|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ĐẠI HỌC BÁCH KHOA HÀ NỘI**  **TRƯỜNG ĐIỆN – ĐIỆN TỬ**  logo_128  BÁO CÁO ĐỒ ÁN TỐT NGHIỆP CỬ NHÂN  **Tên đề tài:**  **..........**  Giảng viên hướng dẫn: PGS. TS. NGUYỄN ĐỨC MINH  TS. HOÀNG PHƯƠNG CHI  Nhóm sinh viên thực hiện:   |  |  |  |  | | --- | --- | --- | --- | | STT | Họ và tên sinh viên | MSSV | Lớp | | 1 | Lê Thành Luân | 20203494 | ET1 08 K65 | | 2 | Hà Văn Đức | 20203677 | ET1 08 K65 | | 3 | Trần Minh Đức | 20203370 | ET1 06 K65 | | 4 | Phạm Thị Xuân Mai | 20203734 | ET1 06 K65 | | 5 | Phạm Hồng Thi | 20203767 | ET1 02 K65 |   Trường/ Khoa/ Viện: Trường Điện - Điện tử  Số điện thoại: 0334997344  Hà Nội, 5-2024 |

**MỤC LỤC**

[DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT i](#_Toc166440030)

[DANH MỤC HÌNH VẼ ii](#_Toc166440031)

[DANH MỤC BẢNG BIỂU iii](#_Toc166440032)

[TÓM TẮT iv](#_Toc166440033)

[CHƯƠNG 1. ĐẶT VẤN ĐỀ 5](#_Toc166440034)

[CHƯƠNG 2. PHƯƠNG PHÁP NGHIÊN CỨU 7](#_Toc166440035)

[2.1 Cơ sở lý thuyết 7](#_Toc166440036)

[2.1.1 Giao thức MQTT 7](#_Toc166440037)

[2.1.2 Mạng 2G 8](#_Toc166440038)

[2.1.3 Cơ chế tiết kiệm năng lượng 8](#_Toc166440039)

[2.2 Yêu cầu hệ thống 12](#_Toc166440040)

[2.3 Kiến trúc 12](#_Toc166440041)

[2.4 Triển khai 13](#_Toc166440042)

[2.4.1 Tổng quan hệ thống 13](#_Toc166440043)

[2.4.2 Sơ đồ thuật toán 14](#_Toc166440044)

[2.4.3 Sơ đồ luồng dữ liệu 15](#_Toc166440045)

[2.5 Chi tiết các bước lập trình 16](#_Toc166440046)

[2.5.1 Lập trình cho module Quectel MC60 16](#_Toc166440047)

[2.5.2 Lập trình lấy dữ liệu từ cảm biến gia tốc BMA253 20](#_Toc166440048)

[2.5.3 Lập trình đóng gói dữ liệu và ghi vào bộ nhớ SPI Flash 22](#_Toc166440049)

[2.5.4 Cấu hình Ngắt và Timer STM32 22](#_Toc166440050)

[2.6 Cách đo đạc 25](#_Toc166440051)

[2.7 Cách xử lý dữ liệu 26](#_Toc166440052)

[2.7.1 Dữ liệu đo năng lượng 26](#_Toc166440053)

[2.7.2 Dữ liệu vị trí từ thiết bị hành trình 27](#_Toc166440054)

[CHƯƠNG 3. KẾT QUẢ NGHIÊN CỨU 28](#_Toc166440055)

[3.1 Kiểm quả đo năng lượng 28](#_Toc166440056)

[3.2 Kết quả xử lý dữ liệu 30](#_Toc166440057)

[CHƯƠNG 4. KẾT LUẬN 31](#_Toc166440058)

[4.1 Kết luận chung 31](#_Toc166440059)

[4.2 Hướng phát triển 31](#_Toc166440060)

[TÀI LIỆU THAM KHẢO 32](#_Toc166440061)

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| Ký hiệu | Nội dung | Ý nghĩa |
| IoT | Internet of Things | Mạng Internet vạn vật |
| MQTT | Message Queuing Telemetry Transport | Giao thức truyền thông hàng đợi thông báo |
| GPS | Global Positioning System | Hệ thống định vị toàn cầu |
| GNSS | Global Navigation Satellite System | Hệ thống vệ tinh định vị toàn cầu |
| UART | Universal Asynchronous Receiver/Transmitter | Bộ nhận và phát không đồng bộ đa năng |
| SPI | Serial Peripheral Interface | Giao diện ngoại vi nối tiếp |
| I2C | Inter-Integrated Circuit | Mạch tích hợp nội viện |
| SRAM | Static Random Access Memory | Bộ nhớ truy cập ngẫu nhiên tĩnh |
| RTC | Real-Time Clock | Đồng hồ thời gian thực |
| PLL | Phase-Locked Loop | Vòng khóa pha |
| HSI RC | High-Speed Internal RC Oscillator | Dao động nội tốc độ cao RC |
| INT | Interrupt | Ngắt |
| DC | Direct Current | Dòng điện một chiều |
| RMSE | Root Mean Square Error | Căn bậc hai trung bình của các sai số bình phương (sai số dự đoán) |

# DANH MỤC HÌNH VẼ

[Hình 2.1. Kiến trúc của Publish/Subscribe của MQTT 7](#_Toc166405401)

[Hình 2.2. Sơ đồ chuyển đổi chế độ năng lượng 10](#_Toc166405402)

[Hình 2.3. Sơ đồ thời gian cho chế độ năng lượng thấp 1/2 EDT (event-driven time) 11](#_Toc166405403)

[Hình 2.4. Kiến trúc hệ thống 12](#_Toc166405404)

[Hình 2.5. Tổng quan hệ thống 13](#_Toc166405405)

[Hình 2.6. Sơ đồ thuật toán 14](#_Toc166405406)

[Hình 2.7. Sơ đồ luồng dữ liệu 16](#_Toc166405407)

[Hình 2.8. Thiết lập kết nối UART3 với phần GNSS của MC60 bằng công cụ STM32CubeMX 17](#_Toc166405408)

[Hình 2.9. Thiết lập kết nối UART2 với phần GNSS của MC60 bằng công cụ STM32CubeMX 18](#_Toc166405409)

[Hình 2.10. Dữ liệu thô từ cảm biến GPS 19](#_Toc166405410)

[Hình 2.11. Mô tả hàm xử lý chuỗi NMEA GNRMC 19](#_Toc166405411)

[Hình 2.12. Dữ liệu GPS đã được xử lý 20](#_Toc166405412)

[Hình 2.13. Thiết lập kết nối I2C cho BMA253 bằng công cụ STM32CubeMX 21](#_Toc166405413)

[Hình 2.14. Cấu hình ngắt cho STM32F030CCT6 bằng STM32CubeMX 23](#_Toc166405414)

[Hình 2.15. Cấu hình Timer cho STM32F030CCT6 bằng STM32CubeMX 24](#_Toc166405415)

[Hình 2.16. Máy đo Keysight N6705B 25](#_Toc166405416)

[Hình 3.1. Biểu đồ dòng tiêu thụ trong các chế độ của thiết bị 28](#_Toc166405417)

[Hình 3.2. Biểu đồ dòng tiêu thụ với 40 lần gửi dữ liệu của thiết bị 29](#_Toc166405418)

[Hình 3.3. Tọa độ vị trí trước và sau khi xử lí 30](#_Toc166405419)

# DANH MỤC BẢNG BIỂU

[Bảng 1.1. So sánh các thiết bị hành trình 2G có trên thị trường 5](#_Toc166405420)

# TÓM TẮT

Trong thời đại công nghệ số phát triển nhanh chóng, thiết bị theo dõi hành trình đã trở thành công cụ không thể thiếu trong nhiều lĩnh vực, từ giao thông vận tải, quản lý tài sản, đến giám sát nhân sự. Các thiết bị này mang lại nhiều lợi ích, giúp cải thiện hiệu suất quản lý, nâng cao độ chính xác và tăng cường tính minh bạch. Tuy nhiên, một trong những thách thức chính mà các thiết bị này đang phải đối mặt là vấn đề tiêu thụ năng lượng kéo dài thời gian hoạt động thiết bị.

Dự án nghiên cứu này nhằm phát triển một cơ chế tối ưu hóa năng lượng tiêu thụ cho các thiết bị theo dõi hành trình phương tiện giao thông. Mục tiêu chính là giảm thiểu lượng năng lượng sử dụng bởi thiết bị, qua đó kéo dài thời gian hoạt động của pin và nâng cao hiệu quả sử dụng năng lượng. Điều này được thực hiện thông qua việc điều khiển linh hoạt các chế độ hoạt động của bộ xử lý trung tâm, bộ thu GPS(Global Positioning System), cảm biến gia tốc và các chế độ truyền dữ liệu.

Nghiên cứu còn tập trung vào việc phân tích và lựa chọn các chế độ hoạt động tiết kiệm năng lượng cho các thành phần chính của thiết bị, như bộ xử lý trung tâm và bộ thu GPS, mà không làm giảm hiệu suất của thiết bị theo dõi. Bên cạnh đó, cơ chế truyền dữ liệu cũng được cải tiến bằng cách sử dụng giao thức MQTT(Message Queueing Telemetry Transport) , một giao thức truyền thông hạng nhẹ giúp giảm băng thông và năng lượng tiêu thụ khi truyền dữ liệu.

Kết quả thực nghiệm cho thấy hệ thống có thể tiết kiệm năng lượng đáng kể, dòng tiêu thụ trung bình thấp nhất 0.8mA khi thiết bị hành trình phát hiện phương tiện không di chuyển. Khi sử dụng cơ chế tối ưu năng lượng, RMSE (Root Mean Square Error) độ lệch so với dữ liệu chuẩn đạt được 9m. Qua đó, nghiên cứu mở ra hướng tiếp cận mới trong việc thiết kế và triển khai các thiết bị định vị có hiệu quả năng lượng cao, góp phần vào sự phát triển bền vững của các ứng dụng IoT. Nghiên cứu này sẽ được chia thành các phần sau:

* Chương 1: Đặt vấn đề
* Chương 2: Phương pháp nghiên cứu
* Chương 3: Kết quả nghiên cứu
* Chương 4: Kết luận

# ĐẶT VẤN ĐỀ

Hiện nay, nhu cầu về các thiết bị theo dõi hành trình ngày càng gia tăng, đặt ra yêu cầu cấp thiết về việc kéo dài thời gian hoạt động của pin và tối ưu hóa chi phí năng lượng. Trong bối cảnh này, việc nghiên cứu và phát triển các cơ chế tiết kiệm năng lượng cho thiết bị theo dõi hành trình là rất quan trọng, không chỉ để nâng cao hiệu quả sử dụng mà còn giúp giảm tác động đến môi trường.

Một số nghiên cứu đã được thực hiện để giải quyết vấn đề năng lượng trong lĩnh vực này, bao gồm các phương pháp tối ưu hóa phần cứng và phần mềm, như sử dụng vi điều khiển với chế độ tiết kiệm năng lượng và giao thức truyền thông hạng nhẹ. Tuy nhiên, việc kết hợp và áp dụng hiệu quả những công nghệ này vào thiết kế thiết bị theo dõi hành trình vẫn cần được nghiên cứu và đánh giá kỹ lưỡng. Nhóm chúng em đã khảo sát các sản phẩm trên thị trường để từ đó đánh giá như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Hình ảnh  Phân loại | A close-up of a device  Description automatically generated | A black rectangular object with white text  Description automatically generated |  |
| Tên sản phẩm | TELTONIKA FMB230 [1] | GO TRACK S200 2G [2] | R11 - 2G [3] |
| Dòng tiêu thụ | < 2.5 mA (Ultra Deep Sleep)  < 5 mA (Deep Sleep)  < 5.5 mA (Online Deep Sleep)  < 18,5 mA (GPS Sleep)  < 34 mA (nominal with no load) | Active/Idle/GPSsleep:  48÷60mA/22÷24mA/ 10÷12mA | Active/Idle:  25mA/2.5mA |

Bảng 1.1. So sánh các thiết bị hành trình 2G có trên thị trường

Các thiết bị sử dụng công nghệ 2G trên bảng 1.1 cho thấy dòng tiêu thụ năng lượng cao, điều này còn hạn chế khả năng hoạt động liên tục của thiết bị, nó đòi hỏi một sự cải tiến đáng kể trong cơ chế quản lý và tối ưu hóa năng lượng để kéo dài tuổi thọ pin và tối đa hóa thời gian hoạt động của thiết bị. Do đó, đề tài "Nghiên cứu phát triển cơ chế tối ưu năng lượng tiêu thụ cho thiết bị theo dõi hành trình" được thực hiện nhằm mục tiêu cải thiện hiệu quả sử dụng năng lượng, kéo dài thời gian hoạt động của thiết bị, và nâng cao hiệu suất tổng thể. Đề tài này sẽ tập trung vào việc phân tích và lựa chọn các chế độ hoạt động tiết kiệm năng lượng cho các thành phần chính của thiết bị, như bộ xử lý trung tâm, bộ thu GPS, và cơ chế truyền dữ liệu. Đồng thời, nghiên cứu cũng tìm hiểu cách thức áp dụng hiệu quả những giải pháp này vào thiết kế tổng thể, góp phần vào sự phát triển bền vững của các thiết bị điện tử và ứng dụng IoT.

# PHƯƠNG PHÁP NGHIÊN CỨU

## Cơ sở lý thuyết

### Kalman Filter

#### Cơ sở toán học

All you need to know is basic terms and operations such as:

Vector and matrix addition and multiplication (cộng và nhân vector, ma trận)

Matrix transpose (ma trận chuyển vị)

Matrix Inverse (you don't need to invert matrices by yourself, you just need to know what the inverse of the matrix is) (ma trận nghịch đảo)

Symmetric matrices (ma trận đối xứng)

Eigenvalues and eigenvectors (vecto riêng)

A diagram of a cycle

Description automatically generated

Hình mô tả hoạt động Dự đoán và Làm đúng của Kalman Filter

A diagram of a mathematical equation

Description automatically generated

Hình mô tả bức tranh hoàn chỉnh về bộ lọc Kalman

A screenshot of a computer

Description automatically generated

Dimensions notation:

* nx𝑛𝑥 is a number of states in a state vector
* nz𝑛𝑧 is a number of measured states
* nu𝑛𝑢 is a number of elements of the input variable

A screenshot of a math application

Description automatically generated

### Mạng 2G

2G (còn viết là 2-G) là tên viết tắt của công nghệ mạng di động viễn thông (hay có thể gọi là công nghệ mạng không dây tế bào - wireless cellular technology) thế hệ thứ hai. [6]

* Tốc độ truyền: Với công nghệ dịch vụ vô tuyến gói tổng hợp (GPRS), mạng 2G cung cấp tốc độ truyền tối đa theo lý thuyết là 0.5 Mbit/s (114 kbit/s trên thực tế).
* Độ trễ: Độ trễ trong mạng 2G tương đối cao so với các mạng di động mới hơn. Vậy 2G không phù hợp cho các ứng dụng đòi hỏi phản hồi thời gian thực như các ứng dụng thời gian thực, trò chơi trực tuyến hay streaming video.
* Hỗ trợ dịch vụ IoT: Mặc dù không được thiết kế cho IoT, mạng 2G vẫn có thể hỗ trợ các thiết bị IoT đơn giản do yêu cầu năng lượng thấp và băng thông không cao. Nó phù hợp cho các ứng dụng IoT không cần truyền dữ liệu lớn hoặc truyền tải thường xuyên.

Tuy nhiên, chất lượng có thể giảm trong các điều kiện sóng yếu hoặc khi di chuyển nhanh.

### Cơ chế tiết kiệm năng lượng

Ở nghiên cứu này chúng em trình bày về các cơ chế tiết kiệm của từng thành phần bao gồm: vi điều khiển STM32F030CCT6, module GSM và GNSS Quectel MC60, cảm biến gia tốc BMA253.

#### Cơ chế tiết kiệm năng lượng của STM32F030CCT6

Các cơ chế độ tiết kiệm năng lượng của STM32 là một tính năng quan trọng được tích hợp trong các vi điều khiển STM32 của STMicroelectronics. Chế độ này giúp giảm tiêu thụ năng lượng của vi điều khiển khi không hoạt động hoặc khi hoạt động ở mức công suất thấp. Dòng vi điều khiển STM32F030x4/x6/x8/xC hỗ trợ ba chế độ tiết kiệm năng lượng [7] bao gồm:

* Chế độ Ngủ (Sleep mode): Trong Chế độ Ngủ, chỉ có CPU là bị dừng lại. Tất cả các thiết bị ngoại vi vẫn tiếp tục hoạt động và có thể đánh thức CPU khi có một ngắt/sự kiện xảy ra.
* Chế độ Dừng (Stop mode): Chế độ Dừng đạt được mức tiêu thụ năng lượng rất thấp trong khi vẫn giữ nội dung của SRAM và các thanh ghi. Tất cả các bộ đồng hồ trong miền điện áp 1.8 V được dừng lại, các dao động PLL, HSI RC và dao động kết tinh HSE được vô hiệu hóa. Bộ điều chỉnh điện áp cũng có thể được đặt ở chế độ bình thường hoặc chế độ tiết kiệm năng lượng thấp.
* Chế độ Chờ (Standby mode): Chế độ Chờ được sử dụng để đạt được mức tiêu thụ năng lượng thấp nhất. Bộ điều chỉnh điện áp nội bộ được tắt để toàn bộ miền điện áp 1.8 V được tắt nguồn. Các dao động PLL, HSI RC và dao động kết tinh HSE cũng được tắt. Sau khi vào Chế độ Chờ, nội dung SRAM và các thanh ghi sẽ bị mất trừ các thanh ghi trong miền RTC và mạch Chế độ Chờ.

Trong nghiên cứu này chúng em sử dụng chế độ Dừng (Stop mode) để phát triển cơ chế tiết kiệm cho thiết bị theo dõi hành trình. Chúng em lựa chọn chế độ này cho nghiên cứu để đánh giá hiệu quả mức tiêu thụ năng lượng của thiết bị theo dõi hành trình trong trường hợp phương tiện sử dụng thiết bị không hoạt động. Lựa chọn thiết bị này cho nghiên cứu như trình bày ở trên sẽ vẫn giữ lại nội dung của SRAM và các thanh ghi giúp cho việc khởi động lại thiết bị nhanh chóng cũng như phù hợp với thiết kế.

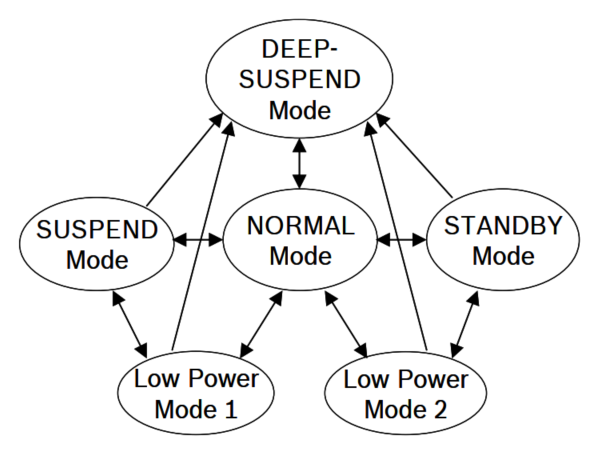
#### Cơ chế tiết kiệm năng lượng của module Quectel MC60

Module MC60 hỗ trợ nhiều chế độ năng lượng thấp khác nhau cho cả bộ phận GSM và GNSS [8] . Đối với phần GSM bao gồm chế độ chức năng tối thiểu và chế độ ngủ. Ở chế độ chức năng tối thiểu, phần GSM có thể giảm đáng kể mức tiêu thụ điện năng bằng cách vô hiệu hóa các chức năng RF. Còn ở chế độ ngủ cổng UART chính sẽ dừng hoạt động, không nhận các lệnh AT. Đối với phần GNSS bao gồm chế độ đầy đủ(Full on Mode) , chế độ chờ (Standby Mode), chế độ dự phòng(Backup Mode), chế độ chu kỳ(Periodic Mode), chế độ AlwaysLocate™ ( AlwaysLocate™ Mode), chế độ GLP (GLP Mode).

Ở phần này chúng em không trình bày kỹ vì trong quá trình nghiên cứu khi sử dụng cơ chế tiết kiệm năng lượng cho module MC60 ảnh hưởng đến hiệu năng của thiết bị rất lớn, độ trễ cao.

#### Cơ chế tiết kiệm năng lượng của cảm biến gia tốc BMA253

Cảm biến gia tốc BMA253 của Bosch cung cấp 6 chế độ năng lượng hoạt động [9] như minh hoạ hình 2.2 bên dưới.



Hình 2.2. Sơ đồ chuyển đổi chế độ năng lượng

Sáu chế độ năng lượng của BMA253 bao gồm:

* Chế độ hoạt động bình thường (normal-mode): hoạt động bình thường, thu thập dữ liệu liên tục.
* Chế độ treo sâu (deep-suspend mode):
* Thiết bị đạt được mức tiêu thụ thấp nhất.
* Chỉ phần giao diện được giữ.
* Không thực hiện việc thu thập dữ liệu và nội dung của các thanh ghi cấu hình bị mất.
* Chế độ treo (suspend mode):
* Phần analog được tắt nguồn.
* Không thực hiện việc thu thập dữ liệu.
* Dữ liệu gia tốc mới nhất và nội dung của tất cả các thanh ghi cấu hình được giữ.
* Đọc và ghi các thanh ghi được hỗ trợ ngoại trừ một số thanh ghi liên quan đến fifo.
* Chế độ chờ (standby mode):
* Thiết bị chuyển đổi định kỳ giữa một giai đoạn ngủ và một giai đoạn thức dậy.
* Giai đoạn thức dậy về cơ bản tương ứng với hoạt động trong chế độ bình thường với việc hoàn toàn bật nguồn cho các mạch điện.
* Giai đoạn ngủ về cơ bản tương ứng với hoạt động trong chế độ treo.
* Chế độ tiết kiệm năng lượng 1:
* Thiết bị sẽ chuyển đổi định kỳ giữa giai đoạn ngủ và giai đoạn thức.
* Giai đoạn đánh thức về cơ bản tương ứng với hoạt động ở chế độ bình thường .Giai đoạn ngủ về cơ bản tương ứng với hoạt động ở chế độ tạm dừng. T
* Chế độ tiết kiệm năng lượng 2:
* Tương tự như chế độ tiết kiệm năng lượng 1, nhưng truy cập vào các thanh ghi được thực hiện bất kỳ lúc nào mà không có hạn chế.
* Tiêu thụ năng lượng nhiều hơn so với chế độ tiết kiệm năng lượng 1.
* Thiết bị chuyển đổi định kỳ giữa một giai đoạn ngủ và một giai đoạn thức dậy, với giai đoạn thức dậy về cơ bản tương ứng với hoạt động trong chế độ bình thường và giai đoạn ngủ về cơ bản tương ứng với hoạt động trong chế độ chờ.

Trong quá trình di chuyển của xe, cảm biến gia tốc luôn phải phát hiện chuyển động của xe, do đó luôn phải truy cập vào các thanh ghi và đồng thời phải giảm điện năng tiêu thụ. Do đó chế độ tiết kiệm năng lượng 2 là chế độ tối ưu nhất cho thiết bị. Một điểm đặc biệt nữa là cảm biến chuyển động có độ nhạy thấp và hoạt động ở tần số thấp (125Hz) vì thế cảm biến sẽ dành phần lớn thời gian ở chế độ ngủ.

A diagram of a sample

Description automatically generated

Hình 2.3. Sơ đồ thời gian cho chế độ năng lượng thấp 1/2 EDT (event-driven time)

## Yêu cầu hệ thống

* Sử dụng timer lấy mẫu dữ liệu GPS theo chu kì , sử dụng hàng đợi để lưu trữ bản tin, kích thước gói đủ số bản tin mong muốn sẽ được gửi lên MQTT broker.
* Sử dụng MQTT với qos = 2.
* Sử dụng bộ nhớ flash để lưu trữ dữ liệu khi bị mất kết nối, khi có kết nối trở lại thì gửi bản tin trong flash.
* Sử dụng cảm biến gia tốc BMA253 phát hiện chuyển động/không chuyển động, sau 5 phút thiết bị phát hiện không chuyển động đưa vào chế độ tiết kiệm năng lượng, có chuyển động sẽ hoạt động bình thường trở lại.
* Sử dụng chế độ tiết kiệm năng lượng để khả năng tồn tại của thiết bị lâu nhất.
* Hiển thị dữ liệu vị trí lên trên bản đồ.

## Kiến trúc

A diagram of a power system

Description automatically generated

Hình 2.4. Kiến trúc hệ thống

* Khối nguồn: Chuyển đổi điện áp cao (12VDC) xuống các điện áp thấp. Các mức điện áp thấp là 4.2VDC nuôi MC60 moudle và để sạc cho “battery” dự phòng để sử dụng trong trường hợp mất nguồn 12VDC, 2.8VDC dùng để nuôi STM32, nand flash, accelerometer(BMA253), các nguồn phụ MC60.
* Khối điều khiển trung tâm: STM32 dùng để điều khiển toàn bộ hệ thống.
* Khối GPS và GSM(MC60): Giao tiếp với vi điều khiển chính thông qua UART, GPS module lấy tọa độ GPS, GSM module kết nối để vào mạng internet thông qua giao thức MQTT.
* Khối phát hiện chuyển động(BMA 253) : Giao tiếp với vi điều khiển thông qua I2C. BMA253 phát hiện chuyển động, không chuyển động, va chạm.
* Khối lưu data(Flash): Giao tiếp với vi điều khiển thông qua SPI. Flash để lưu trữ các dữ liệu GPS.
* Khối giao tiếp với máy tính: Khối này giáo tiếp với vi điều khiển qua SWD và UART. Dùng để nạp chương trình. Nạp các tham số ban đầu.
* Khối đèn led: Thông báo hiển thị để hiển thị trạng thái của thiết bị.

## Triển khai

### Tổng quan hệ thống

A diagram of a data processing process

Description automatically generated

Hình 2.5. Tổng quan hệ thống

Hệ thống định vị được mô tả trong hình trên bao gồm một loạt các thiết bị theo dõi (Trackers), mỗi thiết bị được thiết kế để thu thập dữ liệu về vị trí của chúng và gửi thông tin này lên nền tảng Adafruit IO. Sử dụng công nghệ IoT tiên tiến, mỗi tracker cập nhật dữ liệu vị trí của mình thông qua mạng không(2G). Dữ liệu này sau đó được tổng hợp và lưu trữ trên Adafruit IO, một nền tảng dịch vụ điện toán đám mây, cho phép lưu trữ và truy cập dữ liệu từ xa.

Sau khi dữ liệu được tải lên, nó có thể được truy xuất và sử dụng để hiển thị trên các bản đồ số, cho phép người dùng xác định chính xác vị trí của mỗi thiết bị theo dõi trong thời gian thực. Điều này đặc biệt hữu ích cho các ứng dụng như quản lý phương tiện, theo dõi hành trình, và an ninh cá nhân, nơi mà việc xác định vị trí chính xác và kịp thời là rất quan trọng.

Để tối ưu hóa hiệu quả năng lượng của hệ thống, mỗi thiết bị theo dõi đã được tích hợp cơ chế tiết kiệm năng lượng. Cơ chế này bao gồm các thuật toán tiên tiến cho phép thiết bị chuyển vào chế độ nghỉ hoặc giảm mức tiêu thụ năng lượng khi không cần thiết, qua đó kéo dài tuổi thọ của pin và giảm chi phí vận hành. Chi tiết về các cơ chế tiết kiệm năng lượng này và hiệu quả của chúng trong thực tế sẽ được trình bày chi tiết trong các phần sau của bài báo.

### Sơ đồ thuật toán

A diagram of a flowchart

Description automatically generated

Hình 2.6. Sơ đồ thuật toán

Dưới đây là sơ đồ thuật toán miêu tả một quy trình phần mềm của chúng em gửi dữ liệu từ cảm biến qua mạng 2G sử dụng giao thức MQTT và cơ chế tiết kiệm năng lượng :

* Ngắt thực hiện: Thu thập dữ liệu từ cảm biến được thu thập thông qua hai loại ngắt:
* Ngắt cảm biến gia tốc: Được kích hoạt khi có chuyển động.
* Ngắt hẹn giờ: Kích hoạt theo chu kỳ, thu thập mẫu dữ liệu và đẩy vào hàng đợi.
* Hàm main:

Bước 1. Kiểm tra hàng đợi sau khi thu thập dữ liệu, thuật toán kiểm tra xem hàng đợi đã đầy chưa:

* Nếu hàng đợi chưa đầy, dữ liệu mới được thêm vào.
* Nếu hàng đợi đầy, tiếp tục đến bước tiếp theo.

Bước 2. Kiểm tra xem kết nối MQTT có thành công không:

* Nếu không, dữ liệu được lưu vào bộ nhớ Flash để lưu trữ tạm thời.
* Nếu kết nối thành công, tiếp tục đến bước tiếp theo.

Bước 3. Gửi dữ liệu, kiểm tra xem có dữ liệu trong bộ nhớ Flash không:

* Nếu có, dữ liệu được gửi qua MQTT với mức QoS=2 (đảm bảo độ tin cậy).
* Nếu không, dữ liệu từ hàng đợi được gửi trực tiếp.

Bước 4. Ngắt kết nối, sau khi gửi dữ liệu, ngắt kết nối MQTT để tiết kiệm năng lượng:

Bước 5. Kiểm tra chuyển động, kiểm tra xem thiết bị có chuyển sang trạng thái "không chuyển động" hay không:

* Nếu không, quay lại bước 1.
* Nếu có, thiết bị sẽ chuyển sang chế độ ngủ (Sleep), tắt MCU và kích hoạt ngắt gia tốc kế để chờ các tín hiệu tiếp theo.

### Sơ đồ luồng dữ liệu

A diagram of a software system

Description automatically generated with medium confidence

Hình 2.7. Sơ đồ luồng dữ liệu

* Dữ liệu thô (Data raw) được vi điều khiển STM32F030CCT6 lấy được từ module MC60 và chuyển đổi Data convert (thời gian, vĩ độ(latitude), kinh độ(longitude)) và dữ liệu của cảm biến gia tốc BMA253 (Data motion) được kiểm tra xem có đang chuyển động hay không.
* Dữ liệu đã được xử lý từ MC60 (Data convert) sẽ được đưa vào một hàng đợi với số lượng phần tử mong muốn để đóng gói thành Data package và khi đã đủ số lượng nếu có kết nối mạng thì Data package sẽ được publish lên Adafruit thông qua giao thức MQTT của module MC60. Và nếu không có kết nối mạng Data package sẽ được lưu trữ vào SPI flash.

Quá trình này cho phép MCU thu thập dữ liệu từ module MC60 và sau đó gửi nó lên MQTT Broker thông qua hàng đợi Queue nhờ vào GSM của MC60. Việc sử dụng bộ nhớ SPI Flash giúp lưu trữ dữ liệu một cách lâu dài và chống mất mát dữ liệu thiết bị hành trình mất kết nối mạng.

## Chi tiết các bước lập trình

### Lập trình cho module Quectel MC60

#### Lập trình kết nối mạng từ phần GSM của MC60

Để thực hiện kết nối mạng và gửi dữ liệu, trước hết cần khởi động và thiết lập kết nối giữa phần GSM của MC60 với MCU STM32F030CCT6 thông qua giao tiếp UART nên việc thiết lập kết nối cũng sử dụng STM32CubeMX. Vì vậy, để có thể kết nối mạng 2G, trước hết ta cần bật kết nối UART ở STM32. Ở công cụ tạo mã STM32CubeMX đã có sẵn chức năng thiết lập UART nên ta chỉ cần thiết lập các thông số UART ở đây, mã thiết lập UART sẽ tự động được tạo trong chương trình.

A screenshot of a computer

Description automatically generated

Hình 2.8. Thiết lập kết nối UART3 với phần GNSS của MC60 bằng công cụ STM32CubeMX

Sau khi kích hoạt UART thành công, việc tiếp theo cần làm là truyền các lệnh AT thông qua UART để thực hiện kết nối mạng. Lệnh AT (Attention) là một chuỗi các ký tự dùng để giao tiếp với các thiết bị thông qua giao thức truyền thông chuẩn UART. Nguyên tắc hoạt động của lệnh AT rất đơn giản. Khi thiết bị nhận được một chuỗi ký tự bắt đầu bằng "AT" (hoặc "at" nếu không phân biệt chữ hoa, chữ thường), nó hiểu rằng đó là một lệnh AT và sẽ thực hiện các hành động tương ứng với lệnh đó.

Chi tiết về chức năng của các lệnh AT sẽ nằm trong tài liệu hướng dẫn của nhà sản xuất (Quectel) cung cấp [10] [11].

#### Lập trình lấy dữ liệu GPS từ phần GNSS của MC60

Như đã nói ở phần 2.3 module Quectel MC60 được kết nối với MCU thông qua giao tiếp UART vì vậy, để có thể lấy dữ liệu GNSS, trước hết ta cần bật kết nối UART ở STM32. Ở công cụ tạo mã STM32CubeMX đã có sẵn chức năng thiết lập UART nên ta chỉ cần thiết lập các thông số UART ở đây, mã thiết lập UART sẽ tự động được tạo trong chương trình.

A screenshot of a computer

Description automatically generated

Hình 2.9. Thiết lập kết nối UART2 với phần GNSS của MC60 bằng công cụ STM32CubeMX

Sau khi kích hoạt UART2 thành công, việc tiếp theo cần làm là khởi động module để lấy dữ liệu. Theo như tài liệu của nhà sản xuất, khi module được cấp nguồn, nó sẽ tự gửi dữ liệu thô (Data raw) qua kết nối UART2.

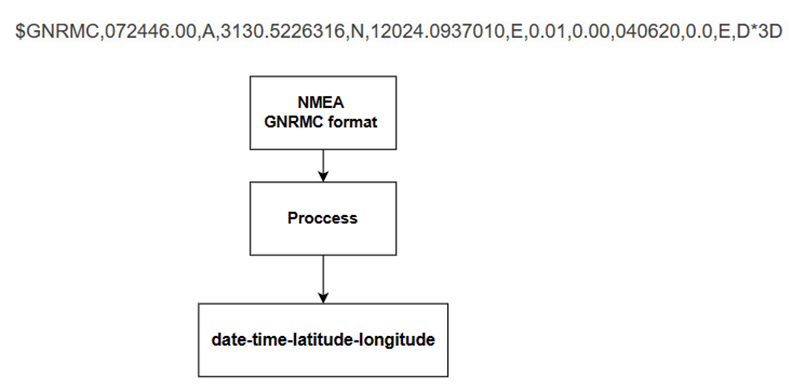
Dữ liệu thô của module Quectel MC60 là các bản tin NMEA (National Marine Electronics Association) là một chuẩn giao thức được sử dụng rộng rãi để truyền tải thông tin định vị và dữ liệu GPS giữa các thiết bị định vị và các thiết bị nhận dữ liệu. Dữ liệu nhận được từ UART2 sẽ được chuyển cho UART1 kết nối với PC và hiển thị lên màn hình thông qua phần mềm Hercules [12] có dạng như hình dưới đây.

A screenshot of a computer

Description automatically generated

Hình 2.10. Dữ liệu thô từ cảm biến GPS

Ở đây, dòng $GNRMC có chứa dữ liệu ta cần về các dữ liệu cần thiết về tọa độ, thời gian. Việc ta cần làm là áp dụng các hàm xử lý chuỗi để lấy được các dữ liệu này về các dạng dữ liệu có thể đọc được.



Hình 2.11. Mô tả hàm xử lý chuỗi NMEA GNRMC

Sau khi thực hiện hàm xử lý chuỗi từ dữ liệu thô thu được, dữ liệu dược xử lý sẽ có kết quả như hình sau:

A screenshot of a computer

Description automatically generated

Hình 2.12. Dữ liệu GPS đã được xử lý

Trên đây là tổng quan về cách lấy dữ liệu GPS từ module Quectel MC60 bằng vi điều khiển STM32F030CCT6. Để biết thêm chi tiết về dữ liệu NMEA [13] và các thông tin khác về thiết lập module Quectel MC60 có thể xem thêm trong tài liệu của nhà sản xuất [8].

### Lập trình lấy dữ liệu từ cảm biến gia tốc BMA253

Ở thiết bị này chúng em sử dụng hai chức năng chính của cảm biến [9]:

* Phát hiện chuyển động (sử dụng cả 3 trục x, y, z)
* Phát hiện ngừng chuyển động (sử dụng cả 3 trục x, y, z)

Đầu tiên cấu hình cho cảm biến những hàm cần thiết:

* Đọc theo chuẩn giao tiếp I2C
* Ngưỡng phát hiện thay đổi gia tốc ±2G
* Dải tần số cập nhật dữ liệu 125Hz
* Ánh xạ ngắt chân ngắt phát hiện chuyển động với ngắt ngoài của vi điều khiển

Khi sử dụng cảm biến này đầu tiên cần cấu hình giao tiếp I2C bằng công cụ STM32CubeMX

A screenshot of a computer

Description automatically generated

Hình 2.13. Thiết lập kết nối I2C cho BMA253 bằng công cụ STM32CubeMX

Khi thiết bị bắt đầu được nạp chương trình:

Sử dụng một cờ để kiểm soát trạng thái của thiết bị, với trạng thái WAKEUP, thiết bị được cấu hình các hàm cần thiết cho module BMA253. Một hàm liên tục đọc thông tin tại thanh ghi trạng thái “phát hiện ngừng chuyển động”, nếu trả về trạng thái không chuyển động, cho phép thanh ghi ngắt “phát hiện chuyển động” được hoạt động (nhằm lấy được tín hiệu ngắt ngoài khi phát hiện chuyển động giúp đánh thức thiết bị khỏi trạng thái dừng (stop mode) của STM32F030CCT6). Thiết bị được đưa về trạng thái stop sau S giây đứng yên.

Khi đang ở trạng thái dừng (stop mode):

Nếu thiết bị có chuyển động, tín hiệu từ cảm biến gia tốc (phát hiện chuyển động) sẽ được kích hoạt kéo lên mức cao, hàm xử lý ngắt ngoài của vi điều khiển sẽ kiểm tra 2 điều kiện thỏa mãn: chân ngắt ngoài tại vị trí tương ứng với cảm biến BMA253 và cờ trạng thái là STOP, chuyển trạng thái cờ về WAKEUP để bật thiết bị và tắt ngắt phát hiện chuyển động.

Mục đích phải kiểm soát thanh ghi ngắt “phát hiện chuyển động”: tránh việc cảm biến gửi ngắt liên tục về vi điều khiển trong quá trình di chuyển gây lãng phí năng lượng và giảm sút tuổi thọ chân tín hiệu do nhận mức điện áp thay đổi liên tục. Đồng thời vi điều khiển tiết kiệm được tài nguyên khi không phải thực hiện nhiều dòng lệnh không có giá trị.

### Lập trình đóng gói dữ liệu và ghi vào bộ nhớ SPI Flash

Theo datasheet của nhà sản xuất [14] thì dữ liệu được lưu nhỏ nhất có giá trị theo 1 page của flash là 256 byte, sử dụng cơ chế lưu trữ 4 page cho mỗi lần cập nhật dữ liệu mới. Nếu dữ liệu không được publish(lỗi kết nối), thực hiện lưu trữ bằng cách sử dụng một con trỏ kiểm soát vị trí hiện tại(bao gồm vị trí đầu và vị trí cuối) của dữ liệu mới nhất . Khi kết nối lại thành công, đẩy dữ liệu lên nền tảng lưu trữ thông qua giao thức MQTT, con trỏ được cập nhật vị trí mới còn dữ liệu cũ vẫn được lưu trữ trong flash. Đặc biệt khi con trỏ tới vị trí giới hạn của bộ lưu trữ, thực hiện xóa toàn bộ dữ liệu cũ là đặt con trỏ về vị trí đầu tiên bắt đầu vòng lưu trữ mới.

Nguyên nhân không xóa dữ liệu cũ sau mỗi lần publish, do theo datasheet của nhà sản xuất [14], thiết bị lưu trữ có tính năng xóa nhỏ nhất mỗi lần một sector tương ứng 4KB, thời gian thực hiện tối đa 400ms và xóa toàn bộ chip thực hiện tối đa 25s nên trong trường hợp tệ nhất cùng xóa toàn bộ các dữ liệu có trong bộ nhớ ngoài bằng lệnh xóa từng sector sẽ hết xấp xỉ 204s lớn hơn rất nhiều so với con số 25s cho toàn bộ chip .

### Cấu hình Ngắt và Timer STM32

#### *Cấu hình ngắt STM32*

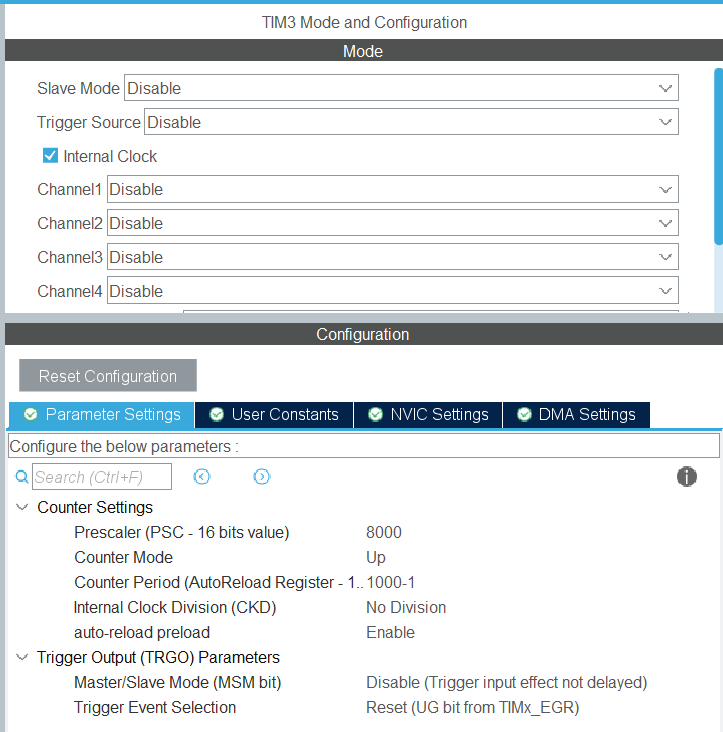
Ngắt (Interrupt) trong vi điều khiển STM32 là một cơ chế cho phép thiết bị phản hồi, nhanh chóng đối với các sự kiện ngoài vi xảy ra trong hệ thống, như xảy ra tín hiệu tại chân GPIO, hoàn thành một giao tiếp truyền nhận dữ liệu, hoặc kết thúc đếm của một bộ đếm thời gian (timer). Khi xảy ra ngắt, vi điều khiển sẽ tạm dừng công việc hiện tại và thực thi một chương trình ngắt để xử lý sự kiện, sau đó trở lại công việc chính. Cơ chế ngắt giúp hạn chế việc sử dụng polling (kiểm tra liên tục) trong việc kiểm tra trạng thái của các ngoại vi và tiết kiệm tài nguyên CPU, cho phép vi điều khiển tập trung vào các công việc quan trọng khác trong khi vẫn có thể đáp ứng kịp thời các sự kiện bên ngoài.

A screenshot of a computer

Description automatically generated

Hình 2.14. Cấu hình ngắt cho STM32F030CCT6 bằng STM32CubeMX

#### *Cấu hình Timer STM32*



Hình 2.15. Cấu hình Timer cho STM32F030CCT6 bằng STM32CubeMX

Timer trong sử dụng để lấy dữ liệu kích hoạt theo chu kỳ, thu thập mẫu dữ liệu và đẩy vào hàng đợi.

## Cách đo đạc

A close-up of a device

Description automatically generated

Hình 2.16. Máy đo Keysight N6705B

Chúng em sử dụng máy đo Keysight N6705B cho việc đo năng lượng tiêu thụ của thiết bị theo dõi hành trình. Máy đo Keysight N6705B mang lại mức năng suất vượt trội cho việc tìm nguồn và đo điện áp cũng như dòng điện DC vào thiết bị bằng cách tích hợp tối đa 4 bộ cấp nguồn tiên tiến với các tính năng DMM, Scope, Arb và Data Logger.

Các bước sử dụng N6705B [15] đo năng lượng tiêu thụ của thiết bị theo dõi hành trình:

Bước 1: Chuẩn bị máy Keysight N6705B.

Bước 2: Bật nguồn máy N6705.

Bước 3: Chọn 1 trong 4 kênh của máy N6705 trên màn hình bằng các phím điều hướng (trái “˂”, phải “˃”, trên “˄”, dưới ”˅”), và phím chọn “Enter”.

Bước 4: Phóng to kênh đã chọn, ấn phím “Meter View”.

Bước 5: Sử dụng các phím điều hướng lên và xuống để chọn 2 ô nguồn và dòng đầu vào trên màn hình.

Bước 6: Cài đặt nguồn đầu vào trên màn hình kênh bằng các phím số trên máy.

Bước 7: Kết nối dây dương của thiết bị đo với “output” dương (đỏ) của kênh 2, dây âm của thiết bị đo với “output” âm (đen) của kênh.

Bước 8: Ấn phím “Menu”.

Bước 9: Màn hình menu hiển thị trên màn hình, chọn Datalogger trên màn hình.

Bước 10: Màn hình Datalogger hiển thị trên màn hình, sử dụng bàn phím số và các phím di chuyển để cài đặt các thông số là khoảng thời gian lấy mẫu (giờ, phút, giây), khoảng cách lấy mẫu (ns), ...

Bước 11: Sau khi cài đặt xong thông số Datalogger, ấn phím “Data Logger”.

Bước 12: Sử dụng các phim di chuyển để chọn các dữ liệu của kênh gồm có V, I, P tương ứng với điện áp, dòng điện, công suất

Bước 13: Ấn phím “On” ở kênh trên máy đã được kết nối với thiết bị (bước 7) và theo dõi kết quả đo được trên màn hình.

Chúng em đo dòng tiêu thụ của thiết bị theo dõi hành trình với thời gian lấy mỗi mẫu là 100ms và trong 10 phút, được lặp lại trong vòng 1 giờ đồng hồ. Sau khi máy thực hiện xong , sử USB (định dạng FAT32) để lưu kết quả dưới dạng file .csv bằng cách sử dụng tính năng ‘Export’ khi ấn phím “Menu”.

## Cách xử lý dữ liệu

### Dữ liệu đo năng lượng

Sau khi thực hiện đo dòng tiêu thụ với thời gian lấy mỗi mẫu là 100ms trong khoảng thời gian là 1 giờ đồng hồ. Chúng em sẽ tính giá trị trung bình dòng điện tiêu thụ với công thức sau:

Trong đó:

* : là giá trị trung bình cho dòng tiêu thụ của thiết bị (mA).
* N: tổng số mẫu thực hiện
* n: số mẫu thứ n của thiết bị

Tiếp theo khi đã có giá trị dòng điện tiêu thụ trong một giờ chúng em sẽ thực hiện tìm thời gian sử dụng của thiết bị bằng công thức:

Công suất tức thời:

Trong đó:

* t : thời gian sử dụng của thiết bị (h)
* : dung lượng của pin (mAh)
* : là giá trị trung bình cho dòng tiêu thụ của thiết bị (mA).

Từ các giá trị tính toán trên chúng em sẽ thực hiện đánh giá với các sản phẩm đã nêu ở Bảng 1.1.

### Dữ liệu vị trí từ thiết bị hành trình

Nhóm chúng em sử dụng phương pháp nội suy để xác định các vị trí trong khoảng thời gian bị mất. Sử dụng Python để lập trình và triển khai phương pháp này.

Dữ liệu sau khi nội suy sẽ được đem so sánh với dữ liệu chuẩn bằng phương pháp bình phương tối thiểu RMSE [16] để đánh giá được độ lệch giữa dữ liệu chuẩn và dữ liệu chúng em thu được. Dữ liệu sẽ được lưu dưới file .csv được đưa vào một đoạn code RMSE trong VBA của Microsoft Excel.

# KẾT QUẢ NGHIÊN CỨU

## Kiểm quả đo năng lượng

A graph showing a number of blue lines

Description automatically generated

Hình 3.1. Biểu đồ dòng tiêu thụ trong các chế độ của thiết bị

Hình 3.1 mô tả là biểu đồ đường dòng tiêu thụ của thiết bị hành trình. Biểu đồ đường này có trục tung là trục thời gian các điểm cách nhau 100ms, trục hoành là trục giá trị dòng điện tiêu thụ có đơn vị là mili ampe (mA).

Phân tích biểu đồ Hình 3.1:

* Theo trục thời gian, bắt đầu đo 0s thiết bị chưa được nạp code có dòng tiêu thụ xấp xỉ 0 mA. Sau đó nạp code, dòng tiêu thụ tăng lê chưa tới 30mA trong vòng 25s.
* Sau khi nạp code, mạch trực tiếp vào trạng thái hoạt động lần đầu, dòng tiêu thụ được hạ xuống còn trong khoảng 10 đến 14mA lúc này thiết bị đang lấy data và đóng gói vào hàng đợi quá trình này diễn ra trong tối đa 1 phút.
* Khi hàng đợi đã đủ số bản tin mong muốn thì sẽ được gửi đi lập tức dòng tiêu thụ có mức thay đổi từ khoảng 23mA đến 30mA trong vòng 15s và có 1 giá trị cao nhất trong khoảng thời gian này với giá trị dòng điện tiêu thụ lên đến gần 70mA, đây là quá trình lấy và gửi dữ liệu của thiết bị.
* Kể từ khi khởi động đến khi dòng tiêu thụ hạ xuống dưới 1mA vì do sau 5 phút không phát hiện chuyển động BMA253 đã gửi tín hiệu để đưa MCU vào chế độ dừng (stop mode) để giảm dòng tiêu thụ vào chế độ tiết kiệm năng lượng cho toàn hệ thống.
* Giá trị dòng tiêu thụ lớn nhất được thu được là 68,681 mA trong trạng thái hoạt động, thể hiện mức tiêu thụ điện năng cao nhất khi thiết bị hoạt động với chế độ tiết kiệm năng lượng của BMA253 và chế độ hoạt động bình thường của MCU và module MC60.
* Giá trị dòng tiêu thụ nhỏ nhất được thu được là 0.792725 mA khi MCU đang trong chế độ dừng (stop mode)

Biểu đồ trên đã cho thấy sự biến đổi của dòng tiêu thụ điện của thiết bị trong quá trình hoạt động và tiết kiệm năng lượng, cung cấp thông tin quan trọng để đánh giá hiệu suất hoạt động và tiêu thụ năng lượng của thiết bị.

A graph showing a number of data

Description automatically generated with medium confidence

Hình 3.2. Biểu đồ dòng tiêu thụ với 40 lần gửi dữ liệu của thiết bị

Hình 3.2 là biểu đồ dòng tiêu thụ trong khoảng một giờ của thiết bị là phiên bản đầy đủ của Hình 3.1. Để đo được như trên chúng em đã tác động để không cho thiết bị vào chế độ dừng (stop mode) và thực hiện đo mỗi 10 phút sau đó ghép lại thành biểu đồ như trên. Áp dụng công thức (1) với dòng tiêu thụ trung bình trong 1 giờ đạt được là 14mAh (chế độ Active của thiết bị) và giá trị trung bình khi không hoạt động là 0.83mA(chế độ Idle của thiết bị). Từ đó, có thể thấy so với các thiết bị ở Bảng 1.1 thì chế độ Active của thiết bị hành trình của chúng em áp dụng cơ chế tiết kiệm năng lượng đạt được mức tiêu thụ thấp hơn từ 1,5-5 lần ở chế độ hoạt động bình thường và ở chế độ Idle thấp hơn 3-6 lần.

## Kết quả xử lý dữ liệu

|  |  |
| --- | --- |
| A map with blue points and a river  Description automatically generated | A map of a city  Description automatically generated |
| A map with blue dots and white text  Description automatically generated | A map with a red line  Description automatically generated |
| A map with blue lines and white text  Description automatically generated with medium confidence | A map of a city  Description automatically generated |

Hình 3.3. Tọa độ vị trí trước và sau khi xử lí

Sau khi xử lý dữ liệu bằng phương pháp nêu ở mục 2.7.2 chúng em đã thu được kết quả như hình 3.3 nhưng có những giá trị bị mất do khả năng bắt sóng 2G kém của thiết bị (hình phía bên trái) . Những phần dữ liệu bị mất đã được nội suy và vẽ lại hiển thị thành đường đi hoàn chỉnh(hình phía bên phải) kiểm tra bằng RMSE so với dữ liệu chuẩn có độ lệch 9m. Qua đó cho thấy khi thực hiện cơ chế tiêu thụ năng lượng thấp không ảnh hưởng đến độ chính xác của thiết bị.

# KẾT LUẬN

## Kết luận chung

Như vậy, nhóm chúng em đã thực hiện phát triển cơ chế tối ưu năng lượng tiêu thụ cho thiết bị theo dõi hành trình. Kết quả cho thấy hiệu năng của thiết bị trong việc tiêu thụ năng lượng là khá tốt. Điều này cho thấy thiết bị có khả năng hoạt động trong thời gian dài mà khi dùng pin. Tuy nhiên, trong một số trường hợp sử dụng cụ thể, có thể xem xét các cải tiến và phương án tối ưu hóa để tiết kiệm năng lượng hơn nữa và nâng cao hiệu suất hoạt động của thiết bị.

Nghiên cứu cũng đã mở ra các hướng nghiên cứu tiềm năng khác, như tối ưu hóa thuật toán xác định vị trí, cải thiện chất lượng thu thập dữ liệu GPS, tích hợp GPS với các công nghệ khác để đáp ứng nhu cầu ngày càng cao của thị trường.

Phần trình bày báo cáo của nhóm em có thể có một số thiếu sót như trình bày, sai sót về khiến thức chuyên môn, mong thầy cô và các bạn liên hệ nhóm em để đóng góp thêm các ý kiến cho những công việc sau thực hiện tốt hơn.

Nhóm chúng em xin cảm ơn!

## Hướng phát triển

Nhóm chúng em xin đưa ra một số hướng nghiên cứu khác cho thiết bị:

* Nghiên cứu thêm các trường giữ liệu từ cảm biến, từ đó áp dụng thuật toán để tăng độ chính xác cho GPS theo thời gian thực.
* Nghiên cứu kết hợp thêm các cảm biến và dùng học máy để phân tích dữ liệu nhằm phân loại các hành vi di chuyển mẫu phổ biến như quay đầu, dừng đèn đỏ, đỗ lâu. Tạo ra một thói quen di chuyển, lấy thói quen di chuyển kết hợp vs dữ liệu về mật độ ùn tắc các tuyến đường để đưa ra tuyến đường tối ưu.
* Tiếp tục nghiên cứu và áp dụng thêm các phương pháp vào cơ chế tối ưu năng lượng để đạt được mức tiêu thụ tối ưu hơn nữa.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | teltonika, “FMB230,” teltonika, [Trực tuyến]. Available: https://teltonika-gps.com/products/trackers/fmb230. |
| [2] | gotrack365, “s200-2g,” gotrack365, [Trực tuyến]. Available: https://gotrack365.com/product/gps-tracking-device-s200-2g/. |
| [3] | seeworldgps, “R11 - 2G,” seeworldgps, [Trực tuyến]. Available: https://www.seeworldgps.com/product/vehicle-gps-tracker/car-gps-tracker/r11-gps-tracking-device/. |
| [4] | “MQTT,” [Trực tuyến]. Available: https://mqtt.org/. |
| [5] | “MQTT là gì?,” aws, [Trực tuyến]. Available: https://aws.amazon.com/vi/what-is/mqtt/. |
| [6] | “Mạng di động 2G,” wikipedia, [Trực tuyến]. Available: https://vi.wikipedia.org/wiki/2G. |
| [7] | STMicroelectronics, “Datasheet STM32F030CCT6,” [Trực tuyến]. Available: https://www.st.com/resource/en/datasheet/stm32f030f4.pdf. |
| [8] | Quectel, “Quectel MC60 Hardware Design,” Quectel, [Trực tuyến]. Available: https://www.quectel.com/wp-content/uploads/2021/03/Quectel\_MC60\_Hardware\_Design\_V2.1.pdf. |
| [9] | Bosch, “BMA253,” Bosch, [Trực tuyến]. Available: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bma253-ds000.pdf. |
| [10] | Quectel, “Quectel MC60 AT Commands Manual V1.2,” Quectel, [Trực tuyến]. Available: https://robu.in/wp-content/uploads/2018/12/Quectel\_MC60\_AT\_Commands\_Manual\_V1.2.pdf. |
| [11] | Quectel, “MC60 GNSS AGPS Application Note,” Quectel, [Trực tuyến]. Available: https://robu.in/wp-content/uploads/2018/12/Quectel\_MC60\_GNSS\_AGPS\_Application\_Note\_V1.1.pdf. |
| [12] | hw-group, “Hercules Download,” hw-group, [Trực tuyến]. Available: https://www.hw-group.com/product-version/hercules. |
| [13] | Quectel, “MC60 Series GNSS Protocol Specification,” Quectel, [Trực tuyến]. Available: https://robu.in/wp-content/uploads/2018/12/Quectel\_MC60\_Series\_GNSS\_Protocol\_Specification\_V1.1.pdf. |
| [14] | winbond, “SPI Flash,” winbond, [Trực tuyến]. Available: https://docs.rs-online.com/068a/0900766b81622f8d.pdf. |
| [15] | Keysight, “Mouser Keysight N6705B User Mannual,” Keysight, [Trực tuyến]. Available: https://www.mouser.com/pdfdocs/2ugN6705-90001.pdf. |
| [16] | N. T. Thinh, “Đánh giá model trong Machine Learing,” VIBLO, [Trực tuyến]. Available: https://viblo.asia/p/danh-gia-model-trong-machine-learing-RnB5pAq7KPG?fbclid=IwAR0pmjl7OPgLeRHaxdcexVRm79gt6lnh0lQt3Ntbvhc2B\_cSCWkLxoV1SHE. |
| [17] | STMicroelectronics, “Reference Manual STM32F030CCT6,” STMicroelectronics, [Trực tuyến]. Available: https://www.st.com/resource/en/reference\_manual/rm0360-stm32f030x4x6x8xc-and-stm32f070x6xb-advanced-armbased-32bit-mcus-stmicroelectronics.pdf. |
|  |  |