắn từ mạng di động và lưu trữ tạm thời. Khi có tin nhắn mới, module sẽ thông báo đến hệ thống hoặc ứng dụng để xử lý. Gửi SMS: Gateway hỗ trợ gửi tin nhắn đến các số điện thoại qua mạng di động. Tin nhắn có thể được tạo và gửi từ ứng dụng trên gateway, đảm bảo tính chính xác và nhanh chóng. Quản Lý Tin Nhắn: Hỗ trợ lưu trữ tin nhắn trong bộ nhớ SIM hoặc bộ nhớ thiết bị, giúp quản lý dữ liệu lâu dài và thuận tiện trong truy xuất.
Tính năng quản lý dữ liệu SMS	Lưu trữ Tin Nhắn: Module SIM có khả năng lưu trữ tin nhắn SMS trong bộ nhớ bộ nhớ trong của thiết bị, cho phép lưu trữ và quản lý tin nhắn lâu dài. Tin Nhắn Văn Bản (Text Messages): Module SIM hỗ trợ định dạng tin nhắn văn bản tiêu chuẩn, cho phép gửi và nhận tin nhắn chứa văn bản thông thường. Tin Nhắn Dữ Liệu (PDU Mode): Đối với các tin nhắn cần mã hóa đặc biệt hoặc chứa dữ liệu không phải văn bản, module có thể hoạt động trong chế độ PDU (Protocol Data Unit), giúp xử lý và truyền tải tin nhắn dữ liệu.

Bảng 2.1 Thông số phần cứng gateway

2.4 Tổng kết

Với tầm quan trọng đặc biệt của việc đảm bảo kết nối liên tục và ổn định cho hệ thống, có thể nhận thấy rằng việc thiết kế và triển khai một thiết bị gateway là một giải pháp hiệu quả, mang lại nhiều lợi ích đáng kể trong vận hành. Thiết bị này không chỉ đóng vai trò là một phương thức dự phòng, giúp duy trì hoạt động ngay cả khi kết nối chính gặp sự cố, mà còn góp phần tối ưu hóa chi phí vận hành và quản lý hệ thống một cách hiệu quả hơn.

Giải pháp sử dụng gateway mang lại khả năng giám sát chặt chẽ quá trình hoạt động của hệ thống, từ đó giúp phát hiện và xử lý nhanh chóng các vấn đề tiềm ẩn liên quan đến kết nối. Bằng cách này, hệ thống có thể giảm thiểu tối đa các tình huống mất kết nối đột ngột, vốn là nguyên nhân chính gây ra sự gián đoạn trong quá trình thu thập

và truyền tải dữ liệu. Điều này đặc biệt quan trọng đối với các ứng dụng yêu cầu dữ liệu chính xác và liên tục, nơi mà bất kỳ sự gián đoạn nào cũng có thể gây ra những hậu quả nghiêm trọng đối với hiệu suất hoạt động và khả năng đưa ra quyết định kịp thời.

Tổng thể, việc thiết kế phần cứng cho thiết bị gateway đóng vai trò trung gian trong hệ thống mạng cảm biến không dây không chỉ là quá trình lựa chọn linh kiện phù hợp, mà còn là sự kết hợp chặt chẽ giữa khả năng thu thập dữ liệu, xử lý cục bộ, kết nối ổn định, và truyền tải dữ liệu linh hoạt. Với kiến trúc được xây dựng, thiết bị gateway không những có thể thay thế hoàn toàn các hệ thống SMS gateway truyền thống

Việc sử dụng Raspberry Pi làm nền tảng xử lý trung tâm kết hợp với module SIM7600 giúp đảm bảo thiết bị vừa có khả năng tính toán cần thiết, vừa đáp ứng được yêu cầu kết nối dữ liệu qua mạng di động trong nhiều điều kiện triển khai khác nhau. Bên cạnh đó, khả năng hoạt động độc lập, tự động khởi động và gửi dữ liệu lên cloud giúp gateway có thể vận hành bền bỉ, giảm thiểu sự phụ thuộc vào nguồn lực kỹ thuật tại hiện trường.

Như vậy, phần cứng của gateway không chỉ là nền tảng vật lý giúp thiết bị vận hành mà còn là yếu tố tiên quyết đảm bảo tính ổn định, toàn vẹn dữ liệu và hiệu quả truyền thông cho toàn bộ hệ thống sensor network. Kiến trúc phần cứng mạnh mẽ và linh hoạt là nền móng vững chắc cho các chức năng phần mềm được phát triển phía sau, từ đó đảm bảo sự thành công và tính ứng dụng rộng rãi của thiết bị gateway trong thực tế. Qua đó có thể thấy thiết bị gateway không chỉ là một giải pháp dự phòng để duy trì kết nối, mà còn là một công cụ quan trọng để tối ưu hóa hiệu quả và chi phí vận hành của hệ thống IoT. Việc phát triển giải pháp này không chỉ giúp giảm thiểu rủi ro và thiệt hại do mất kết nối mà còn góp phần cải thiện toàn diện hiệu suất và độ tin cậy của hệ thống, đáp ứng tốt hơn các yêu cầu ngày càng cao của thực tiễn vận hành.

CHƯƠNG 3: THIẾT KẾ PHẦN MỀM

3.1 Kiến trúc chương trình

Để xây dựng chương trình xử lý cho thiết bị gateway, kiến trúc phần mềm được chia thành hai phần chính là chương trình nhúng và chương trình giám sát. Việc phân tách này giúp tổ chức rõ ràng chức năng, tăng tính ổn định và khả năng mở rộng của hệ thống trong môi trường thực tế.

Chương trình chính được chạy trên thiết bị gateway, chịu trách nhiệm trực tiếp tương tác với phần cứng và thực hiện các task vụ xử lý của gateway. Thành phần này đảm nhiệm việc giao tiếp với module SIM thực hiện thu thập dữ liệu từ các cổng UART, đồng thời xử lý sơ bộ tín hiệu và phát hiện lỗi trước khi chuyển tiếp dữ liệu lên tầng cao hơn. Ngoài ra, chương trình cũng kiểm soát các hoạt động như khởi động hệ thống, giám sát nguồn, quản lý trạng thái thiết bị, và duy trì hoạt động ổn định liên tục trong thời gian dài mà không cần sự can thiệp của người dùng.

Bên cạnh đó, chương trình giám sát sẽ được thiết kế để có thể theo dõi và quản lý tính trạng hoạt động của thiết bị và hệ thống. Với việc xây dựng giao diện giám sát cho chương trình giám sát để thực hiện việc hiển thị các thông số hoạt động của thiết bị và đánh giá được tình trạng kết nối của hệ thống. Khi xảy ra sự cố trong kết nối của hệ thống, các thông số giám sát sẽ tăng giảm bất thường qua đó người vận hành xác định được vấn đề và có các phương pháp xử lý.

Sự phối hợp giữa chương trình nhúng và chương trình giám sát giúp thiết bị gateway có thể vận hành trơn tru, vừa đảm bảo hiệu năng xử lý ở cấp phần cứng, vừa thực hiện được các chức năng cao cấp phục vụ mục đích thu thập, xử lý và truyền tải dữ liệu trong hệ thống mạng cảm biến một cách thông minh và linh hoạt. Đây là mô hình phân lớp điển hình trong thiết kế phần mềm nhúng hiện đại, đặc biệt phù hợp với các hệ thống IoT và gateway thông minh.

Với chương trình chính cho thiết bị gateway, việc đảm bảo khả năng xử lý một số lượng lớn gói tin SMS đồng thời là một yêu cầu kỹ thuật vô cùng quan trọng và không thể bỏ qua. Hệ thống phải có khả năng tiếp nhận, xử lý nhanh chóng và chính xác tất cả các gói tin SMS được gửi từ các thiết bị sensor, bất kể số lượng hay tốc độ gửi của chúng. Để đáp ứng được yêu cầu này, hệ thống cần phải được trang bị hiệu suất xử lý cao, với khả năng thực hiện xử lý song song một cách hiệu quả. Điều này không chỉ đảm bảo rằng không có gói tin nào bị bỏ sót hoặc mất mát trong quá trình xử lý mà còn bảo đảm rằng tất cả dữ liệu thu thập được đều được xử lý một cách chính xác và đầy đủ.

Hệ thống cũng cần phải duy trì độ tin cậy cao trong suốt quá trình hoạt động liên tục, đặc biệt là trong các tình huống khẩn cấp hoặc khi khối lượng dữ liệu tăng đột biến. Điều này có nghĩa là hệ thống cần phải hoạt động ổn định và mượt mà, ngay cả khi khối lượng công việc tăng lên đột ngột, ví dụ như trong trường hợp có nhiều thiết bị sensor gửi dữ liệu. Để đạt được điều này, hệ thống cần phải được thiết kế với cấu trúc xử lý tối ưu, có khả năng mở rộng linh hoạt để đáp ứng các yêu cầu xử lý ngày càng phức tạp. Khả năng xử lý của hệ thống phải được phân bổ đồng đều và hợp lý để tránh tình trạng quá tải hoặc tắc nghẽn, từ đó duy trì tính ổn định và tính liên tục của toàn bộ hệ thống.

Ngoài ra, để đảm bảo khả năng tích hợp ngay và hoạt động trơn tru với các bộ sensor hiện tại đang hoạt động, một yêu cầu kỹ thuật quan trọng đối với gateway là phải có khả năng xử lý được nhiều cấu trúc gói tin khác nhau. Điều này rất cần thiết vì nó cho phép hệ thống thu thập và xử lý các gói tin dữ liệu từ các thiết bị sensor mà không cần phải thay đổi hoặc cập nhật lại cấu hình của toàn bộ hệ thống sensor hiện có. Khả năng này giúp hệ thống duy trì sự linh hoạt và tương thích với nhiều loại thiết bị sensor khác nhau, từ đó đảm bảo rằng quá trình triển khai hệ thống sẽ không gặp phải những gián đoạn không cần thiết.

3.2 Chương trình gateway

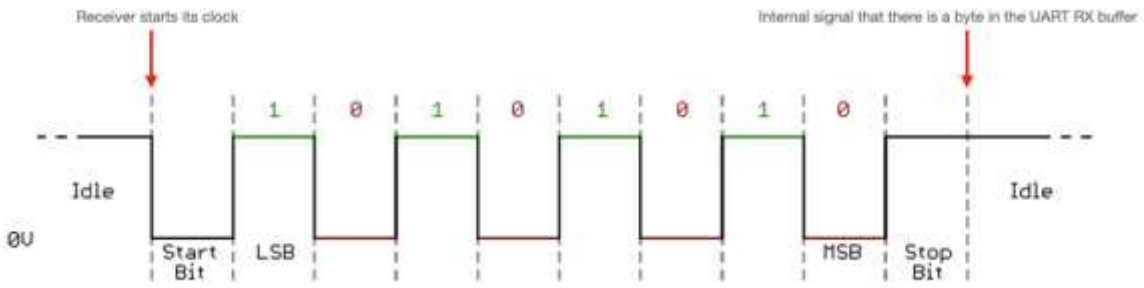
3.2.1 Thiết lập môi trường

(a) Khởi tạo giao tiếp UART

Để thực hiện giao tiếp giữa các thiết bị phần cứng, trong đó có nhiệm vụ nhận và gửi dữ liệu SMS, việc sử dụng giao tiếp UART (Universal Asynchronous Receiver-Transmitter) là một giải pháp hiệu quả và phổ biến. Với cấu hình phần cứng của gateway dựa trên Raspberry Pi 5, cần xây dựng chương trình hỗ trợ giao tiếp UART để kết nối với module SIM.

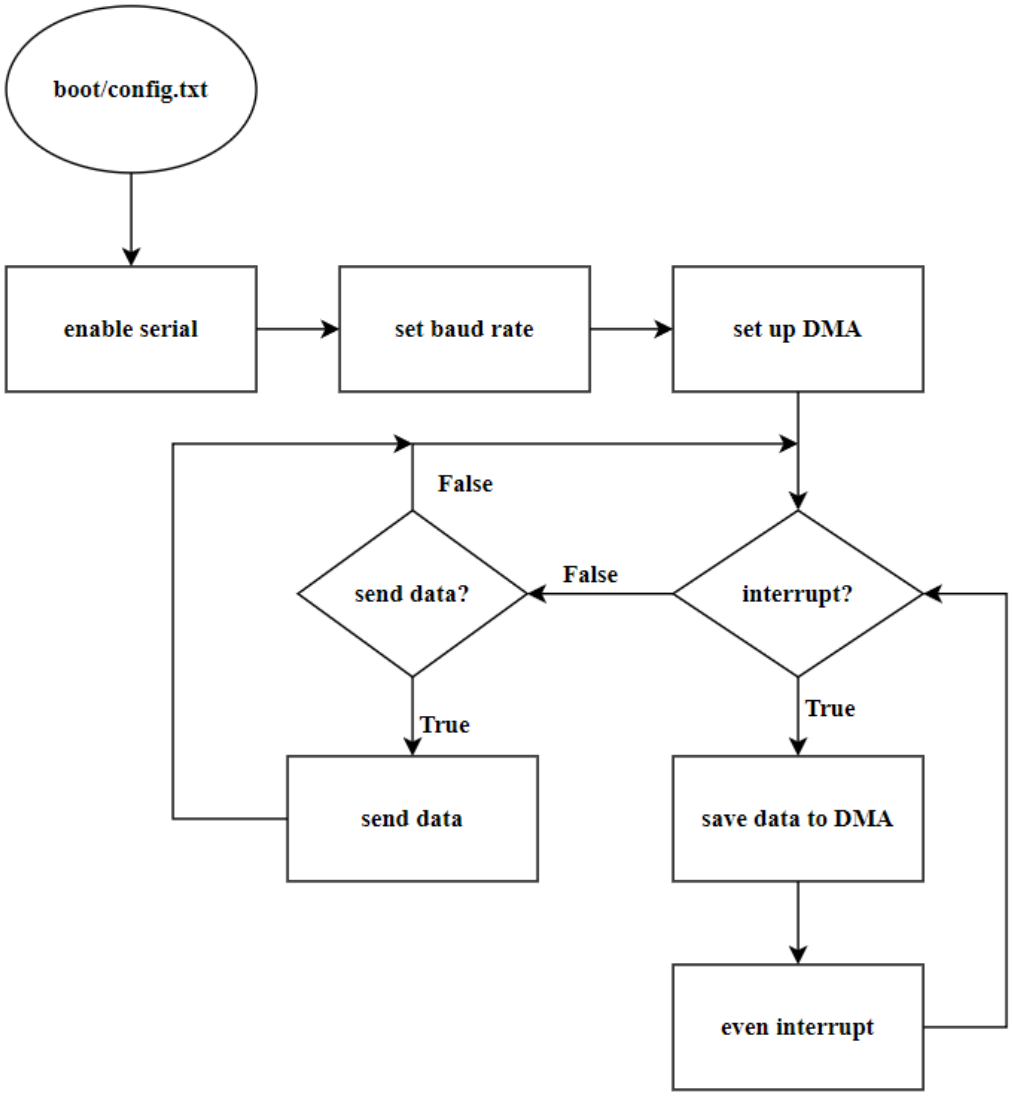
Giao thức UART (Universal Asynchronous Receiver/Transmitter) là một trong những phương thức giao tiếp phổ biến giữa CPU và các thiết bị ngoại vi. UART hoạt động dựa trên hai chân chính là TX (chân truyền) và RX (chân nhận), cho phép truyền tải dữ liệu một cách đơn giản và hiệu quả. Trong quá trình truyền thông, dữ liệu được truyền từ TX của thiết bị gửi sang RX của thiết bị nhận mà không cần thêm các tín hiệu đồng bộ phức tạp, nhờ vào cơ chế truyền thông bất đồng bộ.

Mặc dù UART không sử dụng xung nhịp chung để đồng bộ giữa các thiết bị, nó vẫn đảm bảo tính chính xác cao thông qua việc truyền từng bit dữ liệu theo một chuỗi liên tiếp. Dữ liệu được truyền với tốc độ baud đã được cài đặt trước giữa hai thiết bị, giúp chúng có thể hiểu đúng và nhận chính xác các gói tin.



Hình 3.1 Khung truyền UART

Với kiến trúc của chương trình nhúng, việc sử dụng giao thức UART đòi hỏi phải thiết kế và lập trình firmware sao cho phù hợp để có thể giao tiếp hiệu quả với phần cứng. Quá trình này bao gồm nhiều bước quan trọng, từ việc cấu hình các tham số của UART đến việc xử lý dữ liệu truyền và nhận thông qua các chân TX và RX.

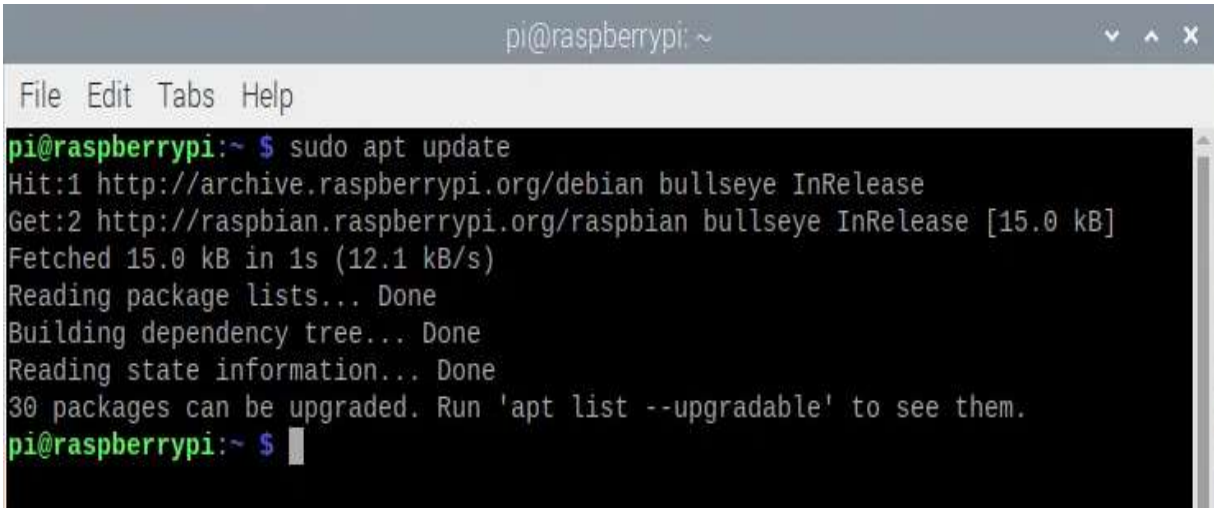


Hình 3.2 Lưu đồ thuật toán giao thức UART

Với giao tiếp UART để trao đổi dữ liệu của hệ thống và mặc định được khởi tạo trên hệ điều hành Raspberry Pi OS với baudrate 9600 bps. Tuy nhiên, để đảm bảo khả năng giao tiếp hiệu quả với module SIM, cần thiết lập lại baudrate của thiết bị lên 115200 bps. Đồng thời, chế độ DMA (Direct Memory Access) phải được kích hoạt để tăng tốc độ xử lý dữ liệu, giảm tải cho CPU và đảm bảo tính liên tục trong quá trình trao đổi dữ liệu. Ngoài ra, khi sử dụng DMA, cần đặt độ ưu tiên của tín hiệu ngắt ở mức cao nhất để đảm bảo thiết bị có thể xử lý dữ liệu ngay lập tức mà không bị gián đoạn bởi các tác vụ khác.

(b) Khởi tạo kết nối mạng

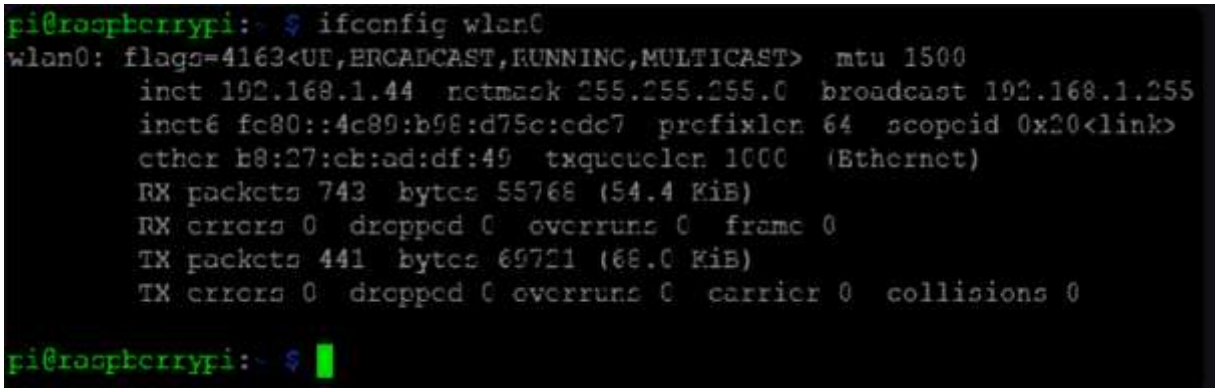
Để đảm bảo quá trình truyền tải dữ liệu từ thiết bị gateway lên server diễn ra một cách ổn định và hiệu quả, cũng như hỗ trợ giám sát hệ thống từ nền tảng cloud, việc thiết kế kết nối mạng cần được thực hiện một cách linh hoạt và liên tục. Kết nối mạng không chỉ đóng vai trò như cầu nối giữa thiết bị gateway và server mà còn quyết định đến tốc độ, độ ổn định, và khả năng truyền tải dữ liệu trong các tình huống sử dụng khác nhau.



Hình 3.3 Cập nhật cấu hình kết nối

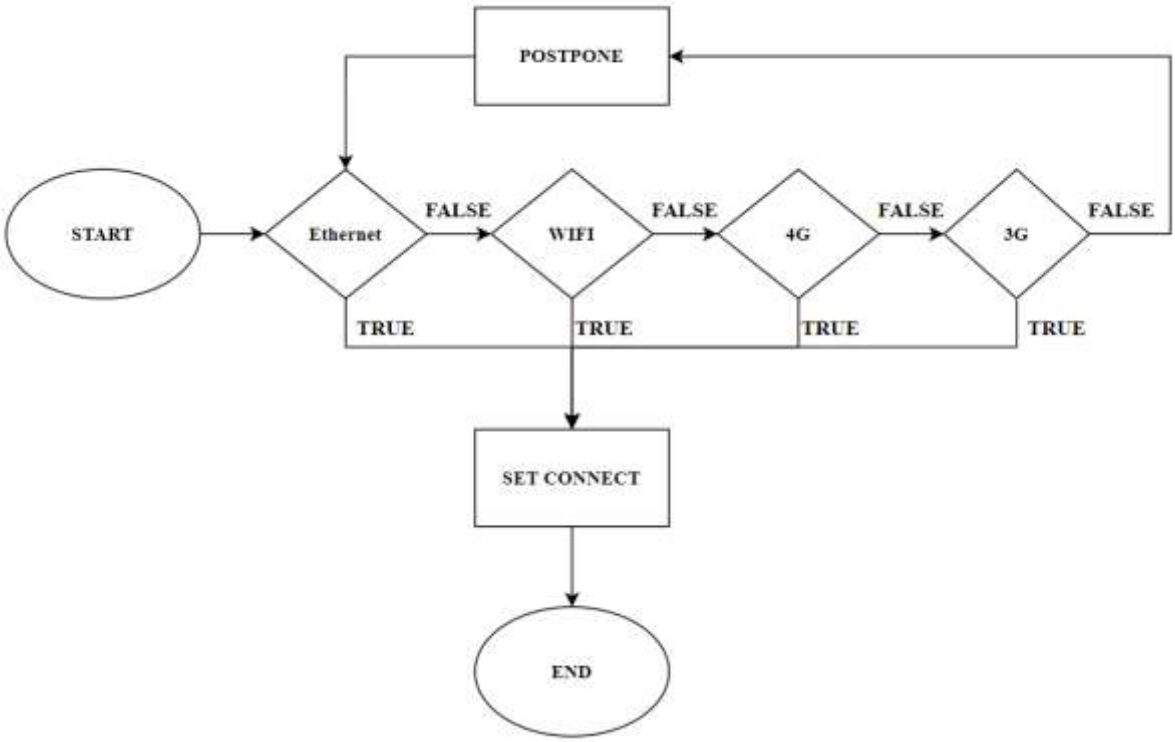
Để thiết lập kết nối mạng cho gateway, người dùng cần thực hiện các bước cấu hình chi tiết theo nhu cầu sử dụng. Với WiFi, có thể truy cập vào giao diện quản lý của gateway qua trình duyệt web hoặc ứng dụng cấu hình, sau đó nhập thông tin mạng bao gồm SSID và mật khẩu, rồi lưu lại cấu hình. Gateway sẽ tự động kết nối với mạng WiFi đã thiết lập và hiển thị trạng thái kết nối trên dashboard quản lý. Trong trường hợp sử dụng mạng LAN, người dùng chỉ cần kết nối gateway với mạng LAN qua cổng Ethernet (RJ45). Thiết bị sẽ tự động nhận địa chỉ IP thông qua DHCP hoặc có thể cấu hình địa chỉ IP tĩnh trực tiếp trong giao diện quản lý. Đặc biệt, gateway hỗ trợ tính năng chuyển đổi tự động giữa các phương thức kết nối. Nếu kết nối WiFi gặp sự cố, thiết bị sẽ tự

động chuyển sang sử dụng mạng LAN để đảm bảo quá trình truyền tải dữ liệu không bị gián đoạn.



Hình 3.4 Kiểm tra cấu hình kết nối

Việc kiểm tra kết nối được thiết kế thành một chương trình để có thể hoạt động độc lập và hoạt động mà không gián đoạn quá trình xử lý của các chương trình khác, chương trình được thiết kế để kiểm tra trạng thái kết nối theo thứ tự ưu tiên, bắt đầu từ kết nối qua cổng LAN, sau đó mới đến WiFi. Quá trình này giúp hệ thống tự động lựa chọn phương thức kết nối ổn định nhất, đảm bảo hiệu suất truyền dữ liệu tốt nhất. Việc kiểm tra được thực hiện liên tục, giúp duy trì kết nối ổn định và hạn chế tối đa tình trạng gián đoạn trong quá trình vận hành.



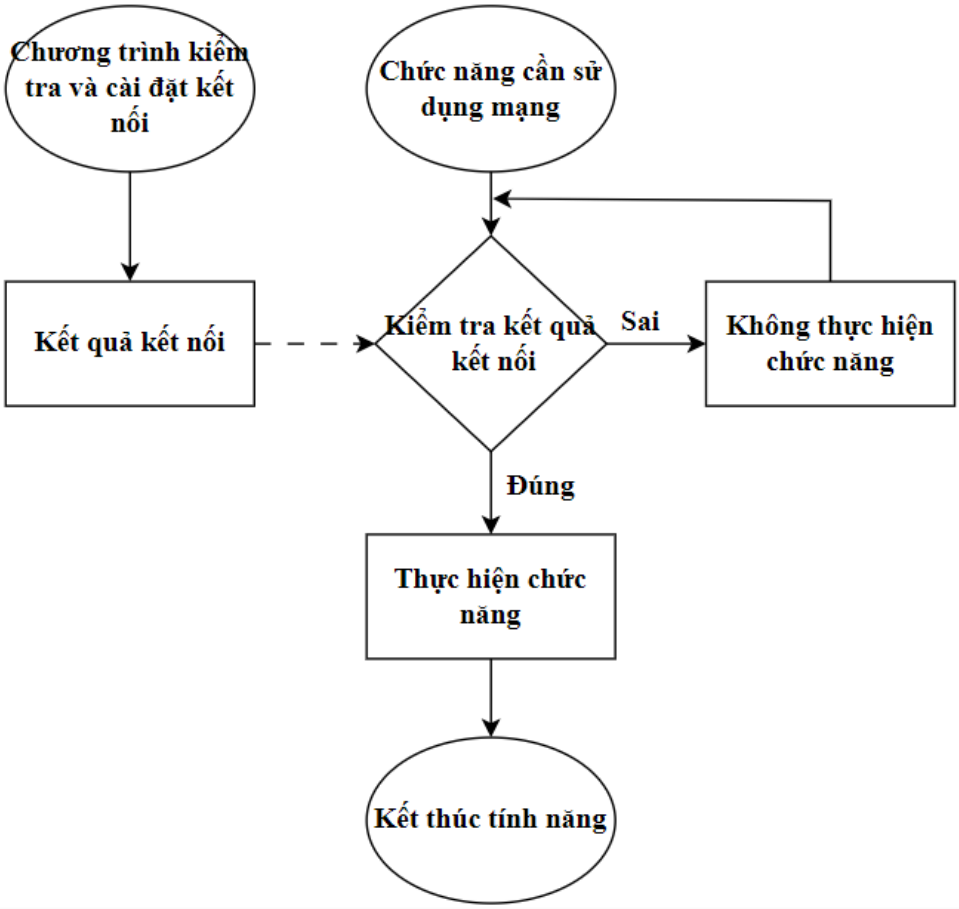
Hình 3.5 Lưu đồ thuật toán thiết lập chế độ kết nối mạng

Nếu kết nối Ethernet không khả dụng hoặc không ổn định, hệ thống sẽ tự động chuyển sang WiFi. Trong quá trình này, hệ thống sẽ ưu tiên chọn các mạng WiFi có cường độ tín hiệu mạnh nhất, điều này giúp giảm thiểu tình trạng mất gói tin (packet loss) và giảm độ trễ khi truyền dữ liệu. Việc đánh giá chất lượng kết nối WiFi được thực hiện thông qua việc đo lường các thông số như cường độ tín hiệu và độ trễ của mạng.

Trong trường hợp cả Ethernet và WiFi đều không đáp ứng được yêu cầu, hệ thống sẽ tự động chuyển sang mạng di động như 4G hoặc 3G. Tại đây, module SIM sẽ được sử dụng để duy trì kết nối. Hệ thống sẽ liên tục giám sát cường độ tín hiệu di động và nếu phát hiện tín hiệu yếu hoặc không ổn định, nó sẽ thực hiện chuyển đổi hoặc reconnect nhằm duy trì quá trình truyền tải dữ liệu một cách ổn định.

Quá trình kiểm tra và chuyển đổi kết nối mạng được hệ thống thực hiện định kỳ bằng cách gửi các gói tin kiểm tra (như ICMP ping) để đo lường độ trễ và tính ổn định của từng loại kết nối. Nếu kết nối hiện tại không đạt yêu cầu, hệ thống sẽ ngay lập tức chuyển sang một kết nối có chất lượng tốt hơn. Điều này giúp đảm bảo rằng quá trình gửi dữ liệu lên server không bị gián đoạn và đảm bảo tính liên tục của hệ thống trong mọi điều kiện mạng khác nhau.

Sau khi hoàn thành việc lựa chọn và kết nối thành công, chương trình sẽ trả về trạng thái kết nối hiện tại. Thông qua thông tin trạng thái này, các chức năng khác trong hệ thống sẽ biết được trạng thái mạng và có thể thực hiện các yêu cầu truyền tải dữ liệu lên server hoặc giám sát từ xa một cách hiệu quả. Quy trình này giúp hệ thống hoạt động ổn định, duy trì kết nối liên tục và tối ưu hóa hiệu quả sử dụng mạng, đặc biệt là trong các trường hợp mạng không ổn định hoặc khi có sự cố về kết nối mạng chính.



Hình 3.6 Lưu đồ thuật toán kiểm tra kết nối mạng khi thực hiện chức năng

Khi các chức năng yêu cầu sử dụng kết nối mạng được kích hoạt, chương trình sẽ tự động thực hiện kiểm tra trạng thái kết nối trước tiên. Kết quả của lần kiểm tra này sẽ được lưu trữ, cho phép chương trình ghi nhớ trạng thái kết nối mà không cần phải thực hiện lại quy trình kết nối từ đầu. Điều này giúp tối ưu hóa hiệu suất hoạt động của chương trình, giảm thiểu thời gian trễ và tài nguyên tính toán cần thiết.

Khi chức năng cần kết nối mạng yêu cầu được thực hiện, chương trình chỉ cần tham chiếu đến kết quả đã lưu để xác định xem thiết bị có đang kết nối ổn định hay không. Nếu kết nối vẫn hoạt động tốt, chương trình sẽ tiếp tục thực hiện các tác vụ mà không cần phải khởi động lại quá trình kiểm tra kết nối, từ đó tăng cường hiệu quả tổng thể của hệ thống. Ngược lại, nếu phát hiện kết nối không còn ổn định, chương trình có thể tự động kích hoạt lại quy trình kiểm tra kết nối để tìm kiếm phương thức kết nối thay thế.

Việc lưu trữ kết quả kiểm tra kết nối không chỉ giúp tiết kiệm thời gian mà còn giảm tải cho hệ thống, cho phép các chức năng khác tiếp tục hoạt động mà không bị gián đoạn do các vấn đề liên quan đến kết nối mạng. Kết quả này cũng sẽ góp phần vào việc

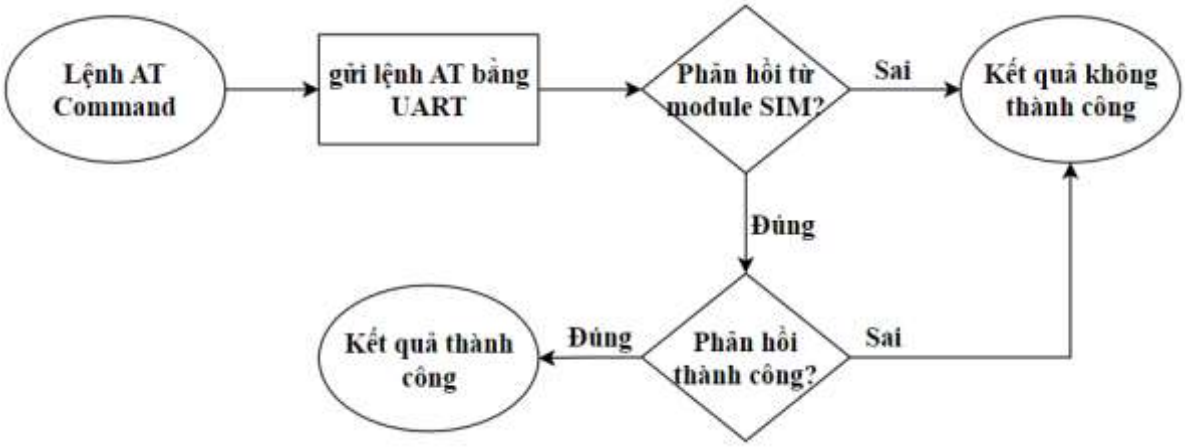
đảm bảo rằng hệ thống luôn duy trì trạng thái hoạt động tối ưu và phản ứng nhanh chóng với các thay đổi trong điều kiện mạng.

3.2.2 Chương trình giao tiếp với module sim

(a) Chương trình kết nối giao tiếp với SIM

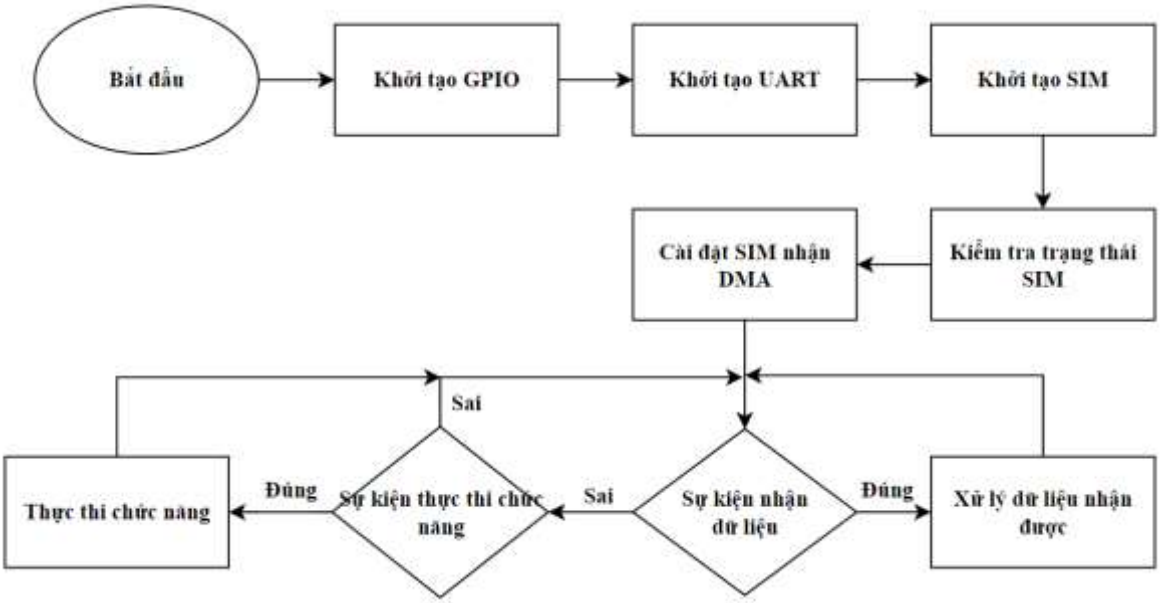
Module SIM7600 thực hiện việc giao tiếp với CPU thông qua giao thức UART, cho phép truyền tải dữ liệu nhanh chóng và hiệu quả. Để sử dụng module này, chúng ta sẽ dựa vào thư viện UART đã được xây dựng sẵn, từ đó tiến hành viết chương trình nhằm khai thác các chức năng của SIM7600. Chương trình sẽ bao gồm các bước như khởi tạo giao thức UART, gửi lệnh AT để cấu hình module, thiết lập kết nối mạng và truyền tải dữ liệu lên server. Nhờ vào khả năng giao tiếp linh hoạt và mạnh mẽ của module SIM7600, chúng ta có thể dễ dàng tích hợp vào các ứng dụng IoT, đảm bảo quá trình thu thập và gửi dữ liệu diễn ra liên tục và ổn định.

Chương trình khởi tạo sẽ tiến hành gửi các lệnh AT command thông qua phương thức UART để có thể kiểm tra phản hồi giữa module SIM với CPU.



Hình 3.7 Lưu đồ thuật toán giao tiếp với SIM

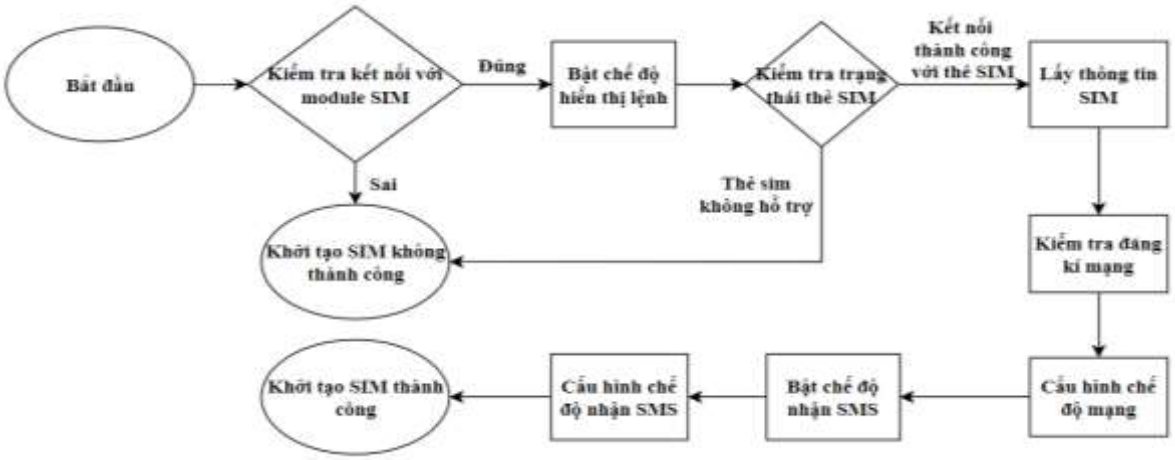
Khi gửi một lệnh AT Command để giao tiếp với module SIM, lệnh này sẽ được truyền qua UART đến module. Sau khi lệnh được gửi, chương trình sẽ bắt đầu theo dõi phản hồi từ module SIM tới CPU. Nếu trong thời gian đợi không có phản hồi từ module, chương trình sẽ trả về kết quả thực thi không thành công để thông báo lỗi kết nối. Ngược lại, nếu nhận được phản hồi, chương trình sẽ phân tích nội dung phản hồi từ module để xác định trạng thái của lệnh, xem phản hồi đó có chỉ ra thành công hay thất bại. Nếu phản hồi chỉ ra thành công, chương trình sẽ trả về kết quả thành công để xác nhận lệnh đã thực thi đúng cách, giúp đảm bảo quá trình giao tiếp và kiểm soát module diễn ra hiệu quả và liên tục.



Hình 3.8 Lưu đồ thuật toán xử lý yêu cầu của module SIM

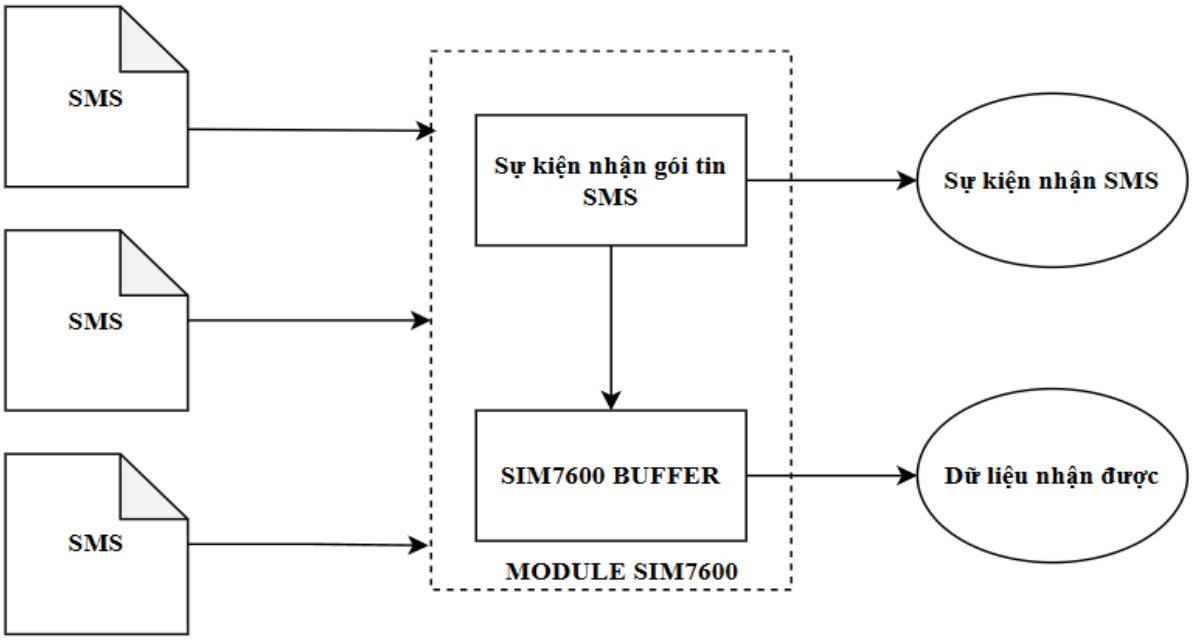
Với thiết kế này, chương trình giao tiếp giữa module SIM và CPU hoạt động như một firmware độc lập, cho phép hệ thống linh hoạt trong việc tùy chỉnh, bổ sung các tính năng mới cũng như mở rộng phát triển trong tương lai. Nhờ đó, hệ thống có thể dễ dàng thích ứng với các yêu cầu kỹ thuật khác nhau, cải thiện hiệu suất và đảm bảo tính ổn định trong quá trình vận hành.

Để sử dụng module SIM7600 hiệu quả, bước đầu tiên là khởi tạo các chế độ cần thiết để kích hoạt đầy đủ các chức năng của module. Việc này bao gồm việc thiết lập các tham số kết nối như chế độ, cấu hình kết nối để đảm bảo giao tiếp ổn định với CPU, và cài đặt các lệnh AT cần thiết để kích hoạt các chức năng quan trọng như gửi và nhận dữ liệu qua mạng di động. Quá trình khởi tạo này đảm bảo rằng module SIM hoạt động đúng và đầy đủ các chức năng.



Hình 3.9 Lưu đồ thuật toán khởi tạo và cài đặt SIM

Một yếu tố quan trọng trong quá trình khởi tạo module SIM là cấu hình để module có khả năng nhận tin nhắn SMS. Khi đã được thiết lập đúng cách, quá trình nhận SMS sẽ diễn ra tự động mà không cần can thiệp thủ công. Sau khi nhận tin nhắn, module SIM sẽ tạo ra một sự kiện để thông báo quá trình tiếp nhận dữ liệu đã hoàn tất, giúp hệ thống xử lý tin nhắn kịp thời. Ngoài ra, module SIM còn được cấu hình để có thể xử lý nhiều SMS đồng thời, tăng cường khả năng đáp ứng và duy trì hiệu suất cao trong môi trường yêu cầu tiếp nhận dữ liệu liên tục.

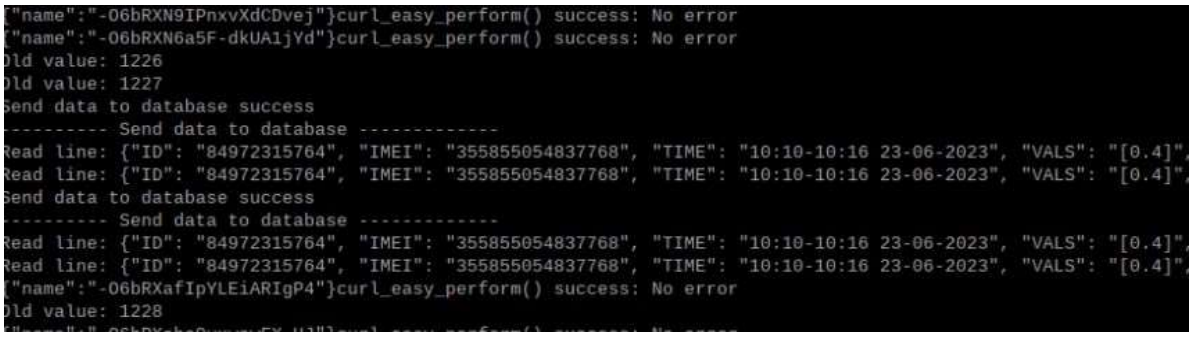


Hình 3.10 Quá trình xử lý module SIM nhận SMS

Cuối cùng, việc xây dựng thư viện giao tiếp không chỉ bao gồm việc gửi và nhận các lệnh AT, mà còn đòi hỏi phải tối ưu hóa các tác vụ quản lý tín hiệu, giảm thiểu độ trễ và đảm bảo hiệu suất truyền dữ liệu. Khi sử dụng DMA (Direct Memory Access) để xử lý các tín hiệu ngắt, hệ thống có thể nhận và xử lý dữ liệu từ module SIM7600 mà không cần phải chờ CPU xử lý từng byte dữ liệu. Điều này giúp tối ưu hóa quá trình giao tiếp và đảm bảo hệ thống hoạt động ổn định, ngay cả khi có nhiều luồng dữ liệu đến cùng lúc.

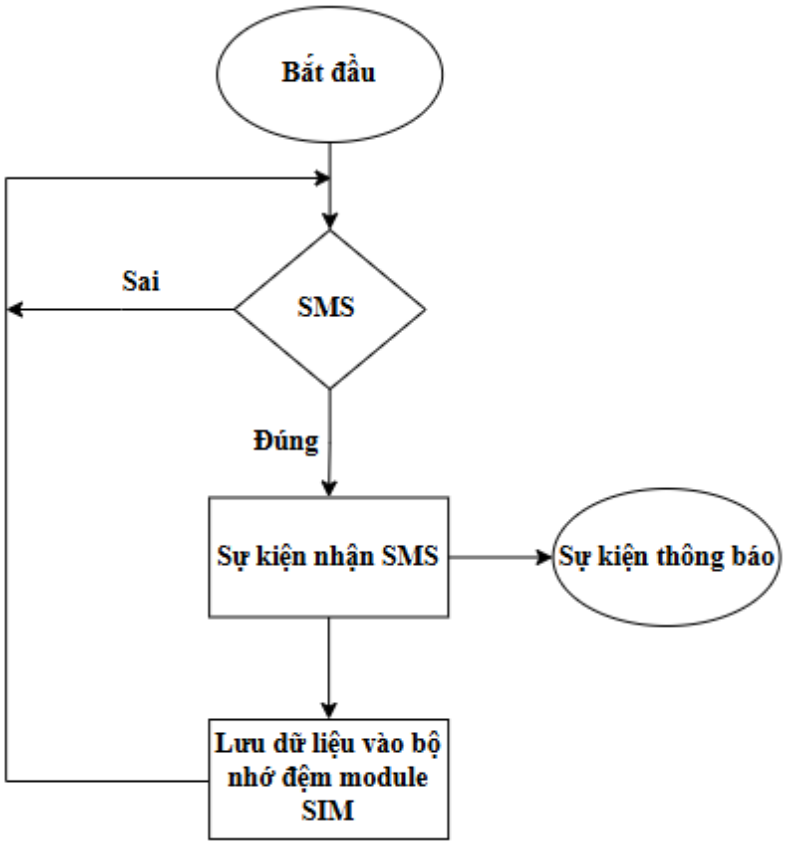
(b) Chương trình nhận dữ liệu SMS

Sau khi hoàn tất quá trình khởi tạo giao tiếp UART trên phần cứng, bước tiếp theo là phát triển phần mềm để tận dụng giao thức này cho việc trao đổi dữ liệu và xử lý tin nhắn SMS. Phần mềm này sẽ đóng vai trò trung gian, giúp hệ thống gateway giao tiếp với các thiết bị ngoại vi, đặc biệt là module SIM, để thực hiện các nhiệm vụ quan trọng như gửi, nhận, và xử lý dữ liệu tin nhắn SMS.



Hình 3.11 Quá trình nhận dữ liệu SMS

Khi chế độ nhận nhiều SMS của module SIM được kích hoạt, mọi dữ liệu gửi từ sensor đến gateway sẽ được module tự động tiếp nhận. Sau mỗi lần nhận dữ liệu, module SIM sẽ kích hoạt một sự kiện để thông báo cho CPU rằng dữ liệu đã được nhận thành công. Điều này giúp CPU có thể phản hồi kịp thời, tối ưu hóa quá trình xử lý dữ liệu từ các sensor mà không cần giám sát thủ công, đảm bảo hệ thống hoạt động một cách liên tục và hiệu quả.



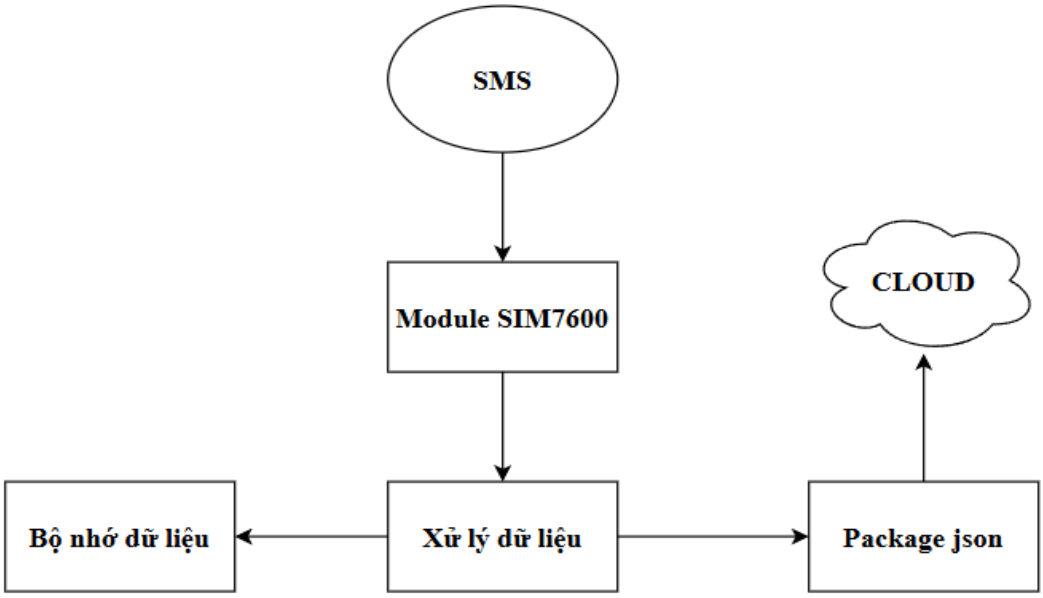
Hình 3.12 Lưu đồ module SIM xử lý sự kiện SMS

Bên cạnh đó, vì dữ liệu SMS có thể bao gồm các tin nhắn không liên quan hoặc không thuộc hệ thống từ các bộ sensor, cần có cơ chế kiểm tra và loại bỏ các gói tin không thuộc về hệ thống trước khi xử lý. Điều này đảm bảo rằng chỉ những dữ liệu hợp lệ được xử lý và ngăn ngừa việc lấy nhầm hoặc làm sai lệch thông tin.

Dữ liệu SMS được nhận từ module sau khi xử lý các lệnh này sẽ được truyền đến hệ thống chính của gateway. Tại đây, các thông tin từ tin nhắn được phân tích, lưu trữ, và sẵn sàng để sử dụng cho các mục đích xử lý tiếp theo. Việc triển khai thành công giao tiếp qua AT Command không chỉ giúp đảm bảo việc thu thập dữ liệu SMS diễn ra chính xác và nhanh chóng mà còn tạo nền tảng để phát triển các tính năng nâng cao như tự động hóa xử lý tin nhắn và tích hợp với các dịch vụ khác.

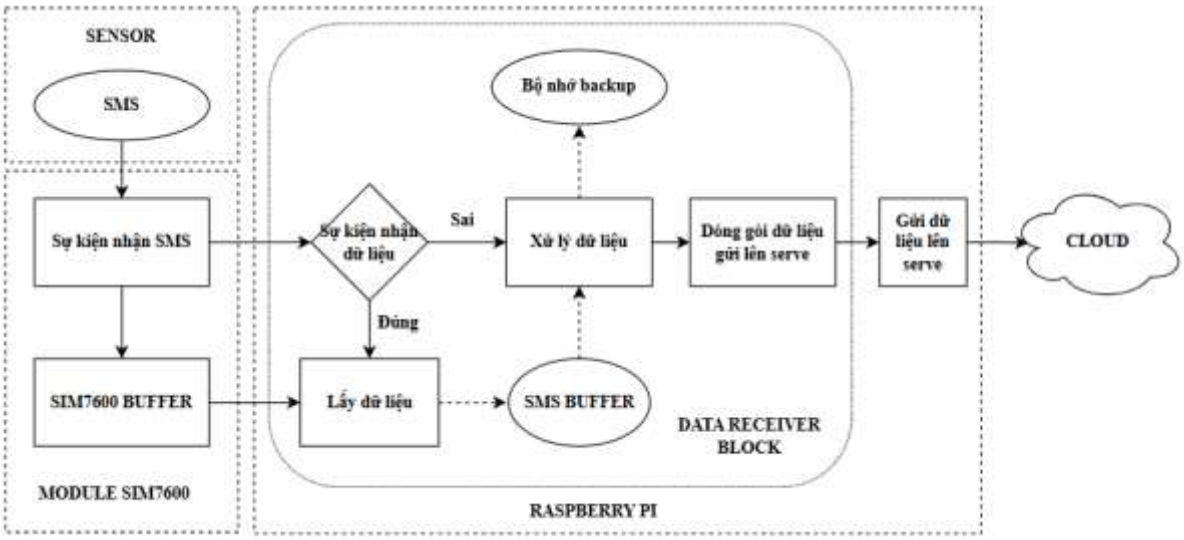
3.2.3 Chương trình xử lý

Chương trình xử lý chính sẽ nhận dữ liệu SMS từ sensor thông qua module SIM, thực hiện các bước xử lý và chuẩn hóa dữ liệu, sau đó gửi lên server. Đồng thời, chương trình cũng sẽ lưu trữ bản sao dữ liệu đã nhận từ sensor vào bộ nhớ, đóng vai trò dự phòng giống như một thiết bị datacenter thu nhỏ. Việc lưu trữ này không chỉ giúp bảo vệ dữ liệu trong trường hợp xảy ra gián đoạn kết nối với server mà còn tăng cường tính toàn vẹn và độ tin cậy của hệ thống IoT, đảm bảo không mất mát thông tin quan trọng từ quá trình giám sát và thu thập dữ liệu.



Hình 3.13 Kiến trúc chương trình chính

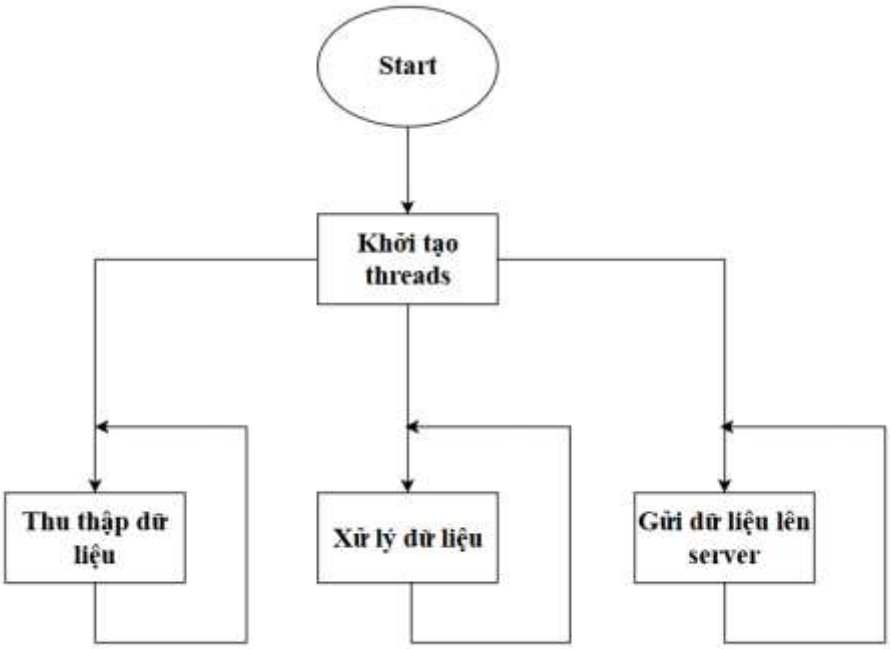
Trong hệ thống, chương trình được thiết kế để tạo sự phối hợp giữa module SIM7600 và Raspberry Pi, đảm bảo quá trình tiếp nhận và xử lý dữ liệu từ sensor diễn ra liên tục. Khi dữ liệu từ các sensor được gửi tới module SIM, module sẽ lưu trữ dữ liệu vào buffer của mình. Raspberry Pi sau đó sẽ truy cập buffer này, lấy dữ liệu và tiến hành các bước xử lý cần thiết, đảm bảo dữ liệu được chuẩn hóa và gửi lên cơ sở dữ liệu một cách chính xác và kịp thời. Cách tiếp cận này giúp tối ưu hóa việc truyền tải dữ liệu, hạn chế gián đoạn và đảm bảo tính chính xác của thông tin trong toàn bộ hệ thống IoT.



Hình 3.14 Kiến trúc xử lý của thiết bị

Với kiến trúc xử lý như trên, chương trình chính thực hiện ba nhiệm vụ chính là thu thập dữ liệu, xử lý dữ liệu, và truyền tải dữ liệu lên server. Để đảm bảo hoạt động liên tục và không bị mất dữ liệu, các nhiệm vụ này phải luôn duy trì liên tục trong quá trình vận hành. Các tác vụ cần chạy đồng thời và không bị gián đoạn, để đảm bảo dữ liệu được thu thập và cập nhật lên server kịp thời, giảm thiểu khả năng mất mát dữ liệu trong quá trình xử lý.

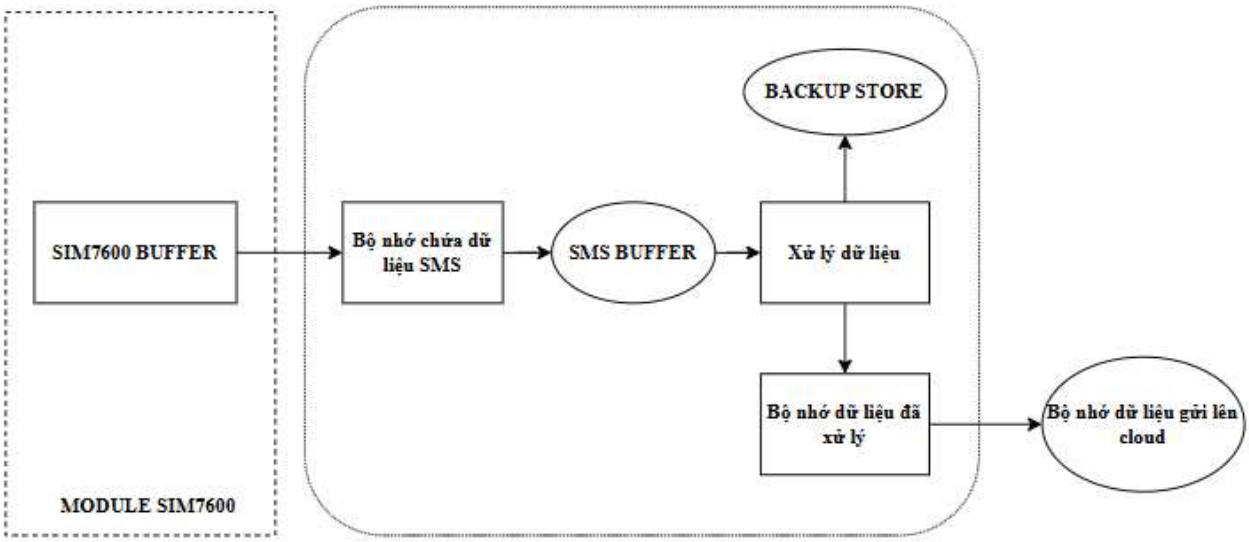
Để đáp ứng yêu cầu này, hệ thống áp dụng multithread để tối ưu hóa lịch trình của các tác vụ. Multithread cho phép lập trình viên định nghĩa và quản lý các task một cách hiệu quả, đảm bảo chúng có thể thực thi song song và không gây trễ cho hệ thống trong suốt quá trình xử lý dữ liệu.



Hình 3.15 Lưu đồ thuật toán chương trình xử lý chính

Mỗi task được quản lý theo thứ tự ưu tiên và thực hiện trong các khoảng thời gian cụ thể mà không xung đột, giúp hệ thống đạt hiệu suất cao và duy trì tính ổn định, nhất quán. Việc áp dụng multithread không chỉ giúp tối ưu hóa tốc độ xử lý mà còn giúp chương trình đạt được hiệu quả cao, đảm bảo tính liên tục và an toàn trong thu thập và truyền tải dữ liệu.

Trong hệ thống sử dụng multithread, việc phân bổ bộ nhớ là rất quan trọng để các task có thể truy cập dữ liệu một cách an toàn và hiệu quả. multithread cung cấp các cơ chế quản lý bộ nhớ như hàng đợi (queues), bộ đệm vòng (circular buffers) cho phép các task trao đổi dữ liệu mà không gây xung đột bộ nhớ. Khi một task hoàn thành xử lý dữ liệu và cần chia sẻ kết quả với một task khác, hệ thống sẽ lưu dữ liệu vào bộ nhớ được phân bổ, giúp đảm bảo tính toàn vẹn của dữ liệu khi nhiều task truy cập.

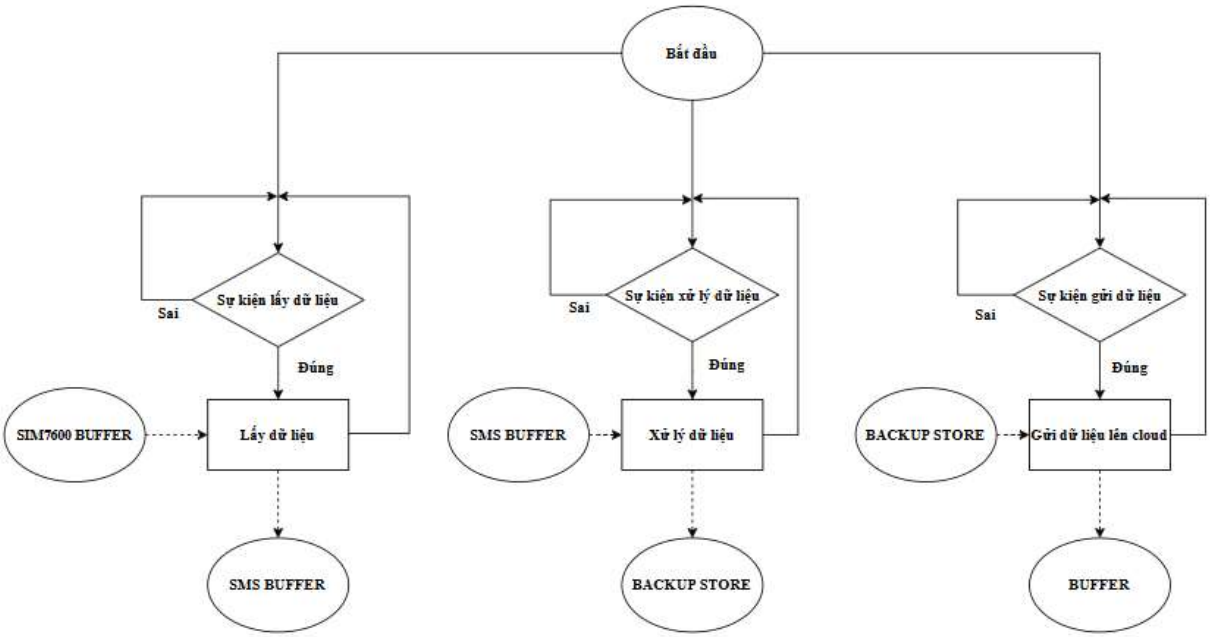


Hình 3.16 Kiến trúc phân bố dữ liệu của chương trình xử lý

Đặc biệt, hàng đợi (queues) là phương tiện phổ biến để truyền tải dữ liệu giữa các task. Khi một task muốn gửi dữ liệu đến task khác, nó chỉ cần ghi dữ liệu vào hàng đợi chung, nơi task nhận có thể lấy dữ liệu khi cần thiết. Điều này giúp duy trì hoạt động đồng bộ và đảm bảo các task có thể truy cập dữ liệu của nhau mà không phải cạnh tranh hay chờ đợi lâu.

Với việc phân bổ bộ nhớ đúng cách và sử dụng các cơ chế đồng bộ của multithread, hệ thống có thể xử lý dữ liệu liên tục mà không làm giảm hiệu suất hoặc gặp phải tình trạng thiếu bộ nhớ.

Ngoài ra, hệ thống sẽ quản lý việc thực thi các task dựa trên cơ chế quản lý sự kiện, đảm bảo các task chỉ kích hoạt khi có thông báo về chức năng hoặc dữ liệu cần xử lý. Cơ chế này giúp tối ưu hóa tài nguyên hệ thống bằng cách giới hạn số lượng task chạy cùng lúc và tránh việc tiêu tốn tài nguyên không cần thiết khi không có dữ liệu hoặc yêu cầu xử lý mới.



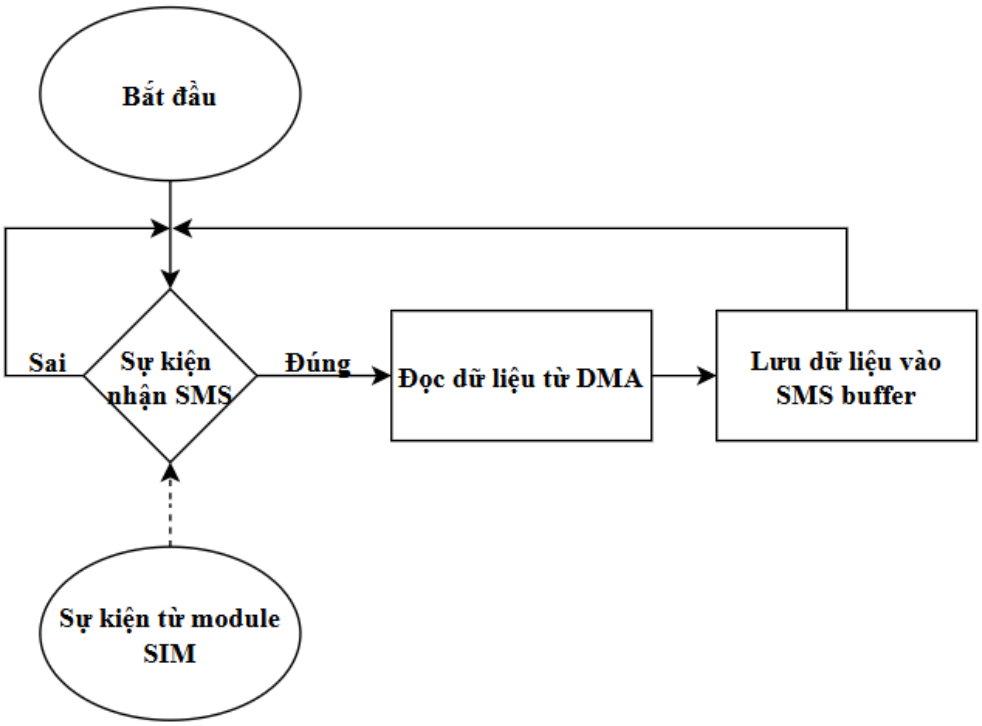
Hình 3.17 Lưu đồ thuật toán xử lý sự kiện

Trong hệ thống này, mỗi bước của quy trình thu thập và xử lý dữ liệu từ sensor được kiểm soát một cách tự động thông qua các sự kiện do module SIM và multithread quản lý. Khi module SIM nhận được dữ liệu SMS từ sensor, nó sẽ tạo ra một sự kiện thông báo dữ liệu mới đã được nhận. Sự kiện này kích hoạt task xử lý dữ liệu, giúp hệ thống lấy dữ liệu từ bộ đệm của module SIM và chuyển sang task tiếp theo.

Sau khi task xử lý dữ liệu hoàn tất, hệ thống sẽ tự động phát ra sự kiện tiếp theo để kích hoạt task gửi dữ liệu, đảm bảo rằng dữ liệu đã qua xử lý được truyền lên server. Các task này hoạt động đồng bộ thông qua cơ chế báo hiệu của các cờ được cài đặt trong multi thread, đảm bảo rằng toàn bộ quy trình từ nhận, xử lý đến gửi dữ liệu diễn ra liên tục mà không cần sự can thiệp thủ công. Điều này không chỉ giúp tối ưu hiệu suất mà còn đảm bảo tính ổn định của hệ thống trong việc cập nhật dữ liệu thời gian thực, ngay cả trong các tình huống đòi hỏi xử lý liên tục và không có độ trễ.

(a) Chương trình thu thập dữ liệu

Task thu thập dữ liệu đóng vai trò trọng yếu trong hệ thống, vì nó đảm nhận nhiệm vụ quản lý và nhận dữ liệu từ sensor một cách liên tục và chính xác, nhằm tránh hiện tượng mất mát dữ liệu. Task này luôn cần duy trì trạng thái sẵn sàng nhận dữ liệu, đảm bảo rằng mọi thông tin từ sensor đều được thu thập đầy đủ và kịp thời.



Hình 3.18 Lưu đồ thuật toán thu thập dữ liệu

Khi module SIM nhận được dữ liệu SMS từ sensor, dữ liệu này sẽ lập tức được ghi trực tiếp vào bộ nhớ DMA để nhanh chóng lưu trữ mà không cần phải chờ quá trình xử lý hoàn tất. Sau khi hoàn tất ghi vào DMA, chương trình sẽ sao chép dữ liệu này vào bộ nhớ đệm SMS buffer, nơi dữ liệu sẽ được lưu trữ tạm thời để xử lý tiếp theo.

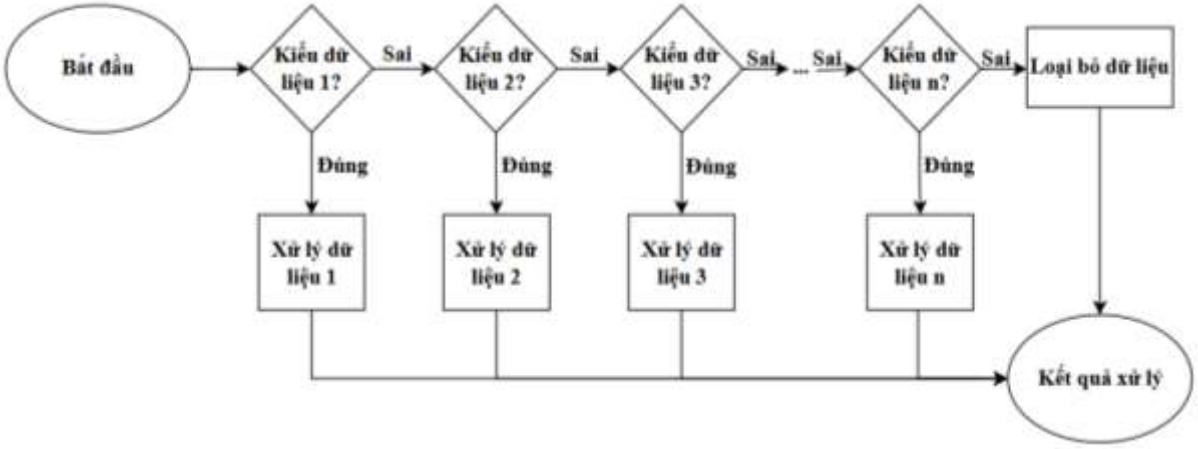
Trong quá trình sao chép từ DMA sang SMS buffer, nếu có thêm dữ liệu SMS mới đến, hệ thống sẽ ghi trực tiếp dữ liệu mới này vào DMA mà không làm gián đoạn quá trình sao lưu vào SMS buffer. Nhờ cơ chế này, hệ thống đảm bảo không bị mất dữ liệu ngay cả khi có các luồng dữ liệu đến liên tục, đồng thời giữ cho việc lưu trữ và xử lý dữ liệu diễn ra mượt mà, hiệu quả.

(b) Chương trình xử lý dữ liệu

Để đáp ứng khả năng tương thích và xử lý dữ liệu từ nhiều phiên bản sensor, hệ thống cần có cơ chế kiểm tra cấu trúc của từng gói tin nhận được. Bằng cách phân tích cấu trúc này, hệ thống có thể xác định phiên bản và định dạng dữ liệu của từng gói tin, từ đó chọn phương pháp xử lý phù hợp.

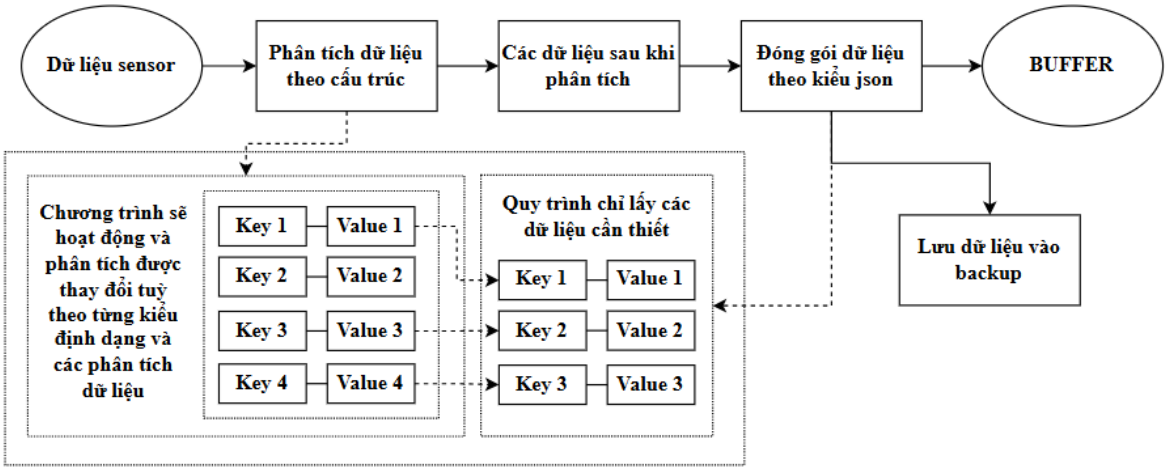
Điều này không chỉ giúp đảm bảo tính chính xác trong việc xử lý dữ liệu từ các nguồn khác nhau mà còn tăng cường khả năng mở rộng của hệ thống khi có thêm các phiên bản sensor mới. Cấu trúc linh hoạt này cho phép chương trình nhận diện và xử lý

dữ liệu một cách nhanh chóng và hiệu quả, ngay cả khi có sự khác biệt về định dạng giữa các phiên bản.



Hình 3.19 Lưu đồ thuật toán kiểm tra định dạng dữ liệu

Sau khi phân tích, dữ liệu sẽ được chọn lọc và đóng gói lại dưới định dạng JSON. Việc chuyển đổi dữ liệu sang JSON giúp tăng tính dễ đọc, dễ sử dụng, và cho phép hệ thống truyền tải thông tin một cách nhất quán và tiêu chuẩn hóa. Định dạng JSON cũng tạo điều kiện thuận lợi cho việc tích hợp với các dịch vụ khác hoặc lưu trữ trong cơ sở dữ liệu, giúp quá trình xử lý và truy xuất dữ liệu nhanh chóng, đồng thời hỗ trợ khả năng mở rộng của hệ thống khi cần trao đổi với các nền tảng khác.



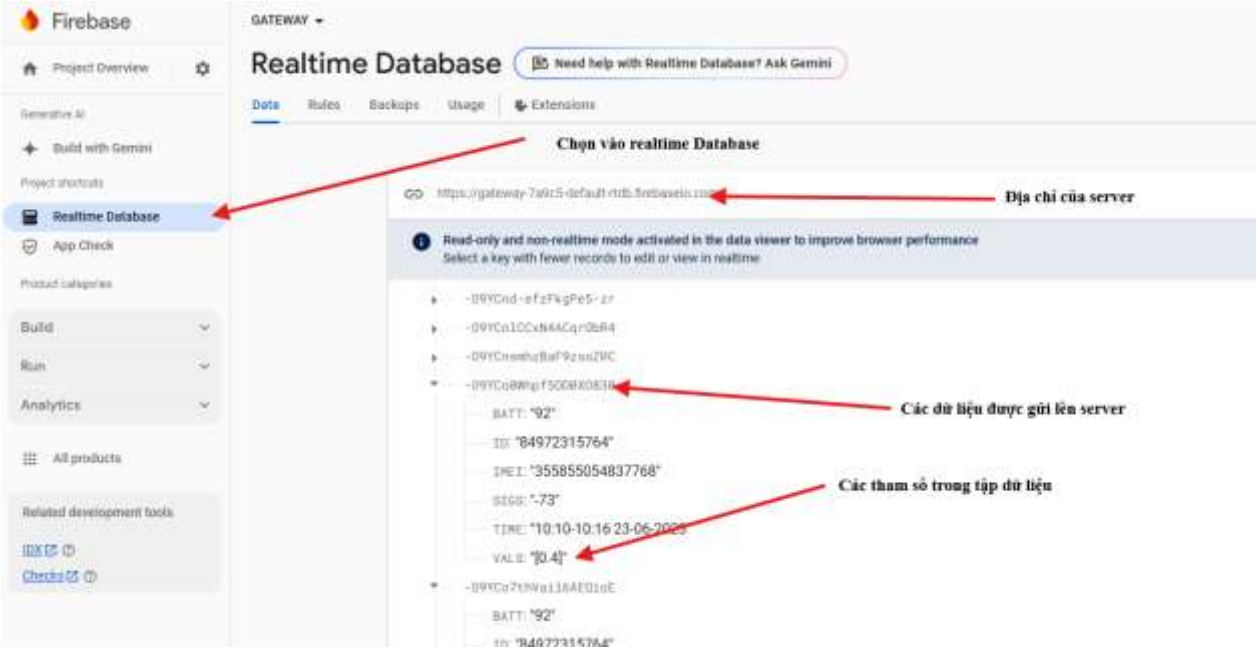
Hình 3.20 Lưu đồ thuật toán xử lý dữ liệu

Các gói tin và dữ liệu gửi đến gateway thường chứa nhiều thông số kèm theo. Trong quá trình phân tích, chương trình sẽ thực hiện các thao tác cần thiết để thu thập và xử lý toàn bộ thông tin. Sau đó, chỉ những thông tin thiết yếu mới được chọn lọc và đóng gói để gửi lên server. Định dạng JSON sẽ được sử dụng để tổng hợp và sắp xếp dữ liệu, tạo ra một cấu trúc nhất quán và có tính tương thích cao với các ứng dụng khác.

Điều này giúp tối ưu băng thông, đảm bảo chỉ các dữ liệu cần thiết được truyền tải và dễ dàng cho việc quản lý, lưu trữ trên server.

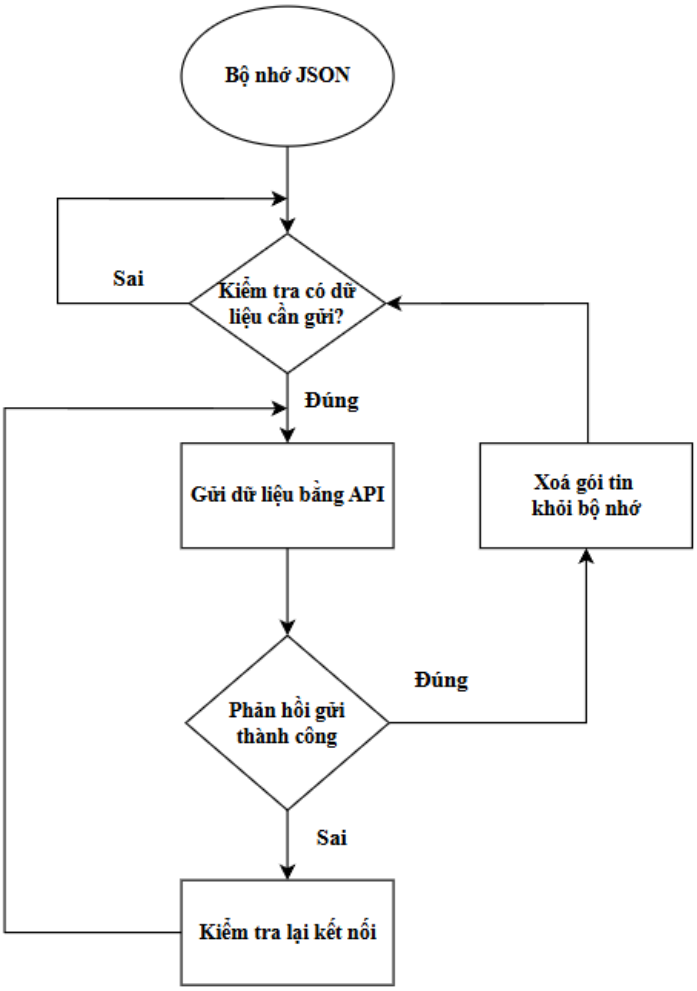
(c) Chương trình gửi dữ liệu lên server

Hiện tại, trong giai đoạn thử nghiệm, nhóm đang sử dụng Firebase làm server lưu trữ dữ liệu. Firebase cung cấp dịch vụ Realtime Database, giúp việc lưu trữ và quản lý dữ liệu trở nên hiệu quả và thuận tiện. Realtime Database hỗ trợ giao thức HTTPS, cho phép hệ thống đẩy dữ liệu lên server một cách bảo mật và nhanh chóng. Điều này giúp nhóm dễ dàng kiểm soát và quản lý các thông tin từ sensor theo thời gian thực, hỗ trợ việc phát triển và thử nghiệm hệ thống một cách mượt mà và liên tục.



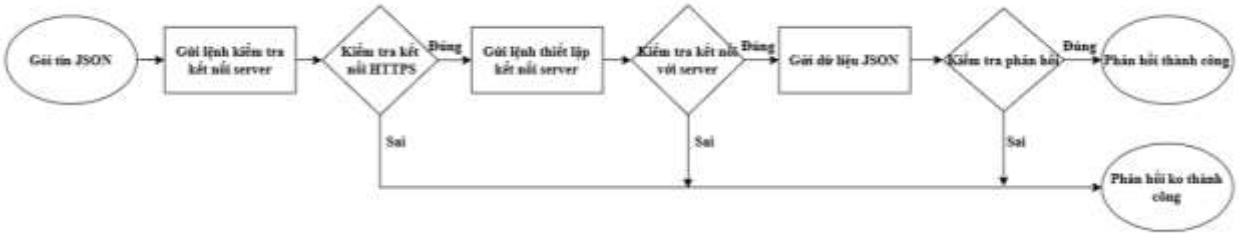
Hình 3.21 Thông tin cơ bản của Cloud Firebase

Firebase cung cấp địa chỉ server HTTPS, giúp đơn giản hóa quá trình post dữ liệu bằng cách gửi dữ liệu dưới dạng JSON qua giao thức HTTPS. Với phương thức này, dữ liệu từ các thiết bị sẽ được đưa lên server một cách nhanh chóng và bảo mật, đảm bảo tính an toàn và hiệu quả trong việc truyền tải dữ liệu theo thời gian thực. Điều này tạo điều kiện thuận lợi cho hệ thống IoT trong việc giám sát và quản lý dữ liệu, đặc biệt là khi kết hợp với Realtime Database để tự động cập nhật và đồng bộ dữ liệu.



Hình 3.22 Lưu đồ thuật toán gửi dữ liệu

Quá trình gửi dữ liệu được thực hiện theo quy trình chặt chẽ để đảm bảo tính liên tục và độ tin cậy của truyền tải. Đầu tiên, hệ thống sẽ kiểm tra bộ nhớ JSON để xác nhận có dữ liệu cần gửi hay không. Nếu dữ liệu tồn tại, các gói tin sẽ lần lượt được lấy ra và gửi lên server thông qua API. Sau mỗi lần gửi, hệ thống kiểm tra phản hồi từ server để xác định xem dữ liệu đã được nhận thành công chưa. Nếu việc gửi thất bại, chương trình sẽ kiểm tra lại kết nối mạng và thực hiện gửi lại gói tin. Khi gửi thành công, gói tin sẽ được xóa khỏi bộ nhớ JSON và hệ thống sẽ tiếp tục xử lý các dữ liệu tiếp theo, đảm bảo không có gói tin nào bị bỏ sót trong quá trình hoạt động.



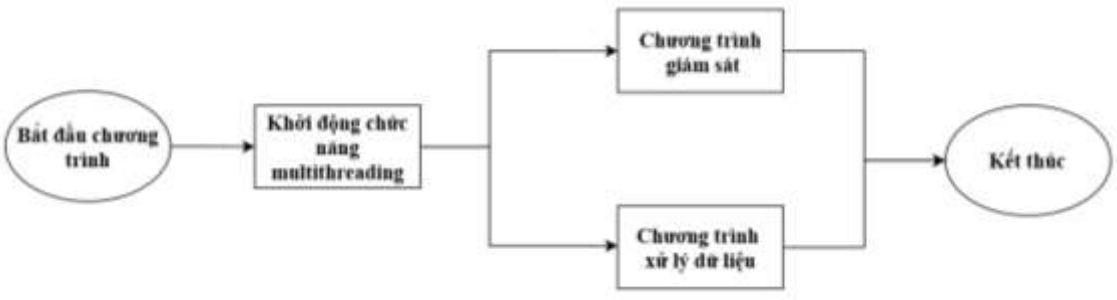
Hình 3.23 Lưu đồ thuật toán gửi dữ liệu bằng API Cloud Firebase

Quá trình sử dụng API trong hệ thống sẽ bắt đầu với việc kiểm tra kết nối giữa thiết bị và server để đảm bảo rằng địa chỉ server đã được cấu hình chính xác và sẵn sàng tiếp nhận dữ liệu. Sau khi xác nhận kết nối thành công, chương trình sẽ tiến hành gửi các chuỗi dữ liệu JSON lên server thông qua API. Trong quá trình này, mỗi phản hồi từ server sẽ được giám sát cẩn thận để đảm bảo rằng dữ liệu đã được nhận đầy đủ và đúng cách. Nếu có bất kỳ lỗi nào xảy ra, hệ thống sẽ lập tức kiểm tra và khắc phục để duy trì tính liên tục và độ tin cậy của quá trình truyền tải.

3.3 Chương trình giám sát

Thiết kế chương trình giám sát cần đảm bảo rằng quá trình giám sát hệ thống diễn ra liên tục, không làm ảnh hưởng đến các chức năng khác và không gây ra bất kỳ độ trễ hay xung đột nào trong hệ thống. Việc sử dụng nhiều luồng (multithreading) là một giải pháp tối ưu, cho phép các tác vụ giám sát và các tác vụ khác chạy song song mà không can thiệp vào nhau. Với việc triển khai theo mô hình này, mỗi luồng có thể đảm nhiệm một vai trò cụ thể, ví dụ như một luồng để giám sát trạng thái mạng, một luồng khác để xử lý phản hồi từ server, và các luồng bổ sung cho việc thu thập và truyền tải dữ liệu. Cấu trúc đa luồng này không chỉ giúp tối ưu hóa tài nguyên mà còn đảm bảo độ ổn định và hiệu quả của chương trình trong các điều kiện hoạt động thực tế.

Phân tách chương trình giám sát thành một luồng riêng và sử dụng một lõi CPU riêng biệt giúp tránh được các xung đột với các tiến trình xử lý khác, đồng thời đảm bảo rằng chương trình giám sát có độ ưu tiên cao và giá trị giám sát được cập nhật chính xác nhất. Với việc phân bổ này, luồng giám sát có thể hoạt động liên tục mà không bị ảnh hưởng bởi các tác vụ khác, nhất là trong các hệ thống yêu cầu độ chính xác và thời gian thực. Cách thiết kế này giúp tối ưu hóa việc sử dụng tài nguyên hệ thống, nâng cao hiệu quả giám sát, và đảm bảo hệ thống hoạt động ổn định ngay cả trong điều kiện tải cao.



Hình 3.24 Lưu đồ thuật toán khởi động chương trình ứng dụng

Khi chương trình khởi động chế độ đa luồng (multithreading), hai luồng riêng biệt sẽ được thiết lập và hoạt động song song. Luồng thứ nhất đảm nhiệm nhiệm vụ xử lý dữ liệu từ các thiết bị sensor, bao gồm việc thu thập, phân tích, và chuẩn bị dữ liệu trước khi gửi lên server. Trong khi đó, luồng thứ hai thực hiện chức năng giám sát toàn

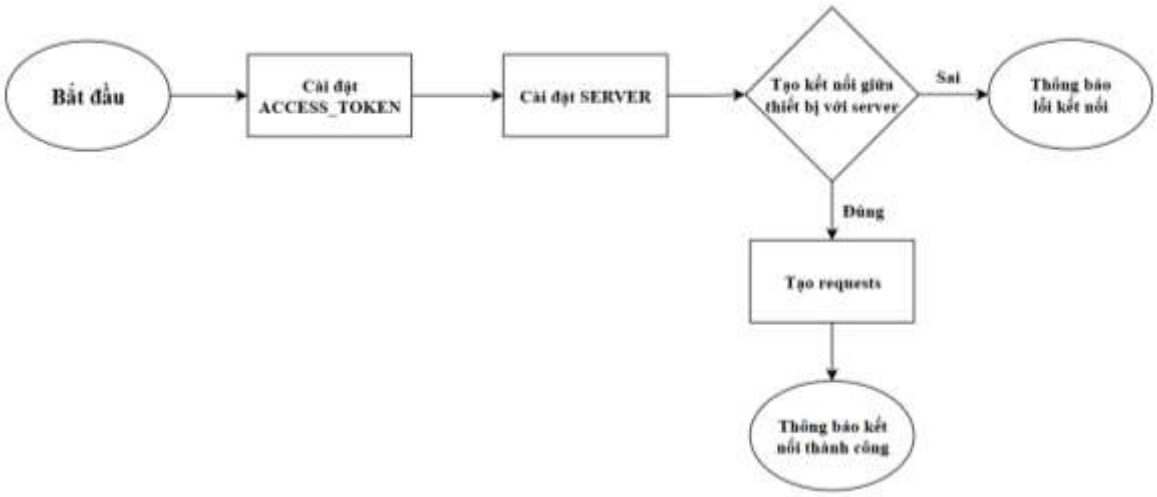
bộ hệ thống, đảm bảo các kết nối ổn định, giám sát trạng thái của các thiết bị, và phản hồi nhanh chóng nếu có sự cố xảy ra.

(a) Lập trình kết nối dashboard

Hiện tại, ThingsBoard đang được sử dụng như một nền tảng để tạo và quản lý dashboard giám sát cho hệ thống. Đây là một lựa chọn hiệu quả do ThingsBoard hỗ trợ việc tạo dashboard trực quan một cách thuận tiện, giúp giám sát các chỉ số quan trọng và tình trạng hoạt động của các thiết bị trong thời gian thực.

Hình 3.25 Kiến trúc truyền thông giám sát

ThingsBoard cũng cung cấp giao thức MQTT, cho phép hệ thống đẩy dữ liệu lên nền tảng một cách nhanh chóng và liên tục. Điều này giúp cập nhật dữ liệu từ các thiết bị IoT lên dashboard tức thời, giảm thiểu độ trễ và đảm bảo độ chính xác của thông tin giám sát.

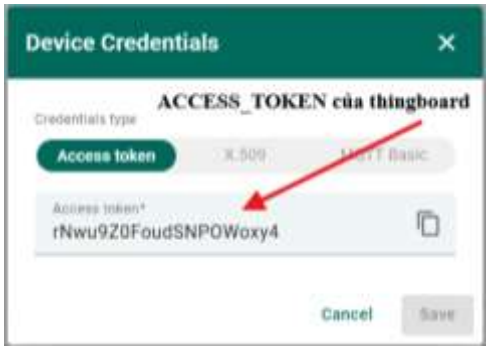


Hình 3.26 Lưu đồ thuật toán kết nối với dashboard

Để thiết lập kết nối với dashboard của ThingsBoard, cần cấu hình hai thông số quan trọng là ACCESS_TOKEN và địa chỉ server. ACCESS_TOKEN đóng vai trò xác thực thiết bị với nền tảng ThingsBoard, giúp hệ thống nhận diện và chấp nhận dữ liệu từ thiết bị. Đồng thời, địa chỉ server của ThingsBoard giúp định tuyến dữ liệu, đảm bảo thiết bị truyền đến đúng hệ thống giám sát đã cài đặt.

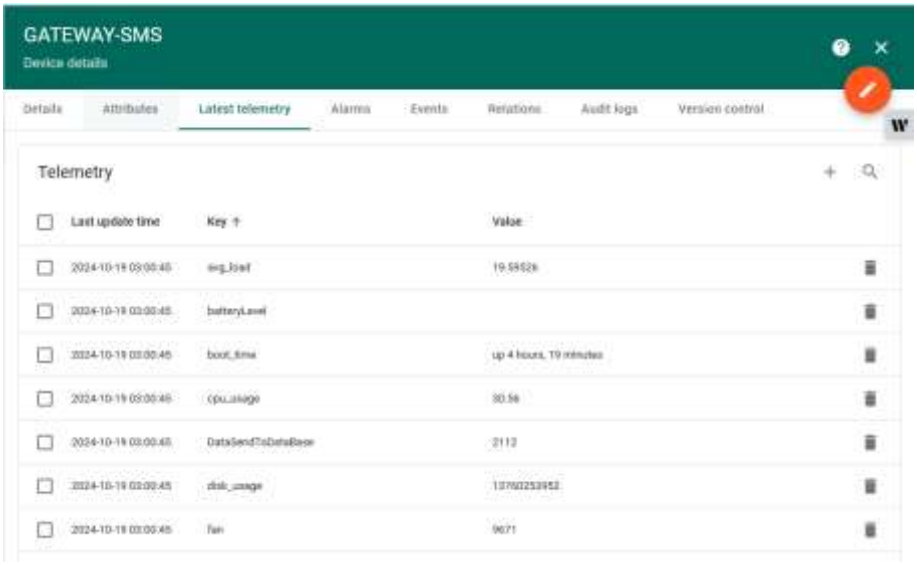
Sau khi hoàn tất cấu hình các thông số trên, thiết bị tiến hành kiểm tra kết nối ban đầu với ThingsBoard để xác nhận rằng kết nối đã thành công. Nếu thành công, thiết bị sẽ nhận được phản hồi từ server, cho phép tiếp tục các tác vụ giám sát. Kết nối được

thiết lập giúp thiết bị truyền dữ liệu giám sát lên ThingsBoard thông qua giao thức MQTT theo yêu cầu.



Hình 3.27 Access token của Thingsboard

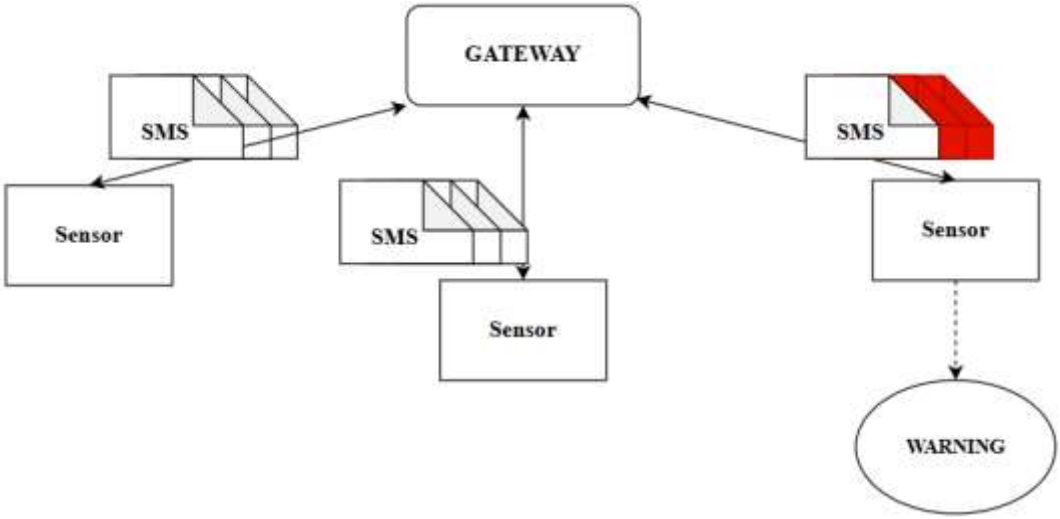
Sau khi cấu hình chính xác các thông số của dashboard trên ThingsBoard, như ACCESS_TOKEN và địa chỉ server, dữ liệu từ thiết bị sẽ được quản lý và hiển thị theo thời gian thực trên dashboard. ThingsBoard sẽ liên tục thu thập, cập nhật, và sắp xếp dữ liệu nhận được từ thiết bị, cho phép người dùng theo dõi và phân tích các thông số hệ thống một cách trực quan và ngay lập tức.



Hình 3.28 các dữ liệu được gửi lên server

(b) Thiết kế giao diện dashboard

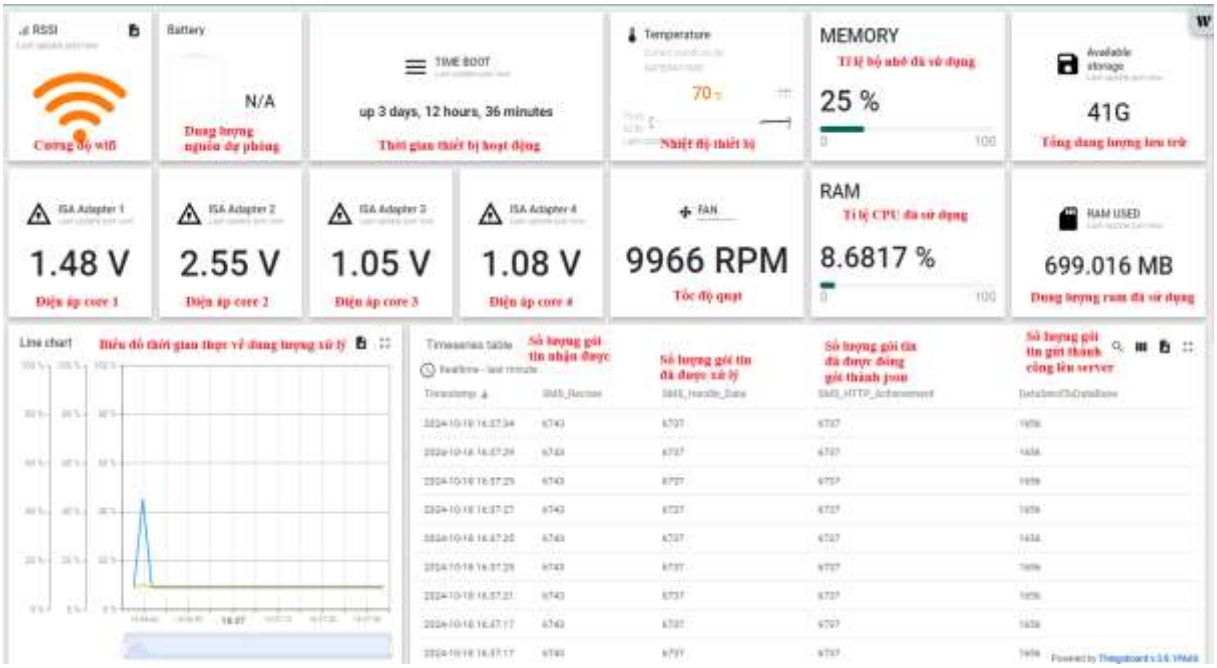
Quá trình thiết kế dashboard cần tập trung vào việc hiển thị toàn diện các thông tin về trạng thái hoạt động của thiết bị và tình hình giám sát các gói tin. Điều này nhằm giúp người dùng có thể theo dõi, đánh giá hiệu quả truyền tải dữ liệu của hệ thống và phát hiện sớm các vấn đề.



Hình 3.29 Kiến trúc hoạt động của chương trình giám sát

Nhờ vào hệ thống giám sát và phân tích dữ liệu từ các bộ sensor, khi có lỗi xảy ra trong quá trình hoạt động, các sai lệch trong dữ liệu SMS sẽ được phát hiện kịp thời. Chương trình xử lý được thiết kế để nhận diện các gói tin không đúng định dạng hoặc chứa thông tin bất thường, từ đó tạo cảnh báo hoặc ghi nhận lỗi.

Ngoài ra, tính năng giám sát số lượng gói tin theo thời gian thực sẽ giúp người giám sát dễ dàng nhận thấy khi có sự gia tăng đột ngột trong lượng dữ liệu gửi đến gateway. Điều này đặc biệt hữu ích trong việc phát hiện các vấn đề bất thường, chẳng hạn như lỗi liên tục trong bộ sensor hoặc sự cố trong hệ thống truyền tải. Qua đó, việc giám sát không chỉ giúp đánh giá hiệu quả hoạt động mà còn hỗ trợ nhận diện sớm các sự cố, từ đó đảm bảo tính ổn định và liên tục của hệ thống.



Hình 3.30 Thiết kế giao diện dashboard

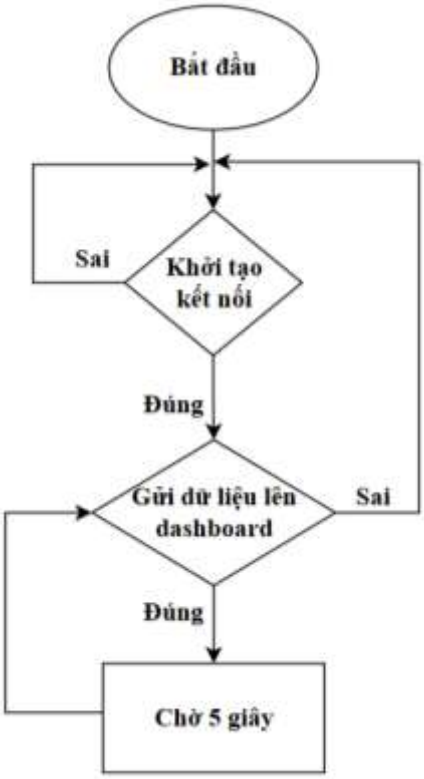
Với thiết kế hệ thống giám sát này, người giám sát có thể dễ dàng theo dõi tình trạng hoạt động tổng thể, từ cường độ kết nối mạng, thông tin về lưu trữ, đến tình trạng xử lý của vi xử lý chính. Đặc biệt, các thông tin như số lượng gói tin từ sensor, số lượng gói tin đã được xử lý và gửi lên server được thống kê chi tiết giúp cung cấp cái nhìn tổng quan về mức độ hoạt động của hệ thống.

Nhờ vào các dữ liệu này, hệ thống có thể phát hiện kịp thời các bất thường, chẳng hạn như sự gia tăng đột ngột của dữ liệu hoặc lỗi trong việc nhận gói tin, qua đó giúp đội ngũ vận hành đưa ra giải pháp sớm. Bên cạnh đó, đây cũng là nguồn dữ liệu quan trọng để đánh giá hiệu suất hoạt động, hỗ trợ việc tối ưu hóa và tăng cường độ ổn định của hệ thống khi vận hành trong thực tế.

Hình 3.31 Lưu đồ thuật toán cập nhật dữ liệu dashboard giám sát

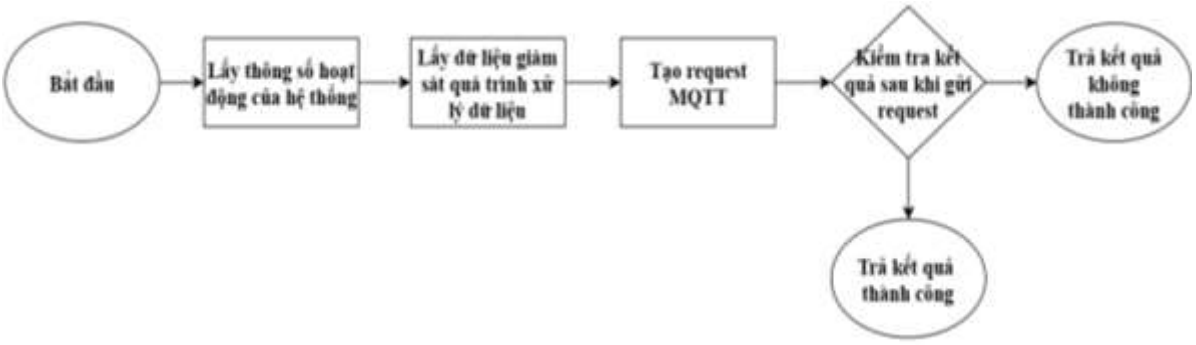
ThingsBoard liên tục kiểm tra và cập nhật dữ liệu từ các thiết bị, giúp đảm bảo rằng các thông số giám sát luôn được hiển thị theo thời gian thực. Điều này cho phép người quản lý hệ thống theo dõi chính xác tình trạng hoạt động tại từng thời điểm, từ đó phát hiện và xử lý kịp thời các vấn đề nếu có, đảm bảo quá trình truyền tải và xử lý dữ liệu luôn được duy trì ổn định.

(c) Lập trình gửi dữ liệu lên dashboard



Hình 3.32 Lưu đồ thuật toán gửi dữ liệu lên Thingsboard

Quá trình gửi dữ liệu lên dashboard được thiết lập với chu kỳ 5 giây mỗi lần. Điều này đảm bảo các thông số được cập nhật liên tục và phản ánh chính xác hoạt động của hệ thống. Với chu kỳ này, hệ thống không chỉ tối ưu hóa việc giám sát mà còn giảm tải cho mạng lưới và thiết bị, đảm bảo hiệu suất ổn định trong suốt quá trình hoạt động.



Hình 3.33 Lưu đồ thuật toán gửi dữ liệu lên dashboard

Trong quá trình gửi dữ liệu, hệ thống sẽ thu thập các thông số hoạt động như trạng thái hệ thống, chỉ số đo lường, và các tham số giám sát quá trình xử lý dữ liệu. Những thông tin này sẽ được tổ chức và tạo thành một request MQTT, cho phép gửi dữ liệu lên dashboard. MQTT giúp đảm bảo quá trình truyền tải diễn ra nhanh chóng và ổn định, phù hợp với giám sát thời gian thực.

3.4 Tổng kết

Phần thiết kế hệ thống đã được xây dựng cẩn thận với mục tiêu đảm bảo khả năng vận hành ổn định, tính linh hoạt cao, và hiệu suất vượt trội trong môi trường thực tế. Hệ thống sử dụng module SIM7600 tích hợp với Raspberry Pi để thiết lập giao tiếp SMS, đảm bảo rằng dữ liệu từ các bộ sensor được thu thập một cách liên tục và không bị gián đoạn, bất kể số lượng thiết bị đầu cuối tăng lên. Điều này đạt được nhờ việc triển khai giao thức UART cho phép truyền nhận dữ liệu ổn định giữa các thiết bị phần cứng và module SIM, qua đó thông tin được chuyển tải trực tiếp vào bộ nhớ DMA, tối ưu cho hoạt động với lượng dữ liệu lớn.

Để tăng cường tính tương thích và khả năng mở rộng, hệ thống được thiết kế để xử lý dữ liệu từ nhiều loại và phiên bản sensor khác nhau. Các gói dữ liệu được kiểm tra cấu trúc trước khi phân tích và xử lý, cho phép hệ thống dễ dàng thích ứng khi có sự thay đổi trong định dạng dữ liệu của sensor. Điều này làm tăng độ bền vững của hệ thống, giúp dễ dàng tích hợp với các phiên bản phần cứng mới mà không cần thay đổi lớn trong cấu trúc phần mềm.

Một phần quan trọng trong thiết kế hệ thống là tích hợp dashboard ThingsBoard cho phép người dùng giám sát toàn bộ hoạt động theo thời gian thực. ThingsBoard hỗ trợ các kết nối MQTT, giúp việc đẩy dữ liệu lên server diễn ra nhanh chóng và liên tục mà không gặp độ trễ lớn. Dashboard cung cấp các thông tin như cường độ kết nối mạng, trạng thái hoạt động của vi xử lý, số lượng gói tin nhận được và gửi đi, giúp người giám sát nắm bắt kịp thời các yếu tố quan trọng của hệ thống. Điều này không chỉ hỗ trợ trong việc quản lý và vận hành hệ thống mà còn giúp nhận biết nhanh các bất thường, sự tăng đột biến về lưu lượng dữ liệu hoặc lỗi trong các gói tin.

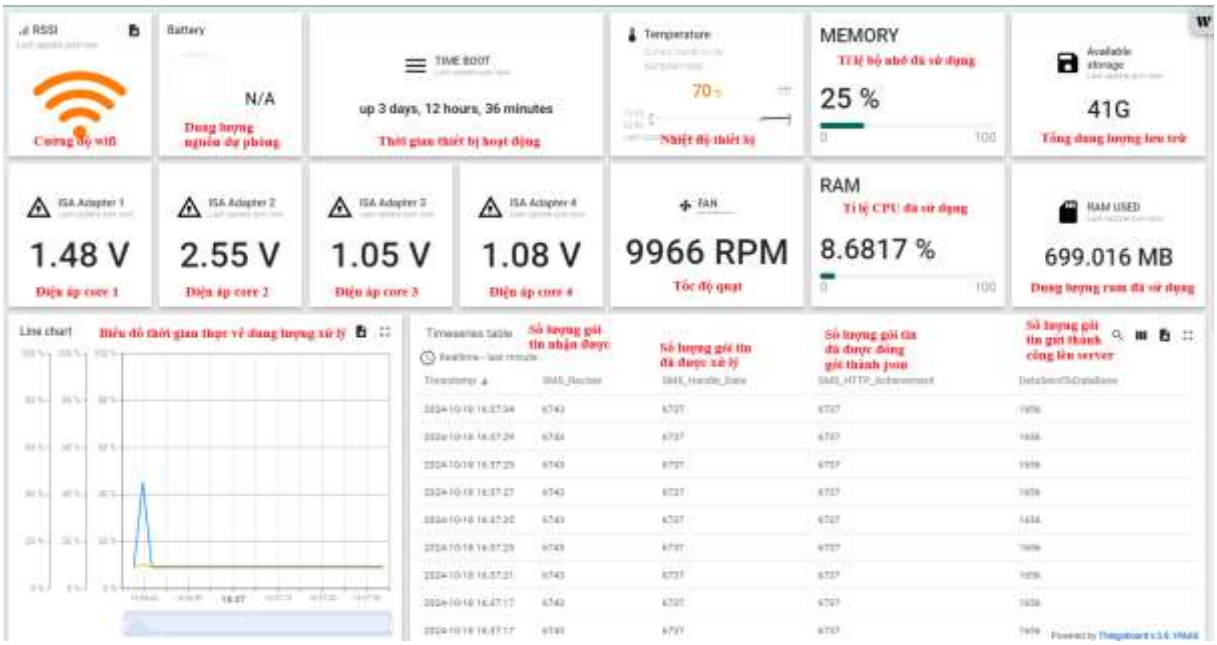
CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ

4.1 Kết quả thử nghiệm

Sau khi tích hợp thiết kế vào hệ thống thu thập dữ liệu, hệ thống đã vận hành ổn định và đạt các yêu cầu đặt ra. Thiết bị gateway đảm bảo rằng các gói tin từ các thiết bị không có khả năng kết nối trực tiếp với internet vẫn có thể được chuyển tiếp dữ liệu một cách an toàn lên server.

Hình 4.1 Hình ảnh phần cứng thiết bị gateway

Toàn bộ quá trình hoạt động được giám sát liên tục qua dashboard ThingsBoard, hiển thị đầy đủ các thông số và chỉ số hoạt động của hệ thống. Các thông tin này bao gồm số lượng gói tin nhận được, tình trạng mạng, trạng thái CPU của thiết bị gateway, và các chỉ số liên quan khác. Nhờ đó, người giám sát có thể theo dõi trạng thái hệ thống theo thời gian thực và đảm bảo rằng tất cả các thành phần đều đang hoạt động ổn định, không có bất thường xảy ra



Hình 4.2 Giao diện dashboard giám sát

Trong quá trình hoạt động, thiết bị sẽ nhận dữ liệu từ các bộ sensor không có khả năng gửi trực tiếp lên server. Thay vào đó, các sensor sẽ gửi dữ liệu qua SMS đến gateway, đảm nhiệm vai trò trung gian tiếp nhận và xử lý dữ liệu trước khi truyền lên server. Gateway đảm bảo thu thập, xử lý, và chuyển dữ liệu đến hệ thống trung tâm, giúp giám sát liên tục và đảm bảo dữ liệu từ tất cả các sensor, kể cả những bộ giới hạn kết nối, được cập nhật đầy đủ trên server.

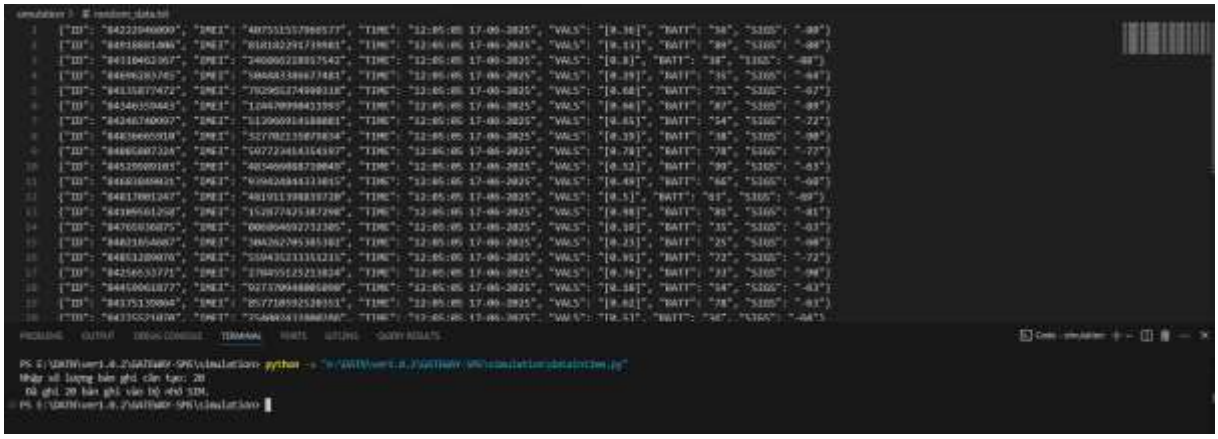
Nhằm đánh giá tính khả thi và độ ổn định của thiết bị gateway trong quá trình vận hành, một quy trình kiểm thử được thiết kế để mô phỏng thực tế thu nhận dữ liệu từ các nút cảm biến. Cụ thể, một thiết bị phát tin nhắn SMS được sử dụng như một nguồn dữ liệu mô phỏng. Thiết bị này sẽ gửi các gói tin mẫu tương tự như dữ liệu do các cảm biến tạo ra đến gateway thông qua SMS. Thông qua quá trình này, khả năng tiếp nhận, xử lý và phản hồi dữ liệu đầu vào của gateway sẽ được kiểm tra toàn diện, từ đó đánh giá hiệu năng hoạt động và độ ổn định của hệ thống trong điều kiện gần với thực tế triển khai.

Mô phỏng thử nghiệm hệ thống nhận dữ liệu		
Chu kì thời gian	Số lượng gói tin gửi đến	Số lượn gói tin gateway nhận
10 phút	5	5
60 phút	30	30
120 phút	60	60

Bảng 4.1 Mô phỏng thử nghiệm hệ thống nhận dữ liệu

Khi thực hiện điều chỉnh thiết bị để mô phỏng gửi dữ liệu liên tục với chu kỳ 1 phút và số lượng lớn dữ liệu, hệ thống cho thấy khả năng xử lý ổn định, đảm bảo cả về số lượng tin nhắn gửi đi lẫn tin nhắn nhận về. Phương pháp kiểm thử cưỡng bức này giúp hệ thống hoạt động ở mức tải cao hơn điều kiện bình thường, từ đó phát hiện các lỗi tiềm ẩn có thể không xuất hiện trong điều kiện vận hành thường ngày. Qua đó, có thể cải tiến khả năng xử lý và tăng tính ổn định của hệ thống, đảm bảo rằng các thông số quan trọng như tốc độ truyền tải và tính toàn vẹn dữ liệu luôn được duy trì ngay cả trong các tình huống tải cao.

Để đánh giá hiệu quả của quá trình xử lý dữ liệu tại gateway, một chương trình mô phỏng đã được xây dựng nhằm tái hiện tình huống gateway nhận đồng thời nhiều gói dữ liệu từ các cảm biến. Chương trình mô phỏng này có chức năng tự động tạo ra các gói tin với tốc độ cao, sau đó ghi trực tiếp vào bộ nhớ nhận dữ liệu của mô-đun SMS, tương đương với việc gateway nhận dữ liệu từ nhiều cảm biến cùng lúc. Quá trình này cho phép kiểm tra khả năng gateway xử lý dữ liệu theo thời gian thực, đánh giá hiệu năng xử lý đồng thời (concurrent processing), tốc độ phản hồi và khả năng ổn định của hệ thống khi đối mặt với lượng lớn dữ liệu đầu vào trong thời gian ngắn.



Hình 4.3 Chương trình mô phỏng sensor gửi dữ liệu SMS

Việc thực hiện chạy chương trình mô phỏng nhằm tái hiện quá trình truyền dữ liệu từ một số lượng lớn sensor đến thiết bị gateway trong một khoảng thời gian xác định. Thông qua mô phỏng này, hệ thống cho phép đánh giá chính xác khả năng tiếp nhận, xử lý và truyền tải dữ liệu của gateway. Từ đó, có thể xác định được liệu gateway có đủ năng lực để xử lý hiệu quả khối lượng dữ liệu lớn trong môi trường thực tế, đảm bảo không xảy ra hiện tượng quá tải hoặc mất mát dữ liệu.

Mô phỏng quá trình xử lý dữ liệu			
Số lượng sensor được mô phỏng	Số lượng dữ liệu được xử lý	Số lượng dữ liệu được gửi lên server	Thời gian hoàn thành
20	20	20	15 giây
30	30	30	18 giây
40	40	40	23 giây

Bảng 4.2 Mô phỏng quá trình xử lý dữ liệu

Quá trình xử lý dữ liệu có thời gian thực thi kéo dài do sử dụng giao thức HTTPS để gửi dữ liệu lên server, đòi hỏi phải chờ phản hồi từ server trước khi tiếp tục. Hơn nữa, các gói tin cần được gửi lần lượt để đảm bảo độ chính xác và an toàn dữ liệu. Mặc dù thời gian xử lý mỗi gói tin khá lâu, hệ thống vẫn đáp ứng yêu cầu cập nhật dữ liệu kịp thời và không gây ảnh hưởng đến các tác vụ khác. Điều này cho thấy hệ thống có khả năng hoạt động ổn định ngay cả khi phải tuân thủ các yêu cầu nghiêm ngặt về bảo mật và thứ tự xử lý dữ liệu.

4.2 Kết quả tính ổn định thiết bị

Trước khi tích hợp thiết bị gateway vào hệ thống, việc truyền dữ liệu qua kết nối 3G/4G không ổn định đã dẫn đến tình trạng mất dữ liệu, gây ảnh hưởng đến khả năng cập nhật dữ liệu đúng chu kỳ. Do thiếu sự giám sát liên tục và không có các cơ chế tự động để phát hiện lỗi, các sự cố mất dữ liệu chỉ có thể được xác định khi kiểm tra thủ công. Điều này khiến việc xác định nguồn gốc sự cố gặp khó khăn, đặc biệt khi khối lượng dữ liệu lớn và yêu cầu truyền tải liên tục.

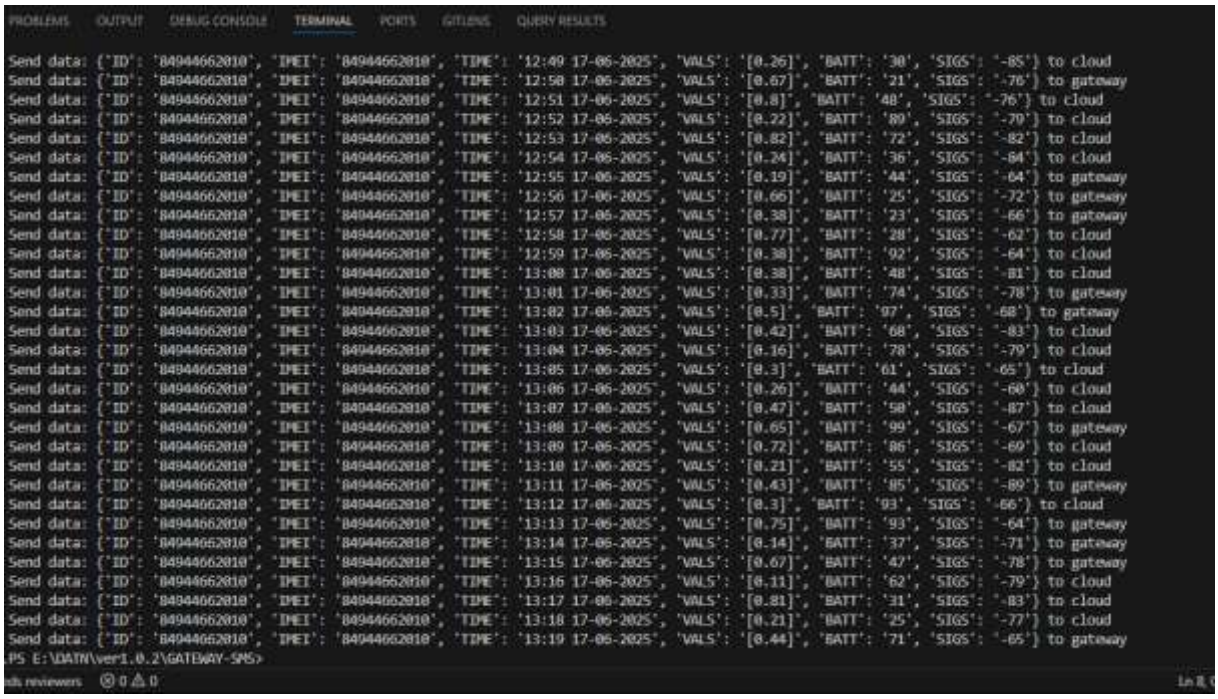
Việc thiếu thiết bị gateway để đảm bảo tính liên tục của việc truyền dữ liệu cũng đồng nghĩa với việc không thể kiểm soát tự động các vấn đề mất kết nối hoặc lỗi gói tin. Các thiết bị không tự động thông báo tình trạng kết nối khiến người giám sát không kịp thời nắm bắt được các trục trặc trong quá trình truyền tải.

Với dashboard giám sát, người dùng có thể dễ dàng theo dõi số lượng dữ liệu thiết bị nhận được từ các bộ sensor cũng như số lượng gói tin đã được xử lý.

Việc lấy dữ liệu giám sát và lọc thông tin từ server theo thời gian cho phép tạo ra các thống kê chi tiết về hiệu suất và độ ổn định của hệ thống trong từng khoảng thời gian cụ thể. Điều này giúp người quản lý dễ dàng phát hiện và so sánh các biến động của hệ thống theo chu kỳ, đánh giá mức độ ổn định và xác định các khoảng thời gian mà việc truyền tải dữ liệu bị gián đoạn hoặc có dấu hiệu bất thường. Qua đó, các biện pháp

cải tiến hoặc điều chỉnh hệ thống có thể được áp dụng nhanh chóng nhằm tối ưu hóa hiệu suất và đảm bảo tính toàn vẹn của quá trình thu thập, xử lý dữ liệu.

Để đánh giá tính ổn định của gateway khi đưa vào hoạt động trong hệ thống, tôi đã xây dựng một chương trình mô phỏng quá trình làm việc thực tế của sensor trong điều kiện kết nối 3G/4G không ổn định. Chương trình này sẽ tự động gửi dữ liệu từ sensor và thực hiện việc gửi ngẫu nhiên các gói tin đến cả cloud và gateway. Mục đích là kiểm tra xem gateway có thể xử lý và nhận đúng các gói tin bất ngờ gửi đến từ sensor trong các tình huống kết nối không ổn định hay không. Qua quá trình này, tôi có thể đánh giá được độ tin cậy và khả năng hoạt động ổn định của thiết bị gateway trong môi trường thực tế.



Hình 4.4 Mô phỏng sensor có kết nối không ổn định

Chương trình mô phỏng được thiết kế nhằm tái hiện quá trình truyền dữ liệu từ các cảm biến với chu kỳ cố định là 1 phút/lần. Trong mỗi chu kỳ, hệ thống sẽ lựa chọn ngẫu nhiên giữa hai kịch bản: gửi dữ liệu lên cloud thông qua kết nối ổn định mô phỏng điều kiện hoạt động bình thường hoặc gửi dữ liệu qua gateway trong điều kiện cảm biến bị mất kết nối mô phỏng sự cố gián đoạn mạng.

Thông qua mô hình này, quá trình hoạt động của thiết bị gateway được đánh giá một cách toàn diện, bao gồm khả năng duy trì ổn định trong suốt thời gian mô phỏng, cũng như khả năng tiếp nhận đầy đủ các gói tin từ cảm biến trong cả hai trạng thái kết nối. Điều này giúp xác định liệu gateway có xảy ra hiện tượng mất gói tin, gián đoạn xử lý hoặc lỗi đồng bộ trong môi trường hoạt động thực tế hay không.

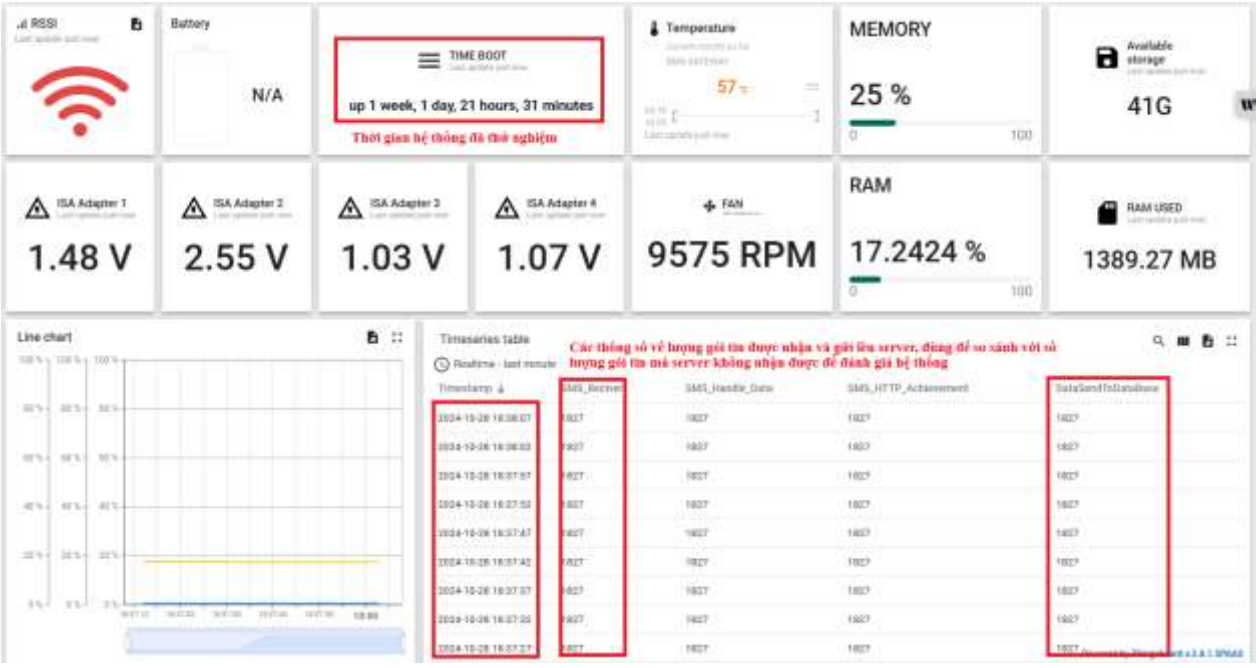
Mô phỏng quá trình sensor gửi dữ liệu và mất kết nối 3G/4G				
Thời gian hoạt động	Số gói tin sensor gửi	Số gói tin server nhận được	Số gói tin mất	Số gói tin được gateway nhận
30 phút	30	12	18	18
60 phút	60	38	22	22

Bảng 4.3 Mô phỏng quá trình sensor gửi dữ liệu và mất kết nối 3G/4G

Qua chạy thử và so sánh độ hiệu quả của hệ thống ta có thể thấy được khi kết hợp gateway vào hệ thống, lượng dữ liệu bị mất trong quá trình các bộ sensor gửi dữ liệu sẽ được gateway nhận và gửi lại lên server, đảm bảo không mất dữ liệu trong quá trình hoạt động.

Gateway được thiết kế với khả năng xử lý các cấu trúc dữ liệu khác nhau, giúp nó có thể tích hợp và tương thích với nhiều phiên bản sensor trong hệ thống. Nhờ khả năng này, gateway có thể tiếp nhận và phân tích dữ liệu từ các phiên bản sensor khác nhau, đảm bảo điều hướng và xử lý thông tin một cách hiệu quả. Điều này giúp giám sát chặt chẽ tình trạng hoạt động của từng phiên bản sensor, nâng cao tính linh hoạt và hiệu suất của hệ thống trong việc xử lý dữ liệu từ các thiết bị khác nhau.

Để đánh giá chính xác hệ thống, cần triển khai thử nghiệm dài hạn với nhiều sensor để theo dõi hiệu suất và tình trạng hoạt động chi tiết, qua đó so sánh với dữ liệu có được ở server để đánh giá hiệu quả của hệ thống.



Hình 4.5 Dashboard theo dõi quá trình thử nghiệm

Với việc tích hợp gateway vào hệ thống, có thể đảm bảo rằng các gói tin từ sensor sẽ không bị mất do gián đoạn kết nối và gateway sẽ đóng vai trò như một kết nối dự phòng ổn định. Khi thử nghiệm trong môi trường hoạt động ổn định của sensor, kết quả cho thấy việc thêm gateway giúp tối ưu hoá truyền dữ liệu và đảm bảo rằng dữ liệu luôn được cập nhật, duy trì hiệu suất cao cho hệ thống trong thời gian dài.

Tất cả các thông số vận hành cùng với dữ liệu thu thập từ hệ thống cảm biến và thiết bị gateway trong suốt quá trình mô phỏng sẽ được hiển thị trực quan và thống kê chi tiết trên màn hình dashboard. Giao diện dashboard được thiết kế theo thời gian thực (real-time), cho phép người vận hành dễ dàng theo dõi toàn bộ tiến trình hoạt động của hệ thống, bao gồm các chỉ số quan trọng như: số lượng gói tin đã gửi, tỷ lệ thành công/thất bại trong quá trình truyền, tần suất gửi dữ liệu, thời gian mất kết nối và mức độ tải của gateway.

Thông qua việc trực quan hóa dữ liệu trên dashboard, người dùng không chỉ nắm bắt được hiện trạng hệ thống tại từng thời điểm mà còn có khả năng phát hiện kịp thời các bất thường, sự cố hoặc tình huống quá tải. Điều này đóng vai trò then chốt trong việc hỗ trợ ra quyết định, tối ưu hóa vận hành, cũng như nâng cao độ tin cậy và hiệu suất của hệ thống cảm biến – gateway – cloud.

4.3 Nhận xét phân tích kết quả đạt được

Qua quá trình thử nghiệm, thiết bị đã chứng minh khả năng đáp ứng hiệu quả các chức năng cốt lõi, vận hành một cách ổn định và đáng tin cậy. Cụ thể:

Đảm bảo nhận và cập nhật dữ liệu liên tục lên server: Thiết bị cho thấy khả năng tiếp nhận chính xác và liên tục các gói tin từ các sensor và nhanh chóng cập nhật lên server thông qua giao thức HTTPS. Điều này đảm bảo dữ liệu luôn được gửi đi đúng thời điểm, duy trì sự ổn định của hệ thống và giúp loại bỏ tình trạng mất dữ liệu trong quá trình truyền tải.

Khả năng xử lý khối lượng dữ liệu lớn trong khoảng thời gian ngắn: Thiết bị đã chứng minh khả năng tiếp nhận và xử lý liên tục lượng dữ liệu lớn từ nhiều thiết bị sensor khác nhau với chu kỳ gửi liên tục mỗi phút một lần. Điều này cho thấy thiết bị có thể hoạt động tốt ngay cả trong môi trường đòi hỏi cường độ xử lý cao, với khả năng đáp ứng mà không gây ra gián đoạn hoặc quá tải cho hệ thống.

Hoạt động ổn định với các cấu trúc dữ liệu đa dạng: Nhờ khả năng xử lý nhiều phiên bản sensor khác nhau, thiết bị có thể nhận diện và xử lý linh hoạt các cấu trúc dữ liệu đa dạng từ các phiên bản khác nhau của sensor. Điều này hỗ trợ rất lớn trong việc giám sát, kiểm tra tính tương thích, đồng thời mở rộng khả năng ứng dụng của thiết bị trong các hệ thống có tính mở rộng cao.

Giám sát hoạt động hệ thống và thống kê dữ liệu một cách chính xác: Nhờ tích hợp với ThingsBoard, thiết bị có thể giám sát và thống kê dữ liệu thu thập trong thời gian thực, cung cấp thông tin đầy đủ về số lượng gói tin được nhận, xử lý, và gửi lên server. Bằng việc theo dõi thông số này, người quản lý dễ dàng đánh giá được hiệu suất và tình trạng hoạt động của thiết bị, phát hiện kịp thời các vấn đề bất thường và điều chỉnh nhanh chóng khi cần.

Tối ưu hóa quá trình kiểm thử và phát hiện lỗi tiềm ẩn: Thiết bị đã được thử nghiệm trong điều kiện hoạt động cường bức, với tốc độ và khối lượng dữ liệu vượt mức thông thường. Kết quả cho thấy thiết bị không chỉ hoạt động ổn định mà còn có khả năng phát hiện các lỗi tiềm ẩn trong hệ thống, đảm bảo rằng thiết bị sẽ hoạt động hiệu quả và chính xác ngay cả khi đối mặt với điều kiện tải cao.

Đảm bảo an toàn và bảo mật dữ liệu khi truyền tải: Giao thức HTTPS cùng với hệ thống kiểm tra và xác thực kết nối trước khi gửi giúp bảo vệ dữ liệu trong quá trình truyền tải, giảm thiểu nguy cơ mất mát hoặc sai lệch dữ liệu. Thiết bị liên tục kiểm tra tình trạng kết nối với server để tránh các lỗi do mất kết nối, đồng thời có cơ chế gửi lại dữ liệu nếu kết nối bị gián đoạn, đảm bảo dữ liệu luôn được cập nhật đúng thời điểm.

4.4 Tổng kết

Qua quá trình thử nghiệm và vận hành thực tế, ta có thể khẳng định rằng thiết bị đã hoạt động một cách ổn định và đáp ứng tốt các yêu cầu đặt ra. Vai trò của thiết bị trong việc giảm thiểu rủi ro mất mát dữ liệu, hỗ trợ lưu trữ và đảm bảo tính toàn vẹn của dữ liệu được thể hiện rõ ràng. Thiết bị không chỉ đóng vai trò quan trọng trong việc thu thập và xử lý dữ liệu từ các sensor mà còn giúp người vận hành dễ dàng theo dõi, đánh giá tình trạng hoạt động của hệ thống qua các công cụ giám sát thời gian thực.

Bên cạnh đó, thiết bị còn hỗ trợ hệ thống IoT trong việc cung cấp các số liệu thống kê chi tiết và kịp thời, đảm bảo rằng người vận hành luôn nắm được tình hình hoạt động và tình trạng kết nối của hệ thống. Nhờ việc sử dụng gateway, quá trình truyền tải và đồng bộ dữ liệu trở nên liền mạch hơn, đặc biệt trong những điều kiện mạng yếu hoặc không ổn định, giúp ngăn chặn tình trạng gián đoạn và thiếu sót dữ liệu.

Tính hiệu quả và sự cần thiết của thiết bị đối với hệ thống IoT thu thập dữ liệu là không thể phủ nhận. Thiết bị không chỉ tăng cường độ tin cậy của hệ thống mà còn giúp quản lý dễ dàng hơn, giảm thiểu rủi ro về dữ liệu và nâng cao hiệu quả vận hành tổng thể. Đây là một phần thiết yếu giúp hệ thống IoT hoạt động ổn định và hiệu quả trong các môi trường khác nhau.

KẾT LUẬN

Qua quá trình thiết kế, triển khai và thử nghiệm mô phỏng, nhóm nhận thấy thiết bị gateway đã được thiết kế khá thành công với mục tiêu đảm bảo khả năng kết nối với 100% gói tin gửi đến đều được xử lý và giao tiếp hiệu quả giữa các thiết bị IoT trong hệ thống, đảm bảo được 100% trong môi trường thử nghiệm. Kết quả thử nghiệm ban đầu cho thấy gateway hoạt động đúng như mong đợi, có thể nhận dữ liệu từ các thiết bị thông qua các gói tin SMS và truyền dữ liệu lên nền tảng ThingsBoard một cách ổn định. Việc tích hợp với ThingsBoard giúp hệ thống hiển thị và giám sát dữ liệu theo thời gian thực qua giao diện dashboard trực quan, từ đó đơn giản hóa việc quản lý, nâng cao tính minh bạch và khả năng phân tích dữ liệu. Tuy nhiên, do chưa thực hiện được thử nghiệm trong môi trường thực tế với các điều kiện khắc nghiệt hơn, cũng như chưa xử lý hết các vấn đề phát sinh từ môi trường vận hành, nên việc đánh giá toàn diện về hiệu suất và độ ổn định vẫn còn hạn chế. Nhóm sẽ tiếp tục nghiên cứu, cải tiến và phát triển phiên bản mới tối ưu hơn để có thể ứng dụng trực tiếp trong hệ thống thực tế.

TÀI LIỆU THAM KHẢO

[1] NIKOLAOS A. PANTAZIS and DIMITRIOS D. VERGADOS, "Article," *A SURVEY ON POWER CONTROL ISSUES IN WIRELESS SENSOR NETWORKS,* 2007.

[2] Fei Hu, Jim Ziobro, Jason Tillett and Neeraj K. Sharma, "Article," *Secure Wireless Sensor Networks: Problems and Solutions.*

[3] Moshaddique Al Ameen and Kyung Kwak, "Article," *Social Issues in Wireless Sensor Networks with Healthcare Perspective,* 2011.

[4] Dheyab Salman Ibrahim, Abdullah Farhan Mahdi and Qahtan M. Yas, "Article," *Challenges and Issues for Wireless Sensor Networks,* 2020.

[5] Lance Doherty, Jonathan Simon and Thomas Watteyne, "Article," *Wireless Sensor Network Challenges and Solutions,* 2012.

[6] Oladayo Olufemi Olakanmi and Adedamola Dada, "Article," *Wireless Sensor Networks (WSNs): Security and Privacy Issues and Solutions,* 2020.

[7] Nguyễn Văn Trường, Dương Tuấn Anh and Nguyễn Quý Sỹ, "Bài báo," *Khảo sát các vấn đề bảo mật trong mạng cảm biến không dây,* 2020.

[8] A. ADULYASAS, "Article," *Connected Coverage Assurance for Sensor Scheduling in Wireless Sensor Networks,* 2015.

[9] M. C. Saranya, "Article," *Blockchain-based Deep Learning Routing in WSN. Cuestiones de Fisioterapia.,* 2025.

[10] MD Jiabul Hoque, Md. Saiful Islam, Istiaque Ahmed and Md. Nurullah, "Article," *Enhancing Precision Agriculture Effciency Through Edge Enhancing Precision Agriculture Effciency Through Edge,* 2024.

[11] Luís Miguel Pires and José Gomes, "Article," *River Water Quality Monitoring Using LoRa-Based IoT,* 2024.