

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

**Lập trình cho thiết bị nhúng giám sát vị trí
NaviTracker**

Hà Thế Hiển

hien.ht215575@sis.hust.edu.vn

Ngành Kỹ thuật máy tính

Giảng viên hướng dẫn: PGS. TS. Lã Thế Vinh

Chữ kí GVHD

Khoa: Kỹ thuật máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 06/2025

LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành tới thầy giáo hướng dẫn, PGS.TS. Lã Thế Vinh, người đã tận tình chỉ bảo, hỗ trợ và động viên em trong suốt quá trình thực hiện đồ án tốt nghiệp này. Em cũng xin cảm ơn gia đình, những người luôn ở bên, ủng hộ và tạo mọi điều kiện thuận lợi để em hoàn thành tốt nhất công việc của mình. Cuối cùng, cảm ơn bạn bè đã chia sẻ và hỗ trợ em vượt qua những khó khăn trong thời gian qua. Em trân trọng và biết ơn những nỗ lực, sự kiên trì và quyết tâm của chính bản thân, đã không ngừng cố gắng để hoàn thiện đồ án này với kết quả tốt nhất.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Việc giám sát và quản lý vị trí các đối tượng, phương tiện ngày càng trở nên quan trọng trong nhiều lĩnh vực như vận tải, logistic, và an ninh. Tuy nhiên, các giải pháp truyền thống thường chỉ cung cấp thông tin vị trí tức thời, thiếu khả năng phân tích lịch sử di chuyển, cảnh báo khi vượt ra khỏi khu vực cho phép (geofence), và khó tích hợp các thông số hệ thống quan trọng như điện áp, trạng thái kết nối vào trong một nền tảng chung. Một số hướng tiếp cận phổ biến hiện nay bao gồm sử dụng thiết bị định vị GPS đơn giản, hệ thống quản lý dựa trên mạng di động GSM, hoặc các nền tảng IoT để quản lý và phân tích dữ liệu vị trí. Các giải pháp này dù hiệu quả nhưng thường bị giới hạn trong khả năng mở rộng, tích hợp, và quản lý dữ liệu tập trung.

Trong đồ án này, tôi lựa chọn tiếp cận theo hướng phát triển phần mềm cho thiết bị nhúng NaviTracker với nền tảng IoT ThingsBoard. Tôi lựa chọn giải pháp này do khả năng linh hoạt cao, dễ dàng tích hợp nhiều loại dữ liệu và hỗ trợ các tính năng quản lý nâng cao như geofence, theo dõi lịch sử di chuyển, và giám sát các thông số hệ thống.

Giải pháp của tôi bao gồm phát triển một phần mềm nhúng trên NaviTracker có khả năng thu thập liên tục tọa độ GPS, tính toán quãng đường di chuyển, và gửi dữ liệu vị trí cùng các thông số hệ thống (như trạng thái pin, chất lượng tín hiệu GPS) tới server ThingsBoard thông qua giao thức MQTT. Trên ThingsBoard, tôi xây dựng dashboard trực quan hiển thị thông tin vị trí hiện tại, tổng hợp quãng đường đã đi, lịch sử hành trình chi tiết, quản lý geofence linh hoạt và cảnh báo khi thiết bị ra vào các khu vực được định nghĩa trước.

Đóng góp chính của đồ án là xây dựng một hệ thống hoàn chỉnh và tích hợp sâu giữa thiết bị nhúng NaviTracker và nền tảng ThingsBoard, cung cấp đầy đủ các tính năng giám sát và quản lý vị trí. Kết quả đạt được sau cùng là hệ thống hoạt động ổn định, trực quan và đáp ứng tốt các yêu cầu thực tế về quản lý và giám sát đối tượng di chuyển, đã được thử nghiệm thực tế và cho kết quả chính xác, đáng tin cậy.

Sinh viên thực hiện
(Ký và ghi rõ họ tên)

ABSTRACT

Mục này khuyến khích sinh viên viết lại mục “Tóm tắt” đề án tốt nghiệp ở trang trước bằng tiếng Anh. Phần này phải có đầy đủ các nội dung như trong phần tóm tắt bằng tiếng Việt. Sinh viên không nhất thiết phải trình bày mục này.

Nhưng nếu lựa chọn trình bày, sinh viên cần đảm bảo câu từ và ngữ pháp chuẩn tắc, nếu không sẽ có tác dụng ngược, gây phản cảm.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án	3
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	4
2.1 Khảo sát hiện trạng	4
2.1.1 Hạn chế của các hệ thống hiện tại	4
2.1.2 Các yêu cầu chức năng của hệ thống	5
2.1.3 Các yêu cầu phi chức năng của hệ thống.....	5
2.2 Phân tích hướng giải quyết.....	6
2.2.1 Mô hình hệ thống.....	7
2.2.2 Lựa chọn công nghệ và nền tảng.....	8
2.2.3 Đánh giá tính khả thi	9
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	11
3.1 Thiết bị NaviTracker	11
3.2 Ngôn ngữ và môi trường phát triển	12
3.3 Giao thức truyền thông MQTT.....	12
3.4 Nền tảng hiển thị: ThingsBoard.....	13
3.5 Các thư viện và phần mềm hỗ trợ	14
3.6 Tổng kết chương	14
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG	16
4.1 Kiến trúc tổng thể phần mềm.....	16

4.2 Thiết kế phân lớp phần mềm.....	17
4.2.1 Lớp giao tiếp phần cứng	17
4.2.2 Lớp xử lý dữ liệu.....	19
4.2.2.1 Các bản tin từ module GNSS ublox NEO-M9 . . .	19
4.2.2.2 Luồng xử lý dữ liệu	21
4.2.3 Lớp truyền thông.....	22
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....	24
5.1 Cài đặt và cấu hình Thingsboard	24
5.2 Cấu hình NAT để truy cập từ Internet.....	25
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	26
6.1 Kết luận	26
6.2 Hướng phát triển.....	26
TÀI LIỆU THAM KHẢO.....	29
PHỤ LỤC.....	30
A. HƯỚNG DẪN VIẾT ĐỒ ÁN TỐT NGHIỆP	30
A.1 Ngành học.....	31
A.2 Đánh dấu (bullet) và đánh số (numering)	31
A.3 Cách thêm bảng	32
A.4 Chèn hình ảnh	32
A.5 Tài liệu tham khảo	32
A.5.0.0.1 Cách liệt kê	32
A.5.0.0.2 Các loại tài liệu tham khảo	33
A.6 Cách viết phương trình và công thức toán học.....	33
A.7 Quy cách đóng quyển.....	33
B. ĐẶC TẢ USE CASE.....	35
B.1 Đặc tả use case “Thông kê tình hình mượn sách”	35
B.2 Đặc tả use case “Đăng ký làm thẻ mượn”	35

DANH MỤC HÌNH VẼ

Hình 2.1	Sơ đồ hệ thống	8
Hình 4.1	Kiến trúc tổng thể phần mềm	17
Hình 4.2	Luồng xử lý dữ liệu lớp phần cứng	18
Hình A.1	Internet vạn vật	32
Hình A.2	Quy cách đóng quyển đồ án	34
Hình A.3	Quy cách đóng quyển đồ án	34

DANH MỤC BẢNG BIỂU

Bảng 4.1	Cấu trúc bản tin NMEA-RMC	20
Bảng 4.2	Cấu trúc bản tin UBX-CFG-GNSS	20
Bảng 4.3	Cấu trúc bản tin UBX-NAV-STATUS	20
Bảng 4.4	Cấu trúc bản tin UBX-RXM-SFRBX	21
Bảng A.1	Table to test captions and labels.	32

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
EUD	Phát triển ứng dụng người dùng cuối(End-User Development)
GWT	Công cụ lập trình Javascript bằng Java của Google (Google Web Toolkit)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
IaaS	Dịch vụ hạ tầng

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong những năm gần đây, công nghệ định vị toàn cầu đã phát triển mạnh mẽ và ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Các thiết bị định vị được tích hợp trong ô tô, thiết bị vận tải, thiết bị giám sát cá nhân, và hệ thống quản lý tài sản ngày càng phổ biến. Tuy nhiên, việc giám sát vị trí một cách chính xác và liên tục vẫn đang gặp nhiều thách thức như độ ổn định của tín hiệu vệ tinh, khả năng quản lý dữ liệu thu được, và sự hạn chế của các thiết bị giám sát truyền thống trong việc tương tác với nền tảng quản lý từ xa.

Thực tế hiện nay, nhu cầu giám sát và quản lý phương tiện vận tải, tài sản hoặc con người trong thời gian thực ngày càng tăng cao, đặc biệt là trong các ngành logistics, an toàn giao thông, quản lý đội xe, và cả trong giám sát người già, trẻ nhỏ. Việc xác định được chính xác vị trí và tình trạng hoạt động của đối tượng theo dõi sẽ giúp tối ưu hóa các hoạt động quản lý, nâng cao hiệu quả vận hành, giảm thiểu rủi ro và tiết kiệm chi phí vận hành cho doanh nghiệp cũng như cá nhân.

Nếu giải quyết được vấn đề giám sát và quản lý vị trí một cách liên tục và chính xác, các doanh nghiệp và cá nhân không chỉ được hưởng lợi từ việc cải thiện đáng kể hiệu quả quản lý và an toàn, mà công nghệ này còn có thể mở rộng ứng dụng sang các lĩnh vực khác như quản lý đô thị thông minh, giám sát môi trường, và hỗ trợ khẩn cấp trong các tình huống thiên tai. Từ đó, việc nghiên cứu và phát triển một hệ thống giám sát vị trí với độ chính xác và tin cậy cao trở thành vấn đề cấp thiết và có ý nghĩa thực tiễn quan trọng hiện nay.

1.2 Mục tiêu và phạm vi đề tài

Hiện nay, với sự phát triển mạnh mẽ của công nghệ định vị toàn cầu (GNSS) và các thiết bị IoT, việc giám sát hành trình, quản lý phương tiện, và đảm bảo an toàn giao thông ngày càng được chú trọng. Nhiều sản phẩm đã được phát triển nhằm đáp ứng nhu cầu theo dõi và quản lý vị trí theo thời gian thực, trong đó có các thiết bị định vị GPS tích hợp trên phương tiện, ứng dụng giám sát từ xa qua internet, và các hệ thống quản lý dữ liệu tập trung. Các thiết bị này thường được thiết kế để ghi lại tọa độ, tốc độ, và lộ trình di chuyển, đồng thời gửi dữ liệu lên máy chủ để quản lý và theo dõi.

Tuy nhiên, các hệ thống hiện tại còn tồn tại một số hạn chế như độ chính xác thấp trong điều kiện tín hiệu yếu, dễ bị tấn công giả mạo tín hiệu GNSS, hoặc thiếu khả năng tùy chỉnh phù hợp với các yêu cầu cụ thể của người dùng. Bên cạnh đó,

một số giải pháp giám sát vị trí hiện tại chưa tích hợp được tính năng phát hiện và cảnh báo khi có tín hiệu giả mạo, dẫn đến nguy cơ mất an toàn và thiếu tin cậy trong các ứng dụng quan trọng.

Dựa trên các phân tích trên, đề án này tập trung vào việc phát triển một thiết bị giám sát hành trình tích hợp, có khả năng giám sát vị trí và phát hiện tín hiệu GNSS giả mạo. Thiết bị sẽ được tối ưu về hiệu năng xử lý và mức tiêu thụ năng lượng để đảm bảo phù hợp với các điều kiện hoạt động khắc nghiệt của IoT. Thiết bị không chỉ thu thập tọa độ GNSS, mà còn thực hiện phân tích tín hiệu để phát hiện những bất thường hoặc nguy cơ tấn công giả mạo, từ đó cung cấp thông tin giám sát chính xác và đáng tin cậy cho người sử dụng.

Mục tiêu chính của đề tài là xây dựng một hệ thống giám sát GNSS tối ưu, có khả năng hoạt động ổn định trong điều kiện khắc nghiệt, đồng thời đảm bảo tính an toàn và chính xác của dữ liệu định vị. Hệ thống sẽ được tích hợp lên nền tảng ThingsBoard để hiển thị vị trí hiện tại, lộ trình di chuyển, và các cảnh báo về tín hiệu giả mạo trong thời gian thực. Kết quả của đề tài không chỉ nâng cao tính tin cậy trong giám sát vị trí, mà còn góp phần vào việc bảo vệ an toàn cho các ứng dụng yêu cầu tính chính xác cao trong quản lý phương tiện và tài sản.

1.3 Định hướng giải pháp

Dựa trên nhiệm vụ được xác định trong phần trước, đề án này lựa chọn hướng tiếp cận sử dụng các công nghệ GNSS, xử lý tín hiệu số và nền tảng IoT ThingsBoard để giải quyết bài toán giám sát hành trình và phát hiện tín hiệu GNSS giả mạo. Công nghệ GNSS được sử dụng để xác định vị trí thời gian thực, trong khi xử lý tín hiệu số sẽ giúp phân tích và phát hiện các tín hiệu bất thường. Nền tảng IoT ThingsBoard sẽ đảm nhiệm vai trò hiển thị thông tin vị trí, quản lý dữ liệu, và phát cảnh báo trực quan. Việc lựa chọn các công nghệ này dựa trên khả năng đáp ứng yêu cầu về độ chính xác, tính ổn định và khả năng mở rộng của hệ thống.

Giải pháp của đề án tập trung vào việc thiết kế một thiết bị giám sát hành trình GNSS tích hợp, với khả năng nhận diện tín hiệu giả mạo. Thiết bị sẽ thu thập dữ liệu GNSS từ các cảm biến, xử lý tín hiệu để phát hiện các bất thường, và gửi dữ liệu vị trí cùng cảnh báo về hệ thống giám sát ThingsBoard. Người dùng có thể theo dõi lộ trình, kiểm tra tính xác thực của tín hiệu định vị và nhận cảnh báo ngay lập tức khi có dấu hiệu đáng ngờ.

Định hướng cụ thể của giải pháp bao gồm: xây dựng phần mềm cho thiết bị IoT giám sát hành trình GNSS có khả năng chống giả mạo tín hiệu và tích hợp hệ thống vào nền tảng ThingsBoard để hiển thị và quản lý dữ liệu theo thời gian thực. Kết quả đạt được là một hệ thống giám sát hành trình an toàn, đáng tin cậy, đáp ứng

nhu cầu quản lý vị trí và bảo vệ dữ liệu định vị trong các ứng dụng thực tế.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau:

Chương 2 trình bày về khảo sát và phân tích yêu cầu của hệ thống giám sát GNSS tích hợp. Nội dung chương tập trung vào việc thu thập thông tin từ các tài liệu nghiên cứu, các giải pháp hiện có, cũng như khảo sát nhu cầu thực tiễn của người dùng đối với việc giám sát hành trình và bảo vệ tín hiệu GNSS. Chương này cũng đưa ra phân tích các yêu cầu chức năng và phi chức năng của hệ thống, từ đó xây dựng các tiêu chí để đánh giá hiệu quả của thiết bị giám sát GNSS tích hợp.

Chương 3 giới thiệu về các công nghệ và công cụ được sử dụng trong quá trình phát triển hệ thống. Các công nghệ chính bao gồm GNSS cho định vị, xử lý tín hiệu số để phát hiện giả mạo, và ThingsBoard để quản lý và hiển thị dữ liệu. Ngoài ra, chương này cũng mô tả ngắn gọn về các thư viện và nền tảng hỗ trợ triển khai giải pháp như MQTT, xử lý tín hiệu DSP và các giao thức truyền nhận dữ liệu từ thiết bị IoT lên server.

Chương 4 tập trung vào thiết kế phần mềm cho thiết bị giám sát GNSS tích hợp. Chương này trình bày kiến trúc hệ thống, mô hình phân lớp của phần mềm và các mô-đun chức năng chính. Các phương pháp tối ưu hiệu năng và tiết kiệm năng lượng cũng được thảo luận trong chương này nhằm đảm bảo thiết bị có thể hoạt động ổn định trong điều kiện khắc nghiệt.

Chương 5 mô tả việc triển khai server để thu thập, xử lý và hiển thị dữ liệu từ thiết bị giám sát. Hệ thống server được xây dựng dựa trên nền tảng ThingsBoard với khả năng hiển thị vị trí theo thời gian thực, lưu trữ lịch sử di chuyển, và quản lý các cảnh báo. Các vấn đề liên quan đến bảo mật và độ tin cậy của hệ thống cũng được phân tích trong chương này.

Chương 6 đưa ra kết luận về kết quả đạt được trong quá trình nghiên cứu và phát triển hệ thống giám sát GNSS tích hợp. Chương này tổng hợp lại các đóng góp chính của đề tài, đồng thời phân tích những điểm mạnh và hạn chế của hệ thống. Bên cạnh đó, chương này cũng đề xuất các hướng phát triển trong tương lai, bao gồm việc nâng cao độ chính xác của thuật toán phát hiện giả mạo và tích hợp thêm các chức năng thông minh vào hệ thống.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Trong chương này, tôi sẽ tập trung vào việc khảo sát và phân tích yêu cầu của đề tài. Đầu tiên, tôi tiến hành khảo sát các công nghệ, thiết bị và giải pháp liên quan, nhằm thu thập các thông tin cần thiết phục vụ cho việc phát triển hệ thống. Tiếp theo, dựa trên kết quả khảo sát, tôi sẽ phân tích các hướng giải quyết khả thi, từ đó đề xuất phương án thực hiện phù hợp nhất cho đề tài.

2.1 Khảo sát hiện trạng

Phần này sẽ trình bày tổng quan khảo sát về một số hạn chế của các hệ thống theo dõi hiện nay.

2.1.1 Hạn chế của các hệ thống hiện tại

Hiện nay, các hệ thống giám sát GNSS (Global Navigation Satellite System) được ứng dụng rộng rãi trong nhiều lĩnh vực như giám sát phương tiện, quản lý đội xe, và định vị cá nhân. Tuy nhiên, các hệ thống này vẫn còn tồn tại nhiều hạn chế, ảnh hưởng đến độ chính xác và tính ổn định trong quá trình vận hành.

Một trong những hạn chế lớn nhất của các hệ thống GNSS hiện tại là độ chính xác bị ảnh hưởng bởi nhiều yếu tố ngoại cảnh. Tín hiệu vệ tinh có thể bị nhiễu do các tác động môi trường như thời tiết xấu, địa hình phức tạp (đồi núi, hẻm sâu), hoặc các vùng đô thị với mật độ cao. Điều này dẫn đến sai số định vị, ảnh hưởng trực tiếp đến khả năng giám sát liên tục và đáng tin cậy của hệ thống.

Một vấn đề đáng lo ngại khác là khả năng bị tấn công giả mạo tín hiệu GNSS. Các tín hiệu giả mạo có thể được tạo ra từ các thiết bị phát sóng trái phép (spoofing), làm cho hệ thống giám sát nhận diện sai vị trí thực tế. Điều này gây ra những hậu quả nghiêm trọng trong các ứng dụng yêu cầu độ chính xác cao như giám sát an ninh, quản lý giao thông, hoặc theo dõi phương tiện công cộng.

Các hệ thống GNSS hiện tại thường khó tích hợp với các nền tảng IoT hiện đại do khác biệt về giao thức truyền thông và cách thức xử lý dữ liệu. Việc truyền dữ liệu GNSS lên các nền tảng IoT để giám sát và phân tích thời gian thực thường yêu cầu xử lý trung gian phức tạp, dẫn đến độ trễ trong việc hiển thị vị trí và trạng thái di chuyển.

Trong các hệ thống giám sát di động, việc duy trì giám sát GNSS liên tục tiêu thụ một lượng năng lượng đáng kể. Điều này làm giảm thời gian hoạt động của thiết bị, đặc biệt trong các ứng dụng di động hoặc khi sử dụng các thiết bị IoT với nguồn pin hạn chế.

Nhìn chung, các hệ thống giám sát GNSS hiện tại còn nhiều hạn chế về độ chính xác, khả năng chống giả mạo, tích hợp với hệ thống IoT, và hiệu năng năng lượng. Để khắc phục các vấn đề này, cần có những giải pháp cải tiến cả về mặt công nghệ định vị lẫn tích hợp hệ thống, từ đó đảm bảo hệ thống giám sát GNSS đạt hiệu quả cao và đáp ứng được các yêu cầu thực tiễn.

2.1.2 Các yêu cầu chức năng của hệ thống

Hệ thống giám sát GNSS và geofence cần đảm bảo khả năng giám sát liên tục để đáp ứng các yêu cầu theo dõi vị trí theo thời gian thực. Điều này đòi hỏi hệ thống phải liên tục thu thập và xử lý dữ liệu định vị từ thiết bị, đồng thời truyền tải lên nền tảng ThingsBoard một cách ổn định. Các thông tin như tọa độ, tốc độ di chuyển và trạng thái kết nối phải được cập nhật tức thời, tránh tình trạng mất tín hiệu hoặc gián đoạn, đảm bảo hệ thống có thể theo dõi và giám sát liên tục các đối tượng được quản lý.

Bên cạnh đó, tính năng geofence là một trong những yêu cầu quan trọng của hệ thống. Hệ thống cần cung cấp khả năng thiết lập các vùng giám sát địa lý trên bản đồ, xác định ranh giới khu vực mà đối tượng được phép di chuyển. Khi đối tượng ra khỏi vùng geofence hoặc di chuyển vào khu vực không được phép, hệ thống cần ngay lập tức phát hiện và gửi cảnh báo tới người quản lý. Việc thiết lập, thay đổi hoặc xóa vùng giám sát phải được thực hiện một cách linh hoạt, dễ dàng từ giao diện quản lý.

Ngoài ra, hệ thống cần tích hợp tính năng cảnh báo tự động khi đối tượng di chuyển vượt quá ranh giới geofence. Điều này giúp người dùng kịp thời phát hiện các tình huống bất thường hoặc vi phạm khu vực giám sát. Các cảnh báo cần được gửi dưới dạng thông báo tức thời qua các kênh như tin nhắn, email hoặc qua nền tảng ThingsBoard, giúp người quản lý nhanh chóng đưa ra các biện pháp xử lý.

Hệ thống cũng cần đảm bảo khả năng lưu trữ và quản lý lịch sử di chuyển của đối tượng trong một khoảng thời gian nhất định. Dữ liệu lịch sử cần được tổ chức và truy xuất dễ dàng, hỗ trợ việc phân tích hành trình, đánh giá hiệu quả giám sát, cũng như phục vụ công tác quản lý và báo cáo. Điều này đòi hỏi hệ thống phải có cơ chế lưu trữ dữ liệu an toàn, với dung lượng đủ lớn để đáp ứng nhu cầu sử dụng lâu dài.

2.1.3 Các yêu cầu phi chức năng của hệ thống

Để đảm bảo hiệu quả trong quá trình vận hành, hệ thống giám sát GNSS và geofence cần đáp ứng các yêu cầu phi chức năng về hiệu năng xử lý và tiêu thụ năng lượng. Hiệu năng của hệ thống phải đảm bảo khả năng thu thập và xử lý dữ liệu định vị theo thời gian thực mà không gây ra độ trễ lớn. Điều này đặc biệt quan

trọng trong các ứng dụng giám sát liên tục như theo dõi phương tiện hoặc giám sát an ninh. Đồng thời, hệ thống cần tối ưu hóa tiêu thụ năng lượng, nhất là đối với các thiết bị di động hoặc IoT sử dụng nguồn pin. Việc giảm thiểu tiêu thụ năng lượng giúp kéo dài thời gian hoạt động của thiết bị, hạn chế việc gián đoạn do hết pin.

Một trong những yêu cầu phi chức năng quan trọng khác là bảo mật dữ liệu và chống giả mạo tín hiệu GNSS. Hệ thống cần đảm bảo rằng các dữ liệu định vị được thu thập và truyền tải đều được mã hóa, bảo vệ trước các tấn công mạng như giả mạo vị trí (spoofing) hoặc can thiệp tín hiệu (jamming). Các cơ chế phát hiện tín hiệu bất thường cần được tích hợp, đảm bảo việc giám sát không bị sai lệch do các cuộc tấn công có chủ đích. Việc bảo mật này giúp đảm bảo tính toàn vẹn và độ tin cậy của dữ liệu thu thập được.

Hệ thống cũng cần có tính tương thích cao với các nền tảng IoT hiện đại, đặc biệt là trong việc tích hợp với ThingsBoard để quản lý thiết bị và giám sát từ xa. Việc sử dụng các giao thức truyền thông phổ biến như MQTT giúp tăng cường khả năng tương tác giữa các thiết bị và nền tảng. Điều này tạo điều kiện thuận lợi cho việc triển khai trên nhiều môi trường khác nhau mà không gặp phải các vấn đề tương thích.

Cuối cùng, tính ổn định và độ tin cậy là yếu tố quan trọng trong mọi hệ thống giám sát. Hệ thống cần hoạt động liên tục, ổn định trong thời gian dài mà không gặp phải sự cố hoặc mất tín hiệu. Việc giám sát và tự động khôi phục trong trường hợp lỗi giúp giảm thiểu tác động tiêu cực khi hệ thống bị gián đoạn. Các phương thức kiểm tra, bảo trì và nâng cấp định kỳ cũng cần được thiết lập rõ ràng, giúp duy trì hiệu năng ổn định trong suốt vòng đời của hệ thống.

2.2 Phân tích hướng giải quyết

Sau khi đã khảo sát và tìm hiểu các hệ thống giám sát GNSS hiện tại cũng như các yêu cầu cần thiết, chương này sẽ tập trung vào việc phân tích các hướng giải quyết nhằm xây dựng một hệ thống giám sát GNSS và geofence hiệu quả, đáp ứng các yêu cầu kỹ thuật đã đề ra.

Phần này sẽ trình bày mô hình hệ thống với các thành phần chính, từ đó xác định vai trò và chức năng của từng thành phần trong quá trình giám sát. Tiếp theo, việc lựa chọn công nghệ và nền tảng sẽ được phân tích kỹ lưỡng để đảm bảo hệ thống đạt được hiệu năng cao, tối ưu hóa năng lượng, và có khả năng chống giả mạo tín hiệu GNSS.

Bên cạnh đó, chương cũng sẽ đánh giá tính khả thi của các giải pháp đề xuất, dựa trên các tiêu chí như độ chính xác, độ tin cậy, khả năng mở rộng và tích hợp

với các nền tảng IoT. Kết quả của quá trình phân tích này sẽ làm cơ sở để tiến hành thiết kế và triển khai hệ thống trong các bước tiếp theo.

2.2.1 Mô hình hệ thống

Để xây dựng một hệ thống giám sát hiệu quả, đáp ứng các yêu cầu về giám sát liên tục, và quản lý vị trí theo thời gian thực, mô hình hệ thống cần được thiết kế với cấu trúc rõ ràng, tích hợp các thành phần chính một cách hợp lý để đảm bảo khả năng thu thập, xử lý và truyền tải dữ liệu định vị.

Hệ thống giám sát GNSS và geofence bao gồm ba thành phần chính: thiết bị giám sát, hệ thống trung tâm xử lý dữ liệu và nền tảng hiển thị thông tin. Thiết bị giám sát được lắp đặt trên đối tượng cần theo dõi, có chức năng thu thập tín hiệu vệ tinh để xác định vị trí hiện tại. Thiết bị này cần tích hợp các module GNSS hiện đại, đồng thời có kết nối mạng để truyền dữ liệu về trung tâm.

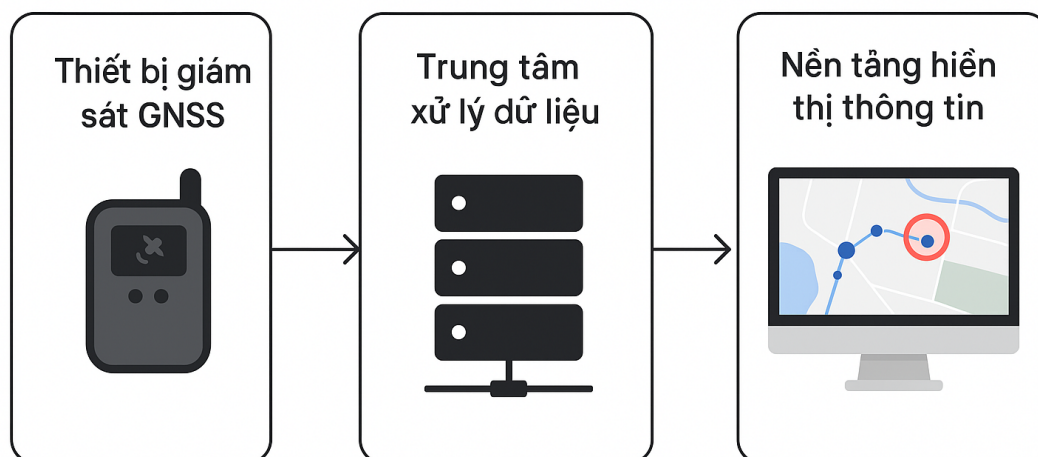
Hệ thống trung tâm xử lý dữ liệu đóng vai trò quan trọng trong việc thu thập, phân tích và lưu trữ dữ liệu định vị. Dữ liệu từ thiết bị GNSS được truyền về trung tâm thông qua các giao thức truyền thông như MQTT hoặc HTTP. Tại đây, dữ liệu được xử lý để phát hiện các sự kiện bất thường như mất tín hiệu hoặc di chuyển ra ngoài vùng geofence. Hệ thống xử lý cần có khả năng phân tích dữ liệu nhanh chóng, đảm bảo độ trễ tối thiểu khi xử lý tín hiệu định vị.

Nền tảng hiển thị thông tin đóng vai trò trực quan hóa dữ liệu, giúp người dùng dễ dàng theo dõi và giám sát đối tượng. Giao diện hiển thị cần tích hợp bản đồ số, thể hiện vị trí hiện tại của đối tượng theo thời gian thực. Ngoài ra, các chức năng cảnh báo khi vượt ranh giới geofence hoặc mất tín hiệu cũng được tích hợp để đảm bảo việc giám sát được diễn ra liên tục và ổn định.

Bên cạnh đó, mô hình hệ thống cũng cần thiết kế các cơ chế lưu trữ dữ liệu để đảm bảo khả năng truy vết và phân tích lịch sử di chuyển. Các dữ liệu định vị được lưu trữ dưới dạng cơ sở dữ liệu thời gian thực trên nền tảng ThingsBoard, hỗ trợ việc truy xuất và phân tích trong các trường hợp cần thiết.

Nhằm đảm bảo tính ổn định và độ tin cậy, mô hình hệ thống cũng cần thiết kế các cơ chế giám sát lỗi và tự động khôi phục khi gặp sự cố. Các thông số như chất lượng tín hiệu, độ trễ truyền tải và dung lượng lưu trữ cần được giám sát thường xuyên để kịp thời phát hiện và khắc phục các sự cố phát sinh.

Tóm lại, mô hình hệ thống giám sát GNSS và geofence được thiết kế với cấu trúc gồm ba thành phần chính: thiết bị giám sát, trung tâm xử lý dữ liệu và nền tảng hiển thị. Các thành phần này liên kết chặt chẽ với nhau, đảm bảo hệ thống hoạt động ổn định, đáp ứng các yêu cầu về giám sát liên tục và bảo mật thông tin.



Hình 2.1: Sơ đồ hệ thống

2.2.2 Lựa chọn công nghệ và nền tảng

Việc lựa chọn công nghệ và nền tảng phù hợp là yếu tố quan trọng để đảm bảo hệ thống giám sát và geofence hoạt động ổn định, hiệu quả và đáp ứng các yêu cầu kỹ thuật đã đề ra. Trong hệ thống này, các công nghệ được sử dụng sẽ bao gồm công nghệ định vị GNSS, nền tảng IoT để thu thập và quản lý dữ liệu, cũng như các giao thức truyền thông để kết nối các thành phần.

Công nghệ định vị GNSS đóng vai trò cốt lõi trong việc xác định vị trí của đối tượng. Các hệ thống định vị phổ biến hiện nay như GPS, GLONASS, Galileo và BeiDou đều có những ưu điểm riêng về độ chính xác, độ phủ sóng và khả năng hoạt động trong các điều kiện địa lý khác nhau. Dựa trên điều kiện triển khai và yêu cầu giám sát, hệ thống của tôi sử dụng công nghệ GPS vì độ phổ biến cao, khả năng định vị tốt và tương thích với nhiều thiết bị IoT hiện có.

Để đảm bảo khả năng giám sát và quản lý thiết bị từ xa, nền tảng IoT được lựa chọn là ThingsBoard. Đây là nền tảng mã nguồn mở, cung cấp nhiều tính năng mạnh mẽ như quản lý thiết bị, thu thập dữ liệu, giám sát trực quan và phát hiện sự kiện. ThingsBoard hỗ trợ giao thức truyền thông MQTT và HTTP, giúp tích hợp dễ dàng với các thiết bị IoT. Việc lựa chọn nền tảng này cũng giúp giảm chi phí phát triển và đảm bảo khả năng mở rộng trong tương lai.

Trong hệ thống giám sát, việc truyền tải dữ liệu vị trí từ thiết bị đến trung tâm

xử lý là một bước quan trọng. Để đảm bảo tính ổn định và độ trễ thấp, giao thức MQTT được sử dụng làm phương tiện truyền thông chính. MQTT là giao thức nhẹ, phù hợp cho các ứng dụng IoT có băng thông hạn chế và yêu cầu truyền tải dữ liệu liên tục. Nhờ cơ chế Publish/Subscribe, hệ thống có thể gửi và nhận thông tin một cách nhanh chóng, đảm bảo độ tin cậy trong giám sát thời gian thực.

Ngoài ra, để đảm bảo tính bảo mật trong việc truyền tải dữ liệu, hệ thống sử dụng mã hóa TLS (Transport Layer Security) để bảo vệ các gói tin trên nền tảng MQTT. Việc mã hóa này giúp đảm bảo tính toàn vẹn của dữ liệu, tránh nguy cơ bị giả mạo hoặc tấn công từ bên ngoài. Đồng thời, cơ chế xác thực hai lớp (Two-Factor Authentication) cũng được tích hợp để đảm bảo chỉ những thiết bị đã đăng ký mới có thể kết nối với hệ thống.

Đối với việc lưu trữ và xử lý dữ liệu, cơ sở dữ liệu thời gian thực TimescaleDB được lựa chọn do khả năng xử lý lượng lớn dữ liệu định vị với tốc độ cao. TimescaleDB tương thích với PostgreSQL, cung cấp khả năng phân tích dữ liệu lịch sử và giám sát hiệu năng. Điều này rất quan trọng trong việc lưu trữ lộ trình di chuyển và phát hiện các xu hướng bất thường.

Tóm lại, hệ thống giám sát GNSS và geofence được xây dựng dựa trên sự kết hợp của công nghệ GPS, nền tảng ThingsBoard và giao thức truyền thông MQTT. Bằng cách sử dụng các công nghệ hiện đại và nền tảng mạnh mẽ, hệ thống đảm bảo khả năng giám sát liên tục, bảo mật cao và dễ dàng mở rộng trong các ứng dụng thực tế.

2.2.3 Đánh giá tính khả thi

Việc đánh giá tính khả thi của hệ thống giám sát là một bước quan trọng nhằm đảm bảo hệ thống có thể đáp ứng các yêu cầu kỹ thuật cũng như phù hợp với thực tiễn triển khai. Trong quá trình đánh giá, các yếu tố như độ chính xác định vị, độ ổn định của hệ thống, khả năng mở rộng và tính bảo mật sẽ được xem xét một cách toàn diện.

Độ chính xác là yếu tố cốt lõi của mọi hệ thống giám sát GNSS. Với việc sử dụng công nghệ GPS, độ chính xác của hệ thống phụ thuộc vào nhiều yếu tố như điều kiện thời tiết, vị trí địa lý và môi trường xung quanh. Qua quá trình thử nghiệm thực tế, thiết bị GNSS cho thấy khả năng định vị với độ sai lệch từ 5 đến 10 mét trong điều kiện thông thường. Trong các môi trường đô thị dày đặc hoặc địa hình phức tạp, độ chính xác có thể giảm xuống đáng kể. Để khắc phục vấn đề này, hệ thống sử dụng các thuật toán lọc nhiễu và bù sai số như Kalman Filter, giúp cải thiện độ tin cậy của vị trí định vị.

Độ ổn định của hệ thống được đánh giá thông qua khả năng duy trì kết nối liên tục giữa thiết bị giám sát GNSS và nền tảng ThingsBoard. Các thử nghiệm cho thấy việc sử dụng giao thức MQTT giúp duy trì kết nối ổn định với độ trễ trung bình dưới 1 giây khi truyền tải dữ liệu định vị. Trong các trường hợp mất kết nối tạm thời, hệ thống tự động thực hiện cơ chế tái kết nối mà không làm gián đoạn quá trình giám sát. Điều này đảm bảo dữ liệu luôn được cập nhật kịp thời, đặc biệt quan trọng trong các tình huống yêu cầu giám sát liên tục.

Một trong những ưu điểm của hệ thống là khả năng mở rộng dễ dàng khi cần giám sát nhiều đối tượng cùng lúc. Với việc sử dụng nền tảng ThingsBoard, hệ thống có thể quản lý hàng trăm thiết bị mà không làm giảm hiệu năng. Cơ chế phân phối dữ liệu theo chủ đề (topic) của MQTT giúp việc xử lý dữ liệu từ nhiều nguồn diễn ra đồng thời mà không gây xung đột. Bên cạnh đó, cơ sở dữ liệu TimescaleDB hỗ trợ lưu trữ lượng lớn dữ liệu định vị trong thời gian dài, giúp việc phân tích lịch sử di chuyển trở nên hiệu quả.

Tính bảo mật của hệ thống được đảm bảo thông qua việc áp dụng các cơ chế mã hóa và xác thực chặt chẽ. Dữ liệu định vị từ thiết bị đến nền tảng ThingsBoard được mã hóa bằng TLS, ngăn chặn việc đánh cắp thông tin trong quá trình truyền tải. Ngoài ra, việc sử dụng xác thực hai lớp (2FA) giúp đảm bảo rằng chỉ những thiết bị được đăng ký trước mới có quyền truy cập vào hệ thống. Điều này giúp hạn chế nguy cơ bị xâm nhập hoặc giả mạo từ bên ngoài.

Thông qua các thử nghiệm ban đầu, hệ thống cho thấy khả năng giám sát liên tục với độ chính xác cao, đặc biệt là trong môi trường mở. Khả năng mở rộng và tính bảo mật được đảm bảo nhờ lựa chọn công nghệ và nền tảng phù hợp. Tuy nhiên, một số thách thức như độ trễ khi tín hiệu yếu hoặc mất kết nối vẫn cần được cải thiện.

Nhìn chung, hệ thống giám sát được thiết kế với sự chú trọng đến độ chính xác, độ ổn định, khả năng mở rộng và bảo mật. Các yếu tố này đã được thử nghiệm và cho kết quả khả quan trong điều kiện thực tế. Việc lựa chọn công nghệ hiện đại, nền tảng ThingsBoard và giao thức MQTT giúp hệ thống đạt được hiệu năng mong muốn, đồng thời đảm bảo tính bảo mật cao trong quá trình triển khai. Các kết quả đánh giá tính khả thi sẽ là nền tảng để tiến hành các bước thiết kế chi tiết và triển khai hệ thống trong thực tế.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Chương này trình bày các công nghệ, nền tảng và giao thức được sử dụng trong quá trình xây dựng hệ thống giám sát vị trí cho thiết bị nhúng NaviTracker. Việc lựa chọn công nghệ được thực hiện dựa trên yêu cầu chức năng và phi chức năng đã phân tích ở Chương 2, đảm bảo khả năng thu thập dữ liệu định vị GNSS, truyền thông ổn định tới nền tảng giám sát từ xa, và hiển thị thông tin một cách trực quan, linh hoạt.

Cụ thể, chương sẽ giới thiệu về nền tảng phần cứng của thiết bị NaviTracker, các giao thức truyền thông được tích hợp để đảm bảo trao đổi dữ liệu hiệu quả, và nền tảng ThingsBoard được sử dụng làm hệ thống hiển thị thông tin. Mỗi công nghệ được phân tích về nguyên lý hoạt động, lý do lựa chọn và vai trò của nó trong hệ thống tổng thể. Đồng thời, chương cũng sẽ so sánh một số lựa chọn thay thế có thể, qua đó làm rõ sự phù hợp của các công nghệ được sử dụng trong đồ án.

3.1 Thiết bị NaviTracker

Thiết bị NaviTracker là một nền tảng phần cứng nhúng được thiết kế chuyên dụng cho mục đích theo dõi vị trí trong thời gian thực. Thiết bị tích hợp các thành phần chính như vi điều khiển trung tâm SC20, mô-đun GNSS để thu thập dữ liệu định vị toàn cầu, và mô-đun truyền thông không dây để gửi dữ liệu đến hệ thống giám sát từ xa. Ngoài ra, thiết bị còn có bộ nhớ trong để lưu trữ tạm thời các dữ liệu trong trường hợp mất kết nối mạng.

Vi điều khiển trung tâm đóng vai trò điều phối toàn bộ hoạt động của thiết bị, bao gồm thu nhận dữ liệu GNSS, định dạng thông tin theo giao thức truyền thông phù hợp, và thực hiện các tác vụ xử lý cục bộ như tính toán quỹ đạo, phát hiện trạng thái di chuyển hoặc ngừng. Mô-đun GNSS được sử dụng có khả năng thu thập dữ liệu với độ chính xác cao và hỗ trợ các chuẩn bản tin phổ biến như NMEA hoặc UBX, giúp dễ dàng trích xuất thông tin vị trí hiện tại.

Thiết bị NaviTracker được lựa chọn nhằm đáp ứng yêu cầu theo dõi vị trí liên tục trong môi trường thực tế, nơi các yếu tố như độ ổn định kết nối, mức tiêu thụ năng lượng và kích thước vật lý đều đóng vai trò quan trọng. So với một số lựa chọn thay thế như sử dụng bo mạch phát triển chung (Arduino, Raspberry Pi), NaviTracker mang lại ưu thế về độ ổn định phần cứng, khả năng tích hợp truyền thông di động và thiết kế nhỏ gọn, phù hợp với các ứng dụng IoT triển khai thực tế ngoài hiện trường.

Việc sử dụng thiết bị này góp phần đảm bảo tính liên tục trong việc thu thập và

truyền dữ liệu định vị, phục vụ hiệu quả cho các chức năng giám sát vị trí và phân tích hành trình được trình bày tại Chương 2.

3.2 Ngôn ngữ và môi trường phát triển

Ứng dụng được phát triển cho thiết bị NaviTracker sử dụng hệ điều hành Android, với ngôn ngữ lập trình chính là Java. Đây là một lựa chọn phổ biến trong phát triển phần mềm di động và thiết bị nhúng chạy Android, nhờ vào hệ sinh thái phong phú, tính ổn định cao và khả năng truy cập trực tiếp các API phần cứng thông qua Android SDK.

Việc lựa chọn Android làm nền tảng phát triển xuất phát từ yêu cầu về khả năng xử lý linh hoạt, dễ dàng tương tác với mô-đun GNSS qua giao tiếp UART, và khả năng tích hợp các thư viện hỗ trợ giao thức MQTT để truyền dữ liệu lên hệ thống giám sát ThingsBoard. Môi trường phát triển sử dụng là Android Studio – một công cụ tích hợp hiện đại do Google phát triển, hỗ trợ đầy đủ cho việc lập trình, biên dịch, mô phỏng và gỡ lỗi ứng dụng Android.

So với các ngôn ngữ khác như C/C++ vốn thường dùng trong các hệ thống vi điều khiển truyền thống, Java trên Android mang lại lợi thế rõ rệt về tính mô-đun, khả năng mở rộng và dễ bảo trì mã nguồn. Đồng thời, hệ điều hành Android cho phép triển khai đồng thời nhiều tác vụ, như thu thập dữ liệu GNSS, lưu trữ cục bộ và truyền tải qua mạng, một cách hiệu quả nhờ vào cơ chế quản lý tiến trình và luồng xử lý riêng biệt.

Việc sử dụng Java và Android giúp rút ngắn thời gian phát triển, đồng thời đáp ứng linh hoạt các yêu cầu được phân tích trong Chương 2, đặc biệt là khả năng truyền dữ liệu định vị lên server và tương tác theo thời gian thực với các thành phần của hệ thống giám sát từ xa.

3.3 Giao thức truyền thông MQTT

MQTT (Message Queuing Telemetry Transport) là một giao thức truyền thông nhẹ, được thiết kế tối ưu cho các thiết bị có tài nguyên hạn chế và hoạt động trong môi trường mạng không ổn định. Trong đề án này, MQTT được sử dụng để truyền dữ liệu định vị và trạng thái hệ thống từ thiết bị NaviTracker lên nền tảng ThingsBoard theo thời gian thực.

Giao thức MQTT hoạt động theo mô hình publish/subscribe, trong đó thiết bị đóng vai trò publisher, gửi dữ liệu lên một máy chủ trung gian gọi là broker. ThingsBoard tích hợp sẵn MQTT broker, cho phép thiết bị đăng dữ liệu telemetry và thuộc tính thiết bị, cũng như nhận cấu hình hoặc lệnh điều khiển từ server thông qua các bản tin có cấu trúc. Việc sử dụng mô hình này giúp giảm độ trễ và tiết kiệm băng

thông, đồng thời hỗ trợ mở rộng hệ thống một cách linh hoạt.

MQTT được lựa chọn vì phù hợp với yêu cầu truyền dữ liệu theo thời gian thực đã nêu trong Chương 2, đặc biệt trong bối cảnh thiết bị có thể hoạt động ở những khu vực có chất lượng mạng không ổn định. So với các giao thức khác như HTTP hay CoAP, MQTT có độ trễ thấp hơn và mức tiêu thụ tài nguyên thấp hơn đáng kể. Đồng thời, giao thức này hỗ trợ duy trì kết nối liên tục và có khả năng gửi lại bản tin nếu kết nối bị gián đoạn, giúp đảm bảo độ tin cậy của hệ thống giám sát.

Việc tích hợp MQTT trên nền Android được thực hiện thông qua thư viện mã nguồn mở Eclipse Paho. Thư viện này cung cấp giao diện lập trình đơn giản, dễ tích hợp với các tiến trình thu thập và xử lý dữ liệu GNSS, đảm bảo cho quá trình truyền thông được diễn ra liên tục và ổn định. Qua đó, MQTT đóng vai trò là cầu nối chính giữa thiết bị và nền tảng ThingsBoard, góp phần hiện thực hóa các chức năng giám sát, cảnh báo và tương tác từ xa trong hệ thống.

3.4 Nền tảng hiển thị: ThingsBoard

ThingsBoard là một nền tảng mã nguồn mở hỗ trợ quản lý thiết bị IoT, thu thập dữ liệu từ xa, và hiển thị trực quan thông tin thông qua các dashboard cấu hình động. Trong đồ án này, ThingsBoard đóng vai trò là hệ thống giám sát trung tâm, tiếp nhận dữ liệu từ thiết bị NaviTracker thông qua giao thức MQTT, xử lý dữ liệu và cung cấp giao diện tương tác cho người dùng.

Nền tảng ThingsBoard cung cấp các thành phần chức năng chính bao gồm quản lý thiết bị (device management), lưu trữ dữ liệu thời gian thực (telemetry storage), tạo cảnh báo (rule engine) và hiển thị dữ liệu (dashboard visualization). Việc cấu hình và vận hành được thực hiện thông qua giao diện web trực quan, giúp rút ngắn thời gian triển khai hệ thống giám sát.

So với các nền tảng giám sát khác như Blynk, Node-RED hoặc Grafana, ThingsBoard có ưu điểm vượt trội về khả năng tích hợp sâu với các giao thức IoT tiêu chuẩn như MQTT, CoAP, HTTP, đồng thời hỗ trợ tạo lập các quy tắc xử lý logic ngay tại server thông qua rule chain. Điều này cho phép hệ thống phản ứng linh hoạt với các sự kiện như thiết bị vượt quá ranh giới địa lý (geofence), mất tín hiệu định vị, hoặc ngắt kết nối mạng.

Việc lựa chọn ThingsBoard nhằm đáp ứng trực tiếp các yêu cầu đã phân tích trong Chương 2 như: hiển thị tọa độ hiện tại, ghi lại lịch sử di chuyển, tính toán quãng đường di chuyển, thiết lập vùng geofence và gửi cảnh báo. Nhờ khả năng cấu hình linh hoạt và khả năng mở rộng, nền tảng này phù hợp để triển khai cả trong môi trường thử nghiệm lẫn ứng dụng thực tế ngoài hiện trường.

3.5 Các thư viện và phần mềm hỗ trợ

Trong quá trình phát triển ứng dụng giám sát vị trí cho thiết bị NaviTracker, một số thư viện và phần mềm hỗ trợ đã được sử dụng nhằm đơn giản hóa việc lập trình, nâng cao độ ổn định của hệ thống, và rút ngắn thời gian triển khai. Các thành phần này chủ yếu phục vụ cho các chức năng thu thập dữ liệu GNSS, xử lý bản tin, truyền thông qua giao thức MQTT, và hiển thị thông tin tại nền tảng ThingsBoard.

Trên phía thiết bị, thư viện Eclipse Paho MQTT Client được sử dụng để hiện thực kết nối MQTT trong môi trường Android. Thư viện này hỗ trợ đầy đủ các chức năng publish/subscribe, quản lý kết nối, và xử lý sự kiện mạng, giúp quá trình truyền dữ liệu từ thiết bị lên server diễn ra ổn định và tin cậy. Ngoài ra, các thư viện truy cập UART như android-serialport-api hỗ trợ thiết bị đọc dữ liệu GNSS trực tiếp từ cổng nối tiếp, trích xuất thông tin định vị theo chuẩn NMEA.

Về phía ThingsBoard, các công cụ tích hợp sẵn như Rule Engine, Attribute Mapping, và Widget Library hỗ trợ xử lý dữ liệu đầu vào và trực quan hóa thông tin trên dashboard. Các widget được sử dụng để hiển thị tọa độ GPS, lịch sử di chuyển, geofence, cũng như trạng thái hoạt động của thiết bị theo thời gian thực.

So với việc tự xây dựng các thành phần này từ đầu, việc sử dụng thư viện và phần mềm có sẵn giúp đảm bảo tính ổn định và tương thích với hệ thống, đồng thời phù hợp với thực tiễn triển khai các ứng dụng IoT hiện nay. Tất cả các thư viện sử dụng đều là mã nguồn mở và được cộng đồng hỗ trợ rộng rãi, đảm bảo dễ dàng bảo trì và mở rộng trong tương lai.

3.6 Tổng kết chương

Trong chương này, tôi đã trình bày các công nghệ, nền tảng và công cụ chính được sử dụng trong quá trình phát triển hệ thống giám sát vị trí NaviTracker. Cụ thể, phần cứng thiết bị NaviTracker được lựa chọn nhờ khả năng tích hợp GNSS và truyền thông di động, đáp ứng yêu cầu giám sát vị trí liên tục. Về phần mềm, ứng dụng được phát triển trên nền Android với ngôn ngữ Java, cho phép triển khai linh hoạt các chức năng như thu thập dữ liệu, truyền thông qua MQTT, và xử lý cục bộ. Giao thức MQTT được sử dụng để đảm bảo việc truyền dữ liệu nhanh, nhẹ và ổn định trong môi trường IoT. Nền tảng ThingsBoard đóng vai trò trung tâm hiển thị, lưu trữ và tương tác với thiết bị thông qua các dashboard trực quan. Cuối cùng, các thư viện và phần mềm hỗ trợ giúp đơn giản hóa việc lập trình, nâng cao độ tin cậy và khả năng mở rộng của hệ thống.

Tất cả các công nghệ trên đều được lựa chọn dựa trên yêu cầu thực tiễn của hệ thống đã phân tích ở Chương 2, đồng thời được cân nhắc kỹ lưỡng so với các lựa

chọn thay thế. Việc tích hợp các thành phần này một cách hợp lý đã tạo nền tảng kỹ thuật vững chắc cho việc hiện thực hóa hệ thống trong các chương tiếp theo.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Chương này trình bày chi tiết về thiết kế phần mềm của hệ thống giám sát GNSS tích hợp. Mục tiêu của thiết kế là đảm bảo hệ thống hoạt động ổn định, hiệu quả và có khả năng thích ứng với môi trường hoạt động của các thiết bị IoT. Nội dung chương bao gồm kiến trúc tổng thể của phần mềm, mô hình phân lớp chức năng, mô tả chi tiết từng mô-đun chính và cơ chế tương tác giữa chúng. Ngoài ra, chương cũng tập trung vào các giải pháp tối ưu hiệu suất xử lý và tiết kiệm năng lượng, nhằm nâng cao độ tin cậy và kéo dài thời gian vận hành của thiết bị trong điều kiện thực tế. Các sơ đồ và bảng minh họa được sử dụng để hỗ trợ việc diễn giải kiến trúc và luồng dữ liệu trong hệ thống.

4.1 Kiến trúc tổng thể phần mềm

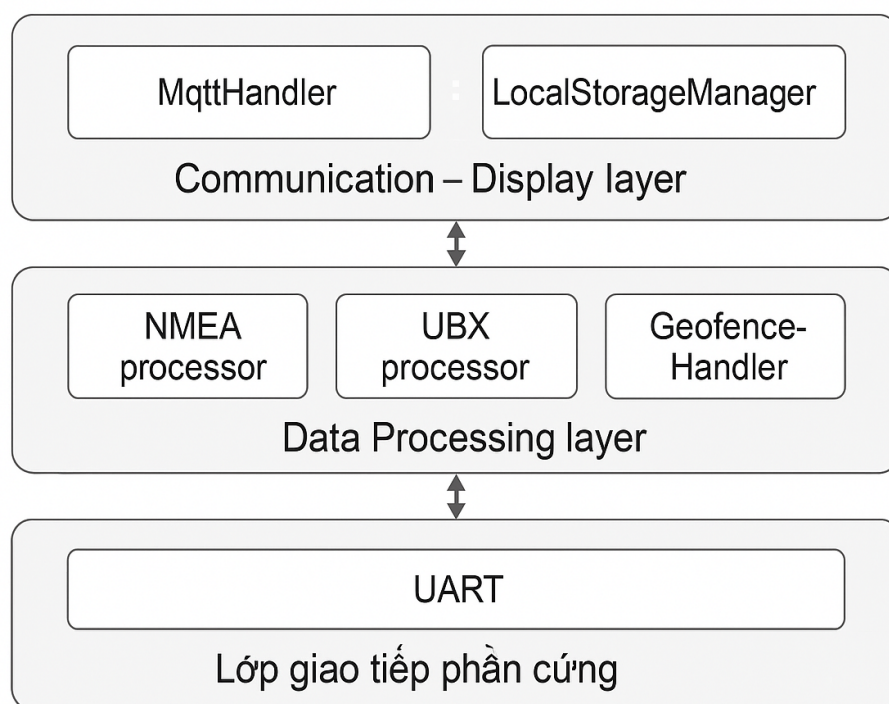
Phần mềm được thiết kế theo kiến trúc phân lớp nhằm đảm bảo tính mô-đun, dễ bảo trì và dễ mở rộng. Hệ thống được chia thành ba lớp chính: lớp giao tiếp phần cứng, lớp xử lý dữ liệu và lớp truyền thông - hiển thị.

Lớp giao tiếp phần cứng thực hiện việc thu thập dữ liệu từ thiết bị GNSS thông qua giao tiếp UART. Các byte dữ liệu được xử lý tuần tự để xây dựng các bản tin định vị hoàn chỉnh theo chuẩn NMEA và UBX.

Lớp xử lý dữ liệu chịu trách nhiệm phân tích, trích xuất và xác thực thông tin định vị từ các bản tin nhận được. Thông tin bao gồm toạ độ, tốc độ, trạng thái tín hiệu và các chỉ số liên quan đến chất lượng định vị. Lớp này cũng xử lý các thuật toán kiểm tra điểm nằm trong vùng địa lý (geofence), đồng thời lưu trữ dữ liệu vào bộ nhớ cục bộ trong trường hợp mất kết nối mạng.

Lớp truyền thông - hiển thị có chức năng gửi dữ liệu định vị và trạng thái thiết bị đến nền tảng ThingsBoard thông qua giao thức MQTT. Lớp này đồng thời tiếp nhận các cấu hình điều khiển từ máy chủ và cập nhật các thông số hoạt động tương ứng. Dữ liệu cũng được hiển thị trực tiếp trên giao diện ứng dụng người dùng để phục vụ giám sát thời gian thực.

Kiến trúc này cho phép phân tách rõ ràng giữa các chức năng, giảm thiểu sự phụ thuộc giữa các mô-đun, đồng thời hỗ trợ triển khai các cơ chế tối ưu tài nguyên và năng lượng phù hợp với đặc thù của thiết bị nhúng.

**Hình 4.1:** Kiến trúc tổng thể phần mềm

4.2 Thiết kế phân lớp phần mềm

4.2.1 Lớp giao tiếp phần cứng

Lớp giao tiếp phần cứng là lớp nền trong toàn bộ kiến trúc phần mềm, chịu trách nhiệm kết nối với thiết bị GNSS và thu nhận dữ liệu thông qua giao tiếp UART. Thành phần trung tâm trong lớp này là lớp `UartReader`, được xây dựng để thực hiện các thao tác mở cổng, đọc dữ liệu byte theo thời gian thực và gửi dữ liệu điều khiển xuống thiết bị.

Ngay khi thiết bị khởi động, ứng dụng sử dụng `UartReader` để mở cổng UART tại đường dẫn cấu hình sẵn (`/dev/ttyHSL0`). Nếu mở cổng thành công, một luồng xử lý riêng sẽ được kích hoạt để liên tục đọc từng byte từ thiết bị GNSS. Mỗi byte nhận được sẽ được chuyển vào chuỗi xử lý kế tiếp thông qua các lớp phân tích dữ liệu.

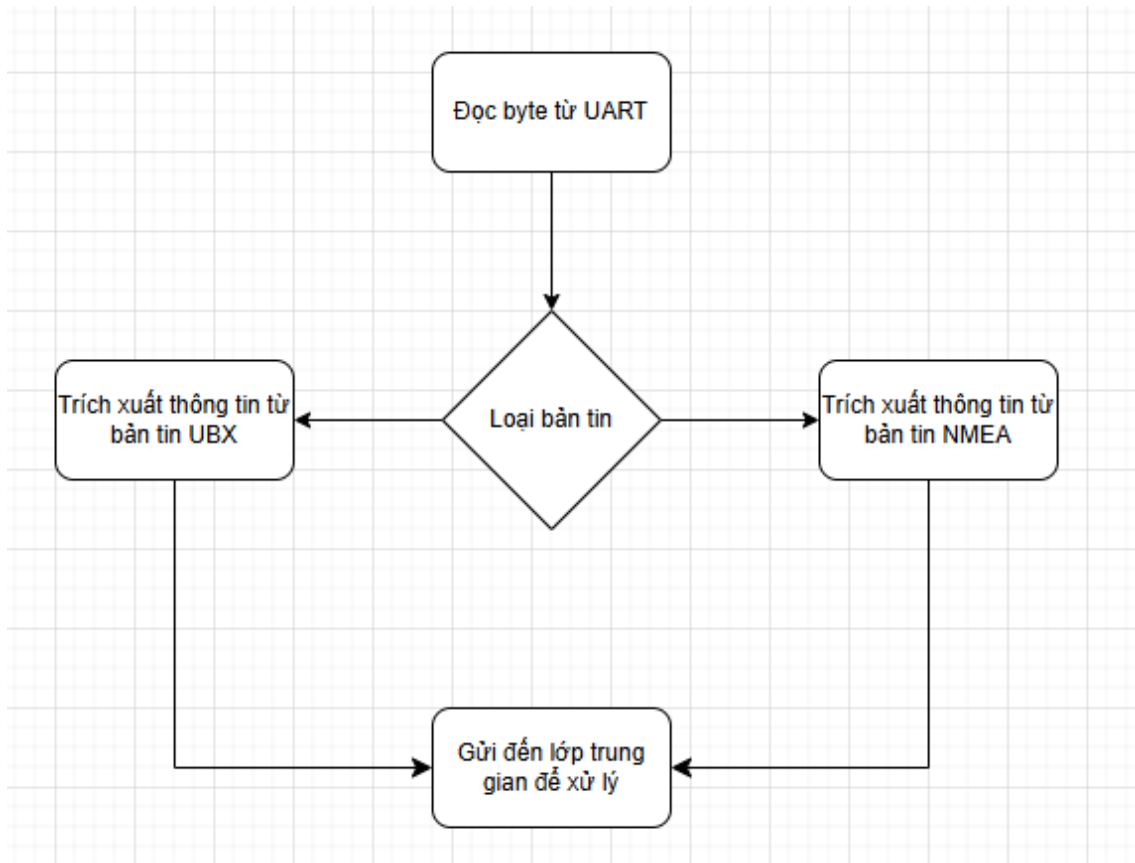
Luồng xử lý dữ liệu sau khi nhận byte từ UART được chia thành hai nhánh chính, tương ứng với hai định dạng bản tin phổ biến của thiết bị GNSS là NMEA và UBX. Nếu dữ liệu là bản tin NMEA, tức có ký tự bắt đầu là dấu `$`, luồng byte sẽ được chuyển đến lớp `NMEAProcessor` để tái tạo thành một chuỗi hoàn chỉnh. Sau đó, chuỗi này được phân tích bởi lớp `NMEAHandler` nhằm trích xuất thông tin định vị như toạ độ, tốc độ di chuyển, hướng đi và tính hợp lệ của tín hiệu.

Ngược lại, nếu byte đầu là cặp `0xB5 0x62` thì dữ liệu thuộc định dạng UBX.

Lúc này, nó sẽ được truyền đến các bộ xử lý như `GNSSProcessor`, `NavStatusProcessor` hoặc `SFRBXProcessor` để kiểm tra độ dài, xác thực checksum và phân tích phần nội dung payload. Các bộ xử lý này hỗ trợ việc trích xuất thông tin cấu hình GNSS hoặc trạng thái vệ tinh đang hoạt động.

Dữ liệu định vị hợp lệ sau khi được trích xuất sẽ được chuyển tiếp lên lớp xử lý trung gian để tính trung bình tọa độ, phát hiện ra các vi phạm geofence hoặc gửi thông tin đến nền tảng ThingsBoard thông qua giao thức MQTT. Bên cạnh chức năng đọc, lớp giao tiếp phần cứng còn cho phép gửi dữ liệu điều khiển xuống thiết bị GNSS. Chẳng hạn, lớp `GNSSControl` có thể tạo bản tin `UBX-CFG-GNSS` để bật hoặc tắt các hệ thống vệ tinh như GPS, GLONASS, Galileo hoặc BeiDou. Bản tin sau đó được gửi đi thông qua UART nhờ phương thức ghi của `UartReader`.

Luồng xử lý UART đảm bảo rằng dữ liệu GNSS được tiếp nhận liên tục và chính xác, bất kể trong điều kiện hoạt động bình thường hay gián đoạn mạng. Đây là bước đầu tiên và cũng là tiền đề quan trọng để hệ thống thực hiện các tác vụ giám sát, phân tích và truyền thông một cách ổn định, đáng tin cậy.



Hình 4.2: Luồng xử lý dữ liệu lớp phần cứng

4.2.2 Lớp xử lý dữ liệu

Lớp xử lý dữ liệu đóng vai trò trung gian trong hệ thống, đảm nhận nhiệm vụ tiếp nhận dữ liệu thô từ lớp giao tiếp phần cứng, sau đó phân tích, trích xuất và xử lý thông tin định vị cần thiết. Lớp này được cấu trúc thành nhiều mô-đun chức năng, mỗi mô-đun đảm nhận một nhiệm vụ chuyên biệt tương ứng với các loại bản tin định vị khác nhau.

4.2.2.1 Các bản tin từ module GNSS ublox NEO-M9

Module GNSS u-blox NEO-M9 hỗ trợ truyền dữ liệu định vị qua hai định dạng bản tin chính là NMEA và UBX. Bản tin NMEA được sử dụng phổ biến để cung cấp thông tin vị trí dưới dạng chuỗi ASCII, trong khi bản tin UBX là định dạng nhị phân độc quyền của u-blox, cho phép truyền thông tin cấu hình và trạng thái với độ chính xác và linh hoạt cao hơn. Hệ thống phần mềm sử dụng bốn bản tin chính: NMEA-RMC, UBX-CFG-GNSS, UBX-NAV-STATUS và UBX-RXM-SFRBX, với chức năng và cấu trúc như sau:

Bản tin RMC (Recommended Minimum Specific GNSS Data) cung cấp thông tin tối thiểu cần thiết về định vị như thời gian, vị trí, tốc độ và hướng. Chuỗi bắt đầu bằng \$GxRMC và kết thúc bằng checksum. Các trường trong bản tin có định dạng văn bản, phân tách bằng dấu phẩy.

Field	Tên trường	Định dạng	Đơn vị	Ví dụ	Mô tả
0	xxRMC	Chuỗi	-	GPRMC	Mã tin nhắn RMC
1	Thời gian	hhmmss.sss	-	083559.00	Thời gian UTC
2	Trạng thái	Ký tự	-	A	Trạng thái tín hiệu (A = hợp lệ)
3	Vĩ độ	ddmm.mmmm	Độ	4717.11437	Vĩ độ
4	NS	Ký tự	-	N	Chỉ báo Bắc - Nam
5	Kinh độ	dddmm.mmmm	Độ	00833.91522	Kinh độ
6	EW	Ký tự	-	E	Chỉ báo Đông - Tây
7	Tốc độ	float	knots	22.4	Tốc độ mặt đất
8	Hướng đi	float	Độ	84.4	Hướng chuyển động
9	Ngày	ddmmyy	-	230394	Ngày tháng năm
10	mv	Số	Độ	-	Biến thiên từ trường
11	mvEW	Ký tự	-	-	Biến thiên từ trường Đông - Tây
12	posMode	Ký tự	-	A	Chỉ báo chế độ
13	navStatus	Ký tự	-	V	Trạng thái định vị

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Field	Tên trường	Định dạng	Đơn vị	Ví dụ	Mô tả
14	cs	Thập lục phân	-	57	Checksum
15	CRLF	Ký tự	-	-	Ký tự xuống dòng

Bảng 4.1: Cấu trúc bản tin NMEA-RMC

Bản tin UBX-CFG-GNSS được sử dụng để cấu hình hệ thống GNSS đang hoạt động trên thiết bị. Bản tin chứa thông tin về số lượng kênh theo phần cứng và phần mềm, số hệ thống GNSS được cấu hình và trạng thái bật/tắt của từng hệ thống.

Offset	Tên trường	Định dạng	Đơn vị	Ví dụ	Mô tả
0	msgVer	uint8	-	0	Phiên bản bản tin
1	numTrkChHw	uint8	kênh	32	Số kênh phần cứng hỗ trợ
2	numTrkChUse	uint8	kênh	32	Số kênh GNSS được sử dụng
3	numConfigBlocks	uint8	-	6	Số hệ thống GNSS được cấu hình
4	repeated blocks	struct	-	...	Dữ liệu cấu hình cho từng hệ thống GNSS (ID, số kênh, cờ enable, flags, v.v.)

Bảng 4.2: Cấu trúc bản tin UBX-CFG-GNSS

Bản tin UBX-NAV-STATUS cung cấp thông tin về trạng thái định vị hiện tại của thiết bị, bao gồm kiểu fix, các cờ trạng thái, và thông tin xác thực tín hiệu.

Offset	Tên trường	Định dạng	Đơn vị	Ví dụ	Mô tả
0	iTOW	uint32	ms	501867000	Thời gian GPS (Time of Week)
4	gpsFix	uint8	-	3	Trạng thái định vị (3 = 3D fix)
5	flags	bitfield	-	0x0D	Các cờ phản ánh chất lượng và độ tin cậy tín hiệu
6	fixStat	bitfield	-	0x00	Trạng thái xác định vị trí
7	flags2	bitfield	-	0x00	Các cờ bổ sung

Bảng 4.3: Cấu trúc bản tin UBX-NAV-STATUS

Bản tin UBX-RXM-SFRBX chứa dữ liệu thô từ tín hiệu vệ tinh, thường được sử dụng cho mục đích phân tích tín hiệu ở lớp thấp. Bản tin cung cấp thông tin như ID vệ tinh, tần số và các từ dữ liệu 32 bit từ mỗi vệ tinh.

Offset	Tên trường	Định dạng	Đơn vị	Ví dụ	Mô tả
0	gnssId	uint8	-	0	ID hệ thống GNSS (0 = GPS)
1	svId	uint8	-	3	ID vệ tinh
2	reserved1	uint8	-	0	Trường dự phòng
3	freqId	uint8	-	0	ID tần số tín hiệu
4	numWords	uint8	-	10	Số lượng từ dữ liệu GNSS
8	dwrđ	uint32[10]	-	...	Dữ liệu 32 bit từ tín hiệu vệ tinh

Bảng 4.4: Cấu trúc bản tin UBX-RXM-SFRBX

4.2.2.2 Luồng xử lý dữ liệu

Lớp xử lý dữ liệu đóng vai trò trung gian trong hệ thống, đảm nhận nhiệm vụ tiếp nhận dữ liệu thô từ lớp giao tiếp phần cứng, sau đó phân tích, trích xuất và xử lý thông tin định vị cần thiết. Lớp này được cấu trúc thành nhiều mô-đun chức năng, mỗi mô-đun đảm nhận một nhiệm vụ chuyên biệt tương ứng với các loại bản tin định vị khác nhau.

Đối với dữ liệu NMEA, sau khi được tái dựng thành chuỗi hoàn chỉnh từ `NMEAProcessor`, chuỗi này sẽ được truyền vào lớp `NMEAHandler` để trích xuất các thông tin như vĩ độ, kinh độ, tốc độ di chuyển và trạng thái tín hiệu. Việc xử lý các câu lệnh GGA, GNS, RMC, hoặc VTG được thực hiện thông qua các parser chuyên biệt, giúp đảm bảo khả năng trích xuất chính xác từ đa nguồn GNSS.

Trong trường hợp bản tin có định dạng UBX, hệ thống sử dụng các mô-đun chuyên biệt như `GNSSProcessor`, `NavStatusProcessor` và `SFRBXProcessor`. Các mô-đun này thực hiện việc nhận diện chuỗi bản tin, kiểm tra độ dài hợp lệ, xác thực checksum và trích xuất thông tin cấu hình hoặc trạng thái vệ tinh. Thông tin này phục vụ cho việc xác định số lượng hệ thống GNSS đang hoạt động, theo dõi chất lượng tín hiệu hoặc tái cấu hình hoạt động GNSS thông qua các bản tin điều khiển.

Sau khi dữ liệu được phân tích và xác thực, lớp xử lý tiếp tục thực hiện các thao tác cao hơn như tính tốc độ di chuyển dựa trên thời gian thực và khoảng cách giữa các toạ độ, xác định thay đổi vị trí vượt quá ngưỡng cấu hình, hoặc phát hiện sự thay đổi trạng thái hoạt động của GNSS. Các dữ liệu hợp lệ được truyền đến lớp truyền thông để hiển thị hoặc đồng bộ với nền tảng giám sát.

Một chức năng quan trọng khác của lớp xử lý dữ liệu là hỗ trợ cơ chế geofence. Toạ độ được phân tích sẽ được kiểm tra thông qua lớp `GeofenceHandler`, sử dụng thuật toán kiểm tra điểm nằm trong đa giác để xác định xem thiết bị có nằm ngoài vùng giới hạn được cấu hình trước hay không. Kết quả này được sử dụng để gửi cảnh báo đến hệ thống khi có vi phạm ranh giới.

Ngoài ra, để đảm bảo hoạt động ổn định trong môi trường mạng không liên tục, các bản ghi dữ liệu định vị sẽ được lưu tạm thời bằng lớp `LocalStorageManager`. Khi mạng ổn định trở lại, dữ liệu này sẽ được đồng bộ qua lớp truyền thông MQTT. Cơ chế này giúp tránh mất dữ liệu trong quá trình thiết bị di chuyển qua các khu vực có chất lượng kết nối kém.

Toàn bộ lớp xử lý dữ liệu được thiết kế theo hướng mở rộng và tương thích với nhiều loại bản tin GNSS khác nhau, đảm bảo hiệu quả phân tích và tính linh hoạt trong các tình huống vận hành thực tế.

4.2.3 Lớp truyền thông

Lớp truyền thông có vai trò kết nối giữa thiết bị và nền tảng ThingsBoard thông qua giao thức MQTT. Thành phần trung tâm của lớp này là lớp `MqttHandler`, chịu trách nhiệm khởi tạo kết nối, gửi dữ liệu định vị, gửi các thông số trạng thái hệ thống và tiếp nhận các cấu hình điều khiển từ xa.

Ngay khi khởi động, `MqttHandler` tiến hành kết nối đến máy chủ MQTT bằng định danh thiết bị ThingsBoard. Sau khi kết nối thành công, thiết bị sẽ tự động gửi một bản tin `init` chứa các thuộc tính cấu hình hiện tại của thiết bị.

Chương trình sử dụng hai topic chính: `telemetryTopic` để gửi dữ liệu thời gian thực như tọa độ và tốc độ di chuyển và `attributeTopic` để gửi hoặc nhận các thuộc tính cấu hình hệ thống như trạng thái pin, geofence hoặc điều kiện hoạt động. Tùy theo chế độ định vị, hệ thống sử dụng hai chế độ gửi dữ liệu:

- `DEVICE_LOCATION`: gửi tọa độ từ dịch vụ định vị của Android.
- `UBLOX_LOCATION`: gửi tọa độ GNSS trung bình từ module u-blox.

Để tối ưu hiệu suất truyền thông và tiết kiệm năng lượng, lớp này chỉ gửi dữ liệu khi thiết bị có kết nối mạng hoặc khi tọa độ thay đổi vượt qua ngưỡng cấu hình. Nếu mất mạng, dữ liệu sẽ được lưu tạm thời trong bộ nhớ cục bộ thông qua lớp `LocalStorageManager` và tự động đồng bộ lại khi mạng ổn định trở lại.

Ngoài dữ liệu định vị, `MqttHandler` cũng xử lý các bản tin phản hồi từ ThingsBoard. Các bản tin thuộc chủ đề thuộc tính (`attribute`) được phân tích và cập nhật trực tiếp vào các thông số hoạt động như khoảng cách tối đa (`maxDistance`), thời gian tối đa (`maxTimeout`), tên tỉnh giám sát hoặc trạng thái bật/tắt vùng geofence. Nếu geofence được bật, ranh giới địa lý tương ứng với tỉnh được lấy từ file `provinces.json` và gửi ngược lên server để hiển thị trên dashboard.

Cuối cùng, lớp này còn hỗ trợ gửi các thuộc tính trạng thái như mức pin, trạng thái sạc, trạng thái GPS và cờ xác định thiết bị có nằm ngoài khu vực địa lý không (`isOutside`). Toàn bộ lớp được thiết kế bất đồng bộ, với cơ chế tự động kết nối

lại khi xảy ra gián đoạn để đảm bảo tính liên tục trong quá trình truyền nhận dữ liệu.

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Trong hệ thống giám sát vị trí được xây dựng, server đóng vai trò là trung tâm tiếp nhận, xử lý và hiển thị dữ liệu định vị từ thiết bị. Thông tin thu thập từ thiết bị bao gồm tọa độ GPS, tốc độ di chuyển, trạng thái hệ thống và các thông số cấu hình liên quan. Những dữ liệu này được truyền về server theo thời gian thực thông qua giao thức MQTT và được lưu trữ để phục vụ việc giám sát, phân tích và truy xuất sau này.

Nền tảng ThingsBoard được lựa chọn làm giải pháp triển khai server do khả năng hỗ trợ trực quan hóa dữ liệu IoT một cách hiệu quả và dễ cấu hình. Server không chỉ có chức năng hiển thị vị trí hiện tại trên bản đồ, mà còn lưu trữ toàn bộ lịch sử di chuyển, hiển thị trạng thái hoạt động của thiết bị và phát hiện các tình huống vượt ra khỏi vùng giới hạn địa lý (geofence).

Ngoài ra, hệ thống server còn cho phép người dùng cấu hình các thông số từ xa như thời gian gửi dữ liệu, khoảng cách thay đổi tối thiểu để cập nhật vị trí, bật hoặc tắt tính năng geofence và chọn khu vực cần giám sát. Những điều chỉnh này được cập nhật động và gửi ngược trở lại thiết bị thông qua bản tin attributes.

Toàn bộ các chức năng trên được xây dựng nhằm đảm bảo hệ thống hoạt động liên tục, linh hoạt, đồng thời cung cấp giao diện thân thiện cho việc giám sát và quản lý thiết bị từ xa.

5.1 Cài đặt và cấu hình Thingsboard

ThingsBoard là một nền tảng mã nguồn mở chuyên dụng cho các ứng dụng IoT, hỗ trợ thu thập dữ liệu, xử lý, lưu trữ và trực quan hóa thông tin từ các thiết bị đầu cuối. Trong đồ án này, ThingsBoard được sử dụng làm server để tiếp nhận và hiển thị dữ liệu định vị từ thiết bị giám sát.

Server được cài đặt trên máy tính cá nhân chạy hệ điều hành Ubuntu, sử dụng bản Community Edition của ThingsBoard. Việc cài đặt được thực hiện thông qua tập tin .deb. Sau khi cài đặt thành công, ThingsBoard hoạt động như một dịch vụ chạy nền và có thể truy cập thông qua trình duyệt tại địa chỉ mặc định là <http://localhost:8080>.

Quá trình cấu hình ban đầu bao gồm việc khởi tạo tài khoản quản trị viên, tạo thiết bị mới (device), và gán Access Token cho thiết bị. Token này được sử dụng để xác thực mỗi khi thiết bị gửi dữ liệu lên server qua giao thức MQTT.

Để hệ thống hiển thị dữ liệu đúng định dạng, các trường dữ liệu (như latitude, longitude, speed, gps_status, v.v.) được định nghĩa trong widget dashboard. (Cụ thể, ccrulechaincn)

Việc cài đặt và cấu hình ThingsBoard đảm bảo hệ thống server có thể hoạt động ổn định, tiếp nhận dữ liệu theo thời gian thực, và phục vụ cho các chức năng giám sát, hiển thị và điều khiển từ xa. Đây là bước nền tảng quan trọng để xây dựng toàn bộ kiến trúc hệ thống giám sát vị trí.

5.2 Cấu hình NAT để truy cập từ Internet

Mặc định, hệ thống ThingsBoard chỉ có thể truy cập được từ bên trong mạng nội bộ (LAN) nơi server được cài đặt. Tuy nhiên, để thiết bị giám sát có thể gửi dữ liệu từ bất kỳ vị trí nào trên Internet về server, cần thực hiện cấu hình NAT (Network Address Translation) và chuyển tiếp cổng (port forwarding) trên router.

Trước hết, địa chỉ IP nội bộ của máy chủ chạy ThingsBoard cần được gán cố định để tránh thay đổi mỗi khi khởi động lại hệ thống. Việc gán IP tĩnh đã được router tự động thực hiện dựa trên địa chỉ MAC của máy server.

Tiếp theo, cần truy cập vào giao diện cấu hình của router để thực hiện chuyển tiếp các cổng dịch vụ từ bên ngoài vào địa chỉ IP nội bộ của máy chủ. Cụ thể đã mở các cổng sau:

Cổng 8080/TCP: dùng cho giao diện quản trị ThingsBoard qua trình duyệt.

Cổng 1883/TCP: dùng cho giao tiếp MQTT giữa thiết bị và server.

Việc cấu hình NAT và mở cổng thành công cho phép hệ thống giám sát hoạt động liên tục và ổn định từ mọi vị trí có kết nối Internet. Đây là bước quan trọng để hệ thống có thể triển khai thực tế và phục vụ nhu cầu giám sát từ xa.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Sinh viên so sánh kết quả nghiên cứu hoặc sản phẩm của mình với các nghiên cứu hoặc sản phẩm tương tự.

Sinh viên phân tích trong suốt quá trình thực hiện ĐATN, mình đã làm được gì, chưa làm được gì, các đóng góp nổi bật là gì, và tổng hợp những bài học kinh nghiệm rút ra nếu có.

6.2 Hướng phát triển

Trong phần này, sinh viên trình bày định hướng công việc trong tương lai để hoàn thiện sản phẩm hoặc nghiên cứu của mình.

Trước tiên, sinh viên trình bày các công việc cần thiết để hoàn thiện các chức năng/nhiệm vụ đã làm. Sau đó sinh viên phân tích các hướng đi mới cho phép cải thiện và nâng cấp các chức năng/nhiệm vụ đã làm.

MỘT SỐ LƯU Ý VỀ TÀI LIỆU THAM KHẢO

Bibliography

- [1] u-blox AG. *u-blox M9 Receiver Description – Protocol Specification*. Document UBX-13003221, Revision 1.32. u-blox AG, Switzerland, 2021.
https://content.u-blox.com/sites/default/files/u-blox-M9-SPG-4.04_InterfaceDescription_UBX-21022436.pdf

PHỤ LỤC

A. HƯỚNG DẪN VIẾT ĐỒ ÁN TỐT NGHIỆP

Quy định chung

Dưới đây là một số quy định và hướng dẫn viết đồ án tốt nghiệp mà bắt buộc sinh viên phải đọc kỹ và tuân thủ nghiêm ngặt.

Sinh viên cần đảm bảo tính thống nhất toàn báo cáo (font chữ, căn dòng hai bên, hình ảnh, bảng, margin trang, đánh số trang, v.v.). Để làm được như vậy, sinh viên chỉ cần sử dụng các định dạng theo đúng template ĐATN này. Khi paste nội dung văn bản từ tài liệu khác của mình, sinh viên cần chọn kiểu Copy là “Text Only” để định dạng văn bản của template không bị phá vỡ/vi phạm.

Tuyệt đối cấm sinh viên đạo văn. Sinh viên cần ghi rõ nguồn cho tất cả những gì không tự mình viết/vẽ lên, bao gồm các câu trích dẫn, các hình ảnh, bảng biểu, v.v. Khi bị phát hiện, sinh viên sẽ không được phép bảo vệ ĐATN.

Tất cả các hình vẽ, bảng biểu, công thức, và tài liệu tham khảo trong ĐATN nhất thiết phải được SV giải thích và tham chiếu tới ít nhất một lần. Không chấp nhận các trường hợp sinh viên đưa ra hình ảnh, bảng biểu tùy hứng và không có lời mô tả/giải thích nào.

Sinh viên tuyệt đối không trình bày ĐATN theo kiểu viết ý hoặc gạch đầu dòng. ĐATN không phải là một slide thuyết trình; khi người đọc không hiểu sẽ không có ai giải thích hộ. Sinh viên cần viết thành các đoạn văn và phân tích, diễn giải đầy đủ, rõ ràng. Câu văn cần đúng ngữ pháp, đầy đủ chủ ngữ, vị ngữ và các thành phần câu. Khi thực sự cần liệt kê, sinh viên nên liệt kê theo phong cách khoa học với các ký tự La Mã. Ví dụ, nhiều sinh viên luôn cảm thấy hối hận vì (i) chưa cố gắng hết mình, (ii) chưa sắp xếp thời gian học/choi một cách hợp lý, (iii) chưa tìm được người yêu để chia sẻ quãng đời sinh viên vất vả, và (iv) viết ĐATN một cách cầu thả.

Trong một số trường hợp nhất thiết phải dùng các bullet để liệt kê, sinh viên cần thống nhất Style cho toàn bộ các bullet các cấp mà mình sử dụng đến trong báo cáo. Nếu dùng bullet cấp 1 là hình tròn đen, toàn bộ báo cáo cần thống nhất cách dùng như vậy; ví dụ như sau:

- Đây là mục 1 – Thực sự không còn cách nào khác tôi mới dùng đến việc bullet trong báo cáo.
- Đây là mục 2 – Nghĩ lại thì tôi có thể không cần dùng bullet cũng được. Nên tôi sẽ xóa bullet và tổ chức lại hai mục này trong báo cáo của mình cho khoa học hơn. Tôi muốn thầy cô và người đọc cảm nhận được tâm huyết của tôi

trong từng trang báo cáo ĐATN.

A.1 Ngành học

Sinh viên lưu ý viết đúng ngành/chuyên ngành trên bìa và trên gáy theo đúng quy định của Trường. Ngành học hay chuyên ngành học phụ thuộc vào ngành học mà sinh viên đăng ký. Sinh viên có thể đăng nhập trên trang quản lý học tập của mình để xem lại chính xác ngành học của mình.

Một số ví dụ sinh viên có thể tham khảo dưới đây, trong trường hợp có chuyên ngành thì sinh viên không cần ghi chuyên ngành:

Đối với kỹ sư chính quy: Từ K61 trở về trước: Ngành Kỹ thuật phần mềm Từ K62 trở về sau: Ngành Khoa học máy tính Đối với cử nhân: Ngành Công nghệ thông tin Đối với chương trình EliteTech: Chương trình Việt Nhật/KSTN: Ngành Công nghệ thông tin Chương trình ICT Global: Ngành Information Technology Chương trình DS&AI: Ngành Khoa học dữ liệu

A.2 Đánh dấu (bullet) và đánh số (numering)

Việc sử dụng danh sách trong LaTeX khá đơn giản và không yêu cầu sinh viên phải thêm bất kỳ gói bổ sung nào. LaTeX cung cấp hai môi trường liệt kê đó là:

- Đánh dấu (bullet) là kiểu liệt kê không có thứ tự. Để sử dụng kiểu liệt kê đánh dấu, chúng ta khai báo như sau

```
\begin{itemize}
\item Nội dung thứ nhất được viết ở đây.
\item Nội dung thứ hai được viết ở đây.
\item ...
\end{itemize}
```

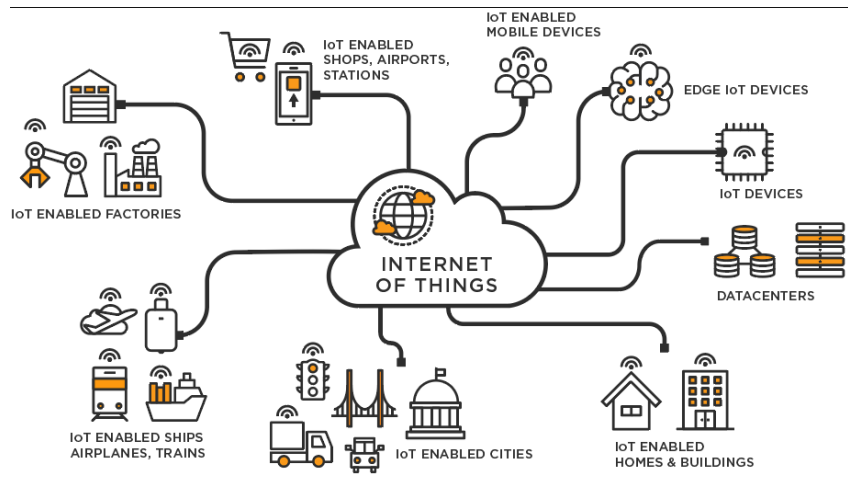
- Đánh số (numering) là kiểu liệt kê có thứ tự. Để sử dụng kiểu liệt kê đánh số, chúng ta khai báo như sau

```
\begin{enumerate}
\item Nội dung thứ nhất được viết ở đây.
\item Nội dung thứ hai được viết ở đây.
\item ...
\end{enumerate}
```

Chú ý các nội dung trình bày trong cả hai môi trường liệt kê theo sau lệnh `\item`. Ngoài ra LaTeX còn cung cấp một số kiểu liệt kê khác, sinh viên có thể tham khảo tại <https://www.overleaf.com/learn/latex/Lists>

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Bảng A.1: Table to test captions and labels.



Hình A.1: Internet vạn vật

A.3 Cách thêm bảng

Bảng A.1 là ví dụ về cách tạo bảng. Tất cả các bảng biểu phải được đề cập đến trong phần nội dung và phải được phân tích và bình luận. Chú ý: Tạo bảng trong Latex khá phức tạp và mất thời gian, vì vậy sinh viên có thể sử dụng các công cụ hỗ trợ tạo bảng (Ví dụ: <https://www.tablesgenerator.com/>). Sinh viên có thể tìm hiểu sâu hơn về cách chèn ảnh trong Latex tại link <https://www.overleaf.com/learn/latex/Tables>.

A.4 Chèn hình ảnh

Hình A.1 là ví dụ về cách chèn ảnh. Lưu ý chú thích của hình vẽ được đặt ngay dưới hình vẽ. Sinh viên có thể tìm hiểu sâu hơn về cách chèn ảnh trong Latex tại https://www.overleaf.com/learn/latex/Inserting_Images.

Chú ý, tất cả các hình vẽ phải được đề cập đến trong phần nội dung và phải được phân tích và bình luận.

A.5 Tài liệu tham khảo

A.5.0.0.1 Cách liệt kê

Áp dụng cách liệt kê theo quy định của IEEE. Ví dụ của việc trích dẫn như sau

scott2013sdn. Cụ thể, sinh viên sử dụng lệnh `\cite{}` như sau **ashton2009internet**. Chỉ những tài liệu được trích dẫn thì mới xuất hiện trong phần Tài liệu tham khảo. Tài liệu tham khảo cần có nguồn gốc rõ ràng và phải từ nguồn đáng tin cậy. Hạn chế trích dẫn tài liệu tham khảo từ các website, từ wikipedia.

A.5.0.0.2 Các loại tài liệu tham khảo

Các nguồn tài liệu tham khảo chính là sách, bài báo trong các tạp chí, bài báo trong các hội nghị khoa học và các tài liệu tham khảo khác trên internet.

A.6 Cách viết phương trình và công thức toán học

Các gói `amsmath`, `amssymb`, `amsfonts` hỗ trợ viết phương trình/công thức toán học đã được bổ sung sẵn ở phần đầu của file `main.tex`. Một ví dụ về tạo phương trình (A.1) như sau

$$F(x) = \int_b^a \frac{1}{3}x^3 \quad (\text{A.1})$$

Phương trình A.1 là ví dụ về phương trình tích phân. Một phương trình khác không được đánh số thứ tự (gán nhãn)

$$x[t_n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[f_k] e^{j2\pi nk/N}$$

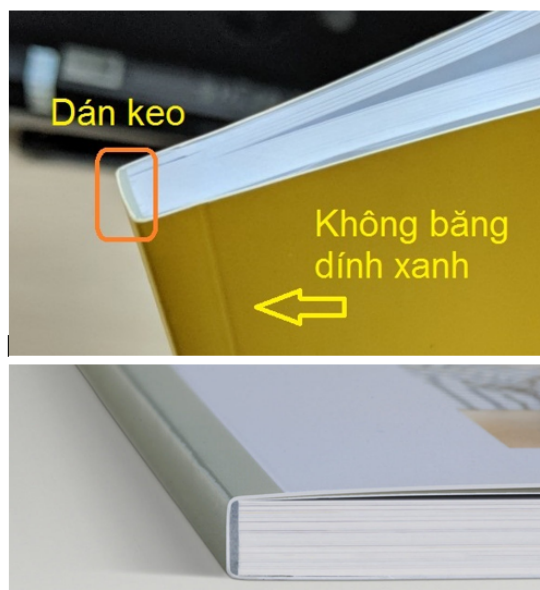
Phương trình này thể hiện phép biến đổi Fourier rời rạc ngược (IDFT).

A.7 Qui cách đóng quyển

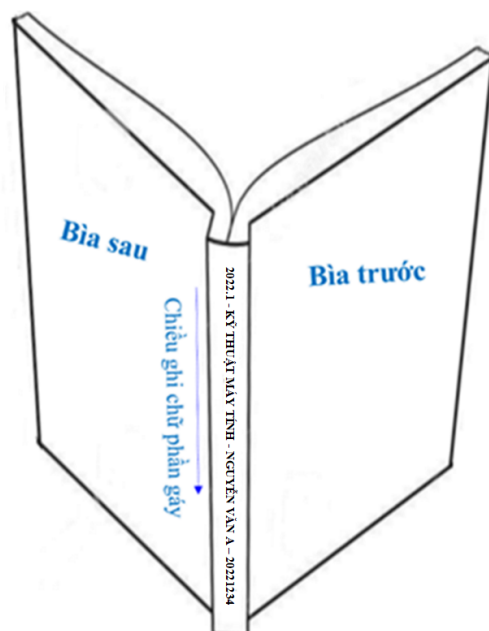
Phần bìa trước chế bản theo qui định; bìa trước và bìa sau là giấy liền khổ. Sử dụng keo nhiệt để dán gáy khi đóng quyển thay vì sử dụng băng dính và dập ghim như mô tả ở Hình A.3 Phần gáy ĐATN cần ghi các thông tin tóm tắt sau: Kỳ làm ĐATN - Ngành đào tạo - Họ và tên sinh viên - Mã số sinh viên. Ví dụ:

2022.1 - KỸ THUẬT MÁY TÍNH - NGUYỄN VĂN A - 20221234

Qui cách ghi chữ phần gáy như hình dưới đây:



Hình A.2: Quy cách đóng quyển đồ án



Hình A.3: Quy cách đóng quyển đồ án

B. ĐẶC TẢ USE CASE

Nếu trong nội dung chính không đủ không gian cho các use case khác (ngoài các use case nghiệp vụ chính) thì đặc tả thêm cho các use case đó ở đây.

B.1 Đặc tả use case “Thống kê tình hình mượn sách”

...

B.2 Đặc tả use case “Đăng ký làm thẻ mượn”

...