

TDD-FDP-CCE

Design Project (Technical Design Document)

Project Name	THIẾT KẾ VÀ PHÁT TRIỂN HỆ THỐNG PHÁT HIỆN VA CHẠM VÀ GỬI CẢNH BÁO KHẨN CẤP, TÍCH HỢP TÌM XE TRONG BÃI BẰNG BLE		
Student	[1] Đào Vĩnh Khang [2] Phan Nhị Hào [3] Võ Xuân Lộc [4] Nguyễn Hoàng Ngân	ID	22139030 22139019 22139040 22139045
Major	Cyber-Physical System	Supervisor	Assoc. Prof. Phan Van Ca

Mục lục

1	Giới thiệu	6
2	Mục tiêu và yêu cầu kỹ thuật	7
2.1	Nhu cầu của người dùng	7
2.2	Yêu cầu kỹ thuật.....	7
2.2.1	Chức năng	7
2.2.2	Phi chức năng.....	8
3	Lý thuyết cơ bản.....	9
3.1	Flutter.....	9
3.2	Hệ thống định vị toàn cầu (GPS)	10
3.3	Các chuẩn truyền dữ liệu	11
3.3.1	Chuẩn giao tiếp I2C.....	11
3.3.2	Chuẩn giao tiếp không dây Bluetooth	13
4	Kiến trúc hệ thống.....	16
4.1	Sơ đồ khối hệ thống.....	16
4.2	Chi tiết chức năng từng khối.....	16
5	Phân tích và thiết kế các khối	18
5.1	Khối cảm biến	18
5.2	Khối cảnh báo	19
5.3	Nút nhấn – Reset hệ thống.....	22
5.4	Khối xử lý trung tâm	23
5.5	Khối nguồn.....	25
5.6	Sơ đồ khối nguyên lý phần cứng.....	28
6	Lưu đồ giải thuật chức năng của hệ thống.....	29
6.1	Lưu đồ giải thuật hệ thống.....	29
6.2	Lưu đồ giải thuật tìm xe.....	32
6.3	Lưu đồ giải thuật phát hiện va chạm và gửi cảnh báo.....	34
7	Kết quả thực hiện	37
7.1	Kết quả thực hiện phần cứng	37
7.2	Kết quả thực hiện trên phần mềm	38
8	Kết luận và hướng phát triển	44
8.1	Kết luận.....	44
8.2	Hướng phát triển.....	45
9	Phụ lục	46
9.1	Phân chia công việc	46
9.2	Chi nguyên vật liệu.....	46

Danh sách hình ảnh

Hình 3.1 Mô tả chuẩn truyền I2C	12
Hình 3.2 Minh họa quá trình truyền/nhận dữ liệu trong giao thức I2C	12
Hình 3.3 Các thành phần bên trong bộ giao thức BLE	14
Hình 4.1 Sơ đồ khối hệ thống	16
Hình 4.2 Sơ đồ kết nối các linh kiện	17
Hình 5.1 Cảm biến gia tốc MPU6050	18
Hình 5.2 Sơ đồ nguyên lý khối cảm biến MPU6065	19
Hình 5.3 Hình ảnh Buzzer 12x9mm – 93dB	20
Hình 5.4 Sơ đồ các chân của Buzzer 12x9mm – 93dB	20
Hình 5.5 Sơ đồ nguyên lý khối cảnh báo	22
Hình 5.6 Sơ đồ nguyên lý nút nhấn	22
Hình 5.7 Module ESP32-C3 Super Mini	23
Hình 5.8 Hình sơ đồ nối chân module ESP32-C3 Super Mini	24
Hình 5.9 Sơ đồ nguyên lý khối xử lý trung tâm	25
Hình 5.10 Hình ảnh IC LM7805R	25
Hình 5.11 Các chân của IC LM7805R	26
Hình 5.12 Sơ đồ nguyên lý khối nguồn	27
Hình 5.13 Sơ đồ nguyên lý hệ thống	28
Hình 6.1 Lưu đồ giải thuật toàn hệ thống	29
Hình 6.2 Lưu đồ giải thuật tìm xe (trước khi nhận được phản hồi)	32
Hình 6.3 Lưu đồ giải thuật tìm xe (sau khi nhận được phản hồi)	33
Hình 6.4 Lưu đồ giải thuật phát hiện va chạm	34
Hình 6.5 Lưu đồ giải thuật xử lý va chạm và gửi cảnh báo	35
Hình 7.1 Thiết bị phần cứng sau khi hoàn thiện	37
Hình 7.2 Hộp nhựa ABS (55x35x1ml) bảo vệ phần cứng	37
Hình 7.3 Giao diện trang chủ khi truy cập vào ứng dụng	39
Hình 7.4 Giao diện quản lý người thân	40
Hình 7.5 Giao diện cấu hình thông báo	41
Hình 7.6 Giao diện cài đặt	42

Danh sách bảng

Bảng 5.1 Thông số kỹ thuật cảm biến gia tốc MPU6050	18
Bảng 5.2 Chức năng các chân cảm biến gia tốc MPU6050	18
Bảng 5.3 Thông số kỹ thuật của Buzzer 12x9mm - 93dB	20
Bảng 5.4 Chức năng các chân của Buzzer 12x9mm – 93dB.....	21
Bảng 5.5 Thông số kỹ thuật module ESP32-C3 Super Mini.....	23
Bảng 5.6 Thông số kỹ thuật của LM7805R	26
Bảng 5.7 Chức năng các chân của LM7805R	27

Danh mục các từ viết tắt

[I2C]	Inter-Integrated Circuit.
[GPS]	Global Positioning System
[BLE]	Bluetooth Low Energy
[IOT]	Internet of things
[SDA]	Serial Data
[SCL]	Serial Clock
[GPIO]	General purpose input/output
[UART]	Universal Asynchronous Receiver-Transmitter
[SPI]	Serial Peripheral Interface

1 Giới thiệu

Tai nạn giao thông, đặc biệt là tai nạn liên quan đến xe máy, vẫn đang là một thách thức lớn đối với nhiều quốc gia đang phát triển, trong đó có Việt Nam. Với tỷ lệ sử dụng xe máy chiếm đa số cùng hệ thống giao thông phức tạp, người điều khiển phương tiện này luôn đối mặt với nguy cơ tai nạn cao. Trong nhiều trường hợp, sau va chạm, nạn nhân không thể kịp thời gọi cứu trợ do bị bất tỉnh hoặc do sự cố xảy ra tại những khu vực vắng vẻ, dẫn đến việc chậm trễ trong công tác ứng cứu và giảm khả năng sống sót.

Bên cạnh vấn đề an toàn giao thông, một tình huống thường gặp khác là việc người dùng gặp khó khăn khi tìm kiếm xe trong các bãi đỗ xe đông đúc, tầng hầm hoặc trung tâm thương mại – nơi tầm nhìn bị hạn chế và các phương tiện có thiết kế tương tự nhau, gây mất thời gian và bất tiện trong quá trình tìm xe máy, đòi hỏi cần có sự hỗ trợ để định hướng khu vực đỗ xe.

Từ những vấn đề thực tiễn nêu trên, nhóm đề xuất thực hiện dự án “Thiết kế và phát triển hệ thống phát hiện va chạm và gửi cảnh báo khẩn cấp, tích hợp tìm xe trong bãi bằng BLE”. Thiết bị được xây dựng dựa trên sự tích hợp giữa cảm biến phát hiện va chạm, GPS trên điện thoại, còi buzzer và vi điều khiển, nhằm tạo ra một giải pháp hỗ trợ người lái xe máy nâng cao mức độ an toàn, đồng thời mang lại sự tiện lợi trong quá trình sử dụng hàng ngày.

2 Mục tiêu và yêu cầu kỹ thuật

2.1 Nhu cầu của người dùng

Trong cuộc sống hàng ngày, người điều khiển xe máy đặc biệt là người lớn tuổi, học sinh và những người thường xuyên di chuyển một mình nhưng không phải lúc nào cũng xảy ra tai nạn ở những nơi đông người, có người chứng kiến hay dễ dàng trong việc gọi trợ giúp. Chính vì vậy, họ luôn cần một giải pháp hỗ trợ an toàn và có tính chủ động cao, không cần phụ thuộc vào bản thân hay môi trường xung quanh; Họ cần một hệ thống có thể thay họ ghi nhận sự cố và thông báo cho người thân kịp thời, từ đó giảm đi những rủi ro không đáng có, giúp họ tận dụng được “khoảng thời gian vàng” sau va chạm.

Không dừng lại ở việc phát hiện va chạm đơn thuần, nhu cầu thực tế còn đòi hỏi thiết bị phải có khả năng tương tác nhanh với điện thoại, gửi thông báo khẩn cấp cho người thân mà không cần hệ thống xử lý phức tạp, rườm rà. Ngoài ra, yếu tố nhỏ gọn, dễ dàng triển khai cũng như không yêu cầu người dùng phải thao tác nhiều hay phải liên tục kích hoạt mỗi lần sử dụng. Đó cũng chính là những tiêu chí mà người dùng thực sự quan tâm.

Khách hàng ngày nay luôn có xu hướng tin tưởng các giải pháp tích hợp công nghệ nhẹ, tiết kiệm năng lượng, nhưng vẫn đáp ứng được yêu cầu về độ tin cậy và chính xác. Một hệ thống như vậy không chỉ nâng cao mức độ an toàn mà còn mang lại cảm giác yên tâm cho cả người sử dụng lẫn người thân của họ. Từ đó có thể khẳng định rằng nhu cầu thiết kế một thiết bị phát hiện va chạm, cảnh báo khẩn cấp và hỗ trợ theo dõi tình trạng an toàn theo thời gian thực là hoàn toàn hợp lý, xuất phát từ nhu cầu sống còn và thực tiễn của người sử dụng xe máy trong xã hội hiện nay.

2.2 Yêu cầu kỹ thuật

2.2.1 Chức năng

Phát hiện va chạm hoặc té ngã: Thiết bị sử dụng cảm biến gia tốc để nhận biết các chuyển động bất thường ví dụ như va chạm xe máy hoặc té ngã.

Gửi cảnh báo khẩn cấp: Khi phát hiện sự cố, hệ thống tự động gửi thông báo cảnh báo đến điện thoại người thân.

Xác định và gửi vị trí hiện tại: Thông qua GPS điện thoại, hệ thống gửi tọa độ vị trí xảy ra sự cố kèm theo thông báo cảnh báo.

Khả năng tìm kiếm: Thiết bị có thể phát tín hiệu âm thanh, giúp người dùng dễ dàng xác định vị trí xe trong phạm vi gần (~80m) bằng BLE.

Tự động hoạt động: Không yêu cầu thao tác thủ công từ người dùng sau khi thiết bị được kích hoạt.

2.2.2 Phi chức năng

Phản hồi nhanh: Hệ thống xử lý và gửi cảnh báo trong vòng 4-5 giây sau khi xảy ra va chạm.

Tiêu thụ năng lượng thấp: Thiết bị có thể hoạt động liên tục nhờ bình ắc-quy của xe máy.

Tính ổn định cao: Hoạt động bền bỉ trong môi trường di chuyển thực tế, ít lỗi, hạn chế cảnh báo sai.

Tính di động: Thiết bị nhỏ gọn, dễ mang theo hoặc gắn lên phương tiện mà không gây vướng víu.

Thân thiện với người dùng: Thiết kế đơn giản, không yêu cầu kỹ năng công nghệ khi sử dụng.

3 Lý thuyết cơ bản

3.1 Flutter

Flutter là nền tảng mã nguồn mở do Google phát triển, dùng để tạo ứng dụng đa nền tảng (Android, iOS, Web, Desktop) từ một mã nguồn duy nhất bằng ngôn ngữ Dart [1].

Đặc điểm:

- Đa nền tảng: Viết một lần, chạy được trên nhiều hệ điều hành như Android, iOS, Web, Windows, macOS, ...
- Sử dụng Dart: Ngôn ngữ lập trình do Google phát triển, dễ học, hướng đối tượng, hiệu suất cao.
- Giao diện tùy biến: Hệ thống widget mạnh mẽ, hỗ trợ Material Design (Android) và Cupertino (iOS).
- Hiệu suất cao: Flutter biên dịch trực tiếp sang mã máy (native), cho tốc độ xử lý nhanh và mượt.
- Hot reload: Cho phép xem thay đổi trong mã nguồn gần như ngay lập tức khi lập trình.

Ưu điểm:

- Phát triển nhanh, giao diện đẹp.
- Tiết kiệm chi phí với một codebase duy nhất.
- Hiệu suất gần tương đương ứng dụng native.
- Cộng đồng lớn, nhiều plugin hỗ trợ sẵn.

Nhược điểm:

- Kích thước ứng dụng thường lớn hơn native.
- Cần học thêm ngôn ngữ Dart.

- Một số chức năng native mới có thể chưa hỗ trợ đầy đủ.

Nếu bạn cần mở rộng sang ứng dụng Flutter trong các lĩnh vực như IoT, giao tiếp BLE/SPI, mình có thể cung cấp thêm.

3.2 Hệ thống định vị toàn cầu (GPS)

Hệ thống định vị toàn cầu (GPS) là hệ thống điều hướng dựa trên vệ tinh được tạo thành từ ít nhất 24 vệ tinh. GPS hoạt động trong mọi điều kiện thời tiết, mọi nơi trên thế giới, 24 giờ mỗi ngày, không mất phí thuê hoặc phí thiết lập sử dụng GPS [3]. Vệ tinh GPS xoay quanh Trái Đất hai lần một ngày trong một quỹ đạo chính xác. Mỗi vệ tinh truyền một tín hiệu nhất định và các tham số quỹ đạo cho phép các thiết bị GPS giải mã và tính toán vị trí chính xác của vệ tinh. Máy thu GPS sử dụng thông tin này và phép tam giác để tính toán vị trí chính xác của người dùng.

Đặc điểm:

- Định vị toàn cầu: Sử dụng tối thiểu 4 vệ tinh để xác định chính xác vị trí 3D (vĩ độ, kinh độ, độ cao).
- Được phát triển bởi Bộ Quốc phòng Mỹ, nhưng được sử dụng phổ biến trong dân sự.
- Hoạt động mọi nơi: Trên đất liền, trên biển, trên không – miễn là thiết bị "nhìn thấy" bầu trời.
- Không cần kết nối Internet: GPS sử dụng sóng vô tuyến từ vệ tinh, không phụ thuộc vào mạng di động hay Wi-Fi.
- Cập nhật thời gian thực: Cho phép đo tốc độ di chuyển, hướng đi và thời gian với độ chính xác cao.

Nguyên lý hoạt động:

1. Thiết bị GPS nhận tín hiệu từ nhiều vệ tinh trên quỹ đạo.
2. Tính toán khoảng cách đến từng vệ tinh dựa vào độ trễ tín hiệu.

3. Tam giác hóa vị trí: Từ khoảng cách đến ít nhất 4 vệ tinh, thiết bị xác định chính xác vị trí hiện tại.

Ưu điểm:

- Độ chính xác cao (từ vài mét đến dưới 1 mét với thiết bị tốt).
- Phổ biến, miễn phí sử dụng.
- Không cần kết nối mạng hoặc nhà cung cấp dịch vụ riêng.

Nhược điểm:

- Không hoạt động tốt trong nhà, hầm, khu vực bị che khuất.
- Cần thời gian để "bắt tín hiệu" khi vừa khởi động (cold start).
- Tốn pin nếu sử dụng liên tục trong thiết bị di động.

3.3 Các chuẩn truyền dữ liệu

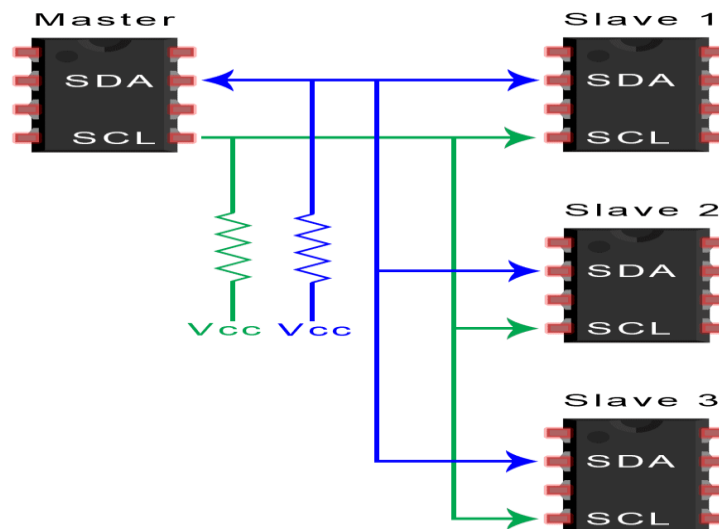
3.3.1 Chuẩn giao tiếp I2C

I2C (Inter-integrated-circuit) là giao thức truyền thông nối tiếp đồng bộ, nó không được sử dụng cho giao tiếp PC- thiết bị mà thay vào đó là với các mô-đun và cảm biến [4]. Đây là giao thức phổ biến trong các hệ thống nhúng để kết nối với các thiết bị điện tử với nhau trên cùng 1 bo mạch.

Đặc điểm

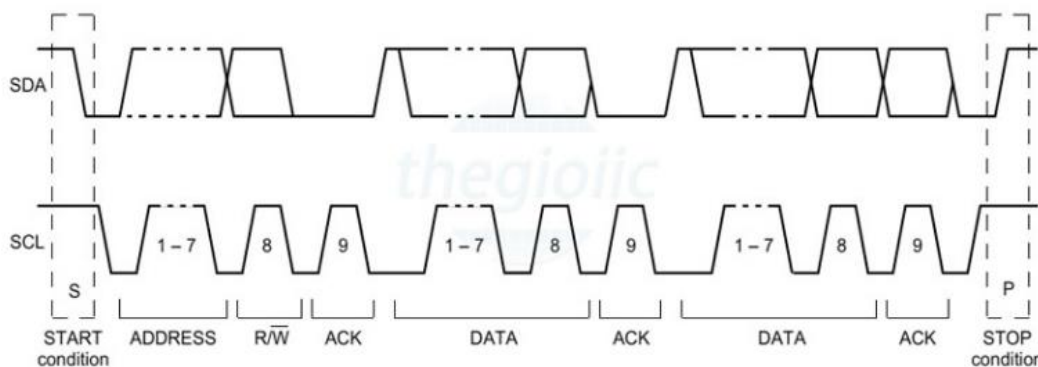
- Cấu trúc vật lý : I2C sử dụng hai đường truyền tín hiệu. Cả hai dây đều dạng open-drain và cần điện trở kéo lên.
- SDA (Serial Data): Đảm nhiệm việc truyền và nhận dữ liệu
- SCL (Serial Clock): Tạo xung nhịp do Master phát ra
- Truyền thông đồng bộ: Sử dụng tín hiệu xung clock (SCL) để đồng bộ hóa quá trình truyền dữ liệu. Thiết bị Master điều khiển xung clock
- Hỗ trợ nhiều thiết bị: Cho phép kết nối nhiều thiết bị (tối đa 127 thiết bị) trên cùng 1 bus. Mỗi thiết bị có 1 địa chỉ riêng

- Mô hình Master – Slave : Thiết bị Master khởi tạo và kiểm soát quá trình truyền thông



Hình 3.1 Mô tả chuẩn truyền I2C

Quá trình truyền/nhận dữ liệu:



Hình 3.2 Minh họa quá trình truyền/nhận dữ liệu trong giao thức I2C

Quá trình truyền/nhận dữ liệu bắt đầu khi Master khởi động bằng cách lần lượt đưa các đường SDA và SCL từ mức cao (1) xuống mức thấp (0).

Master gửi cho mỗi slave một địa chỉ 7 hoặc 10 bit của slave và 1 bit đọc/ghi tới slave mà nó muốn giao tiếp. Sau đó, slave sẽ so sánh địa chỉ với địa chỉ của chính nó. Nếu địa chỉ trùng khớp, Slave xác nhận bằng cách kéo SDA xuống mức thấp và đặt bit ACK thành '0'. Nếu không trùng khớp, cả đường SDA và bit ACK/NACK đều mặc định ở mức cao.

Sau đó, máy chủ sẽ gửi hoặc nhận khung dữ liệu. Nếu Master gửi dữ liệu đến Slave, bit Read/Write sẽ được đặt ở mức 0. Ngược lại, nếu Master nhận dữ liệu từ Slave, bit này sẽ được đặt ở mức 1. Nếu khung dữ liệu được truyền thành công, bit ACK/NACK sẽ được đặt thành 0 để thông báo cho Master tiếp tục truyền dữ liệu.

Để dừng truyền dữ liệu, master gửi tín hiệu dừng đến slave bằng cách chuyển SCL lên cao trước khi chuyển SDA lên cao.

Ưu điểm:

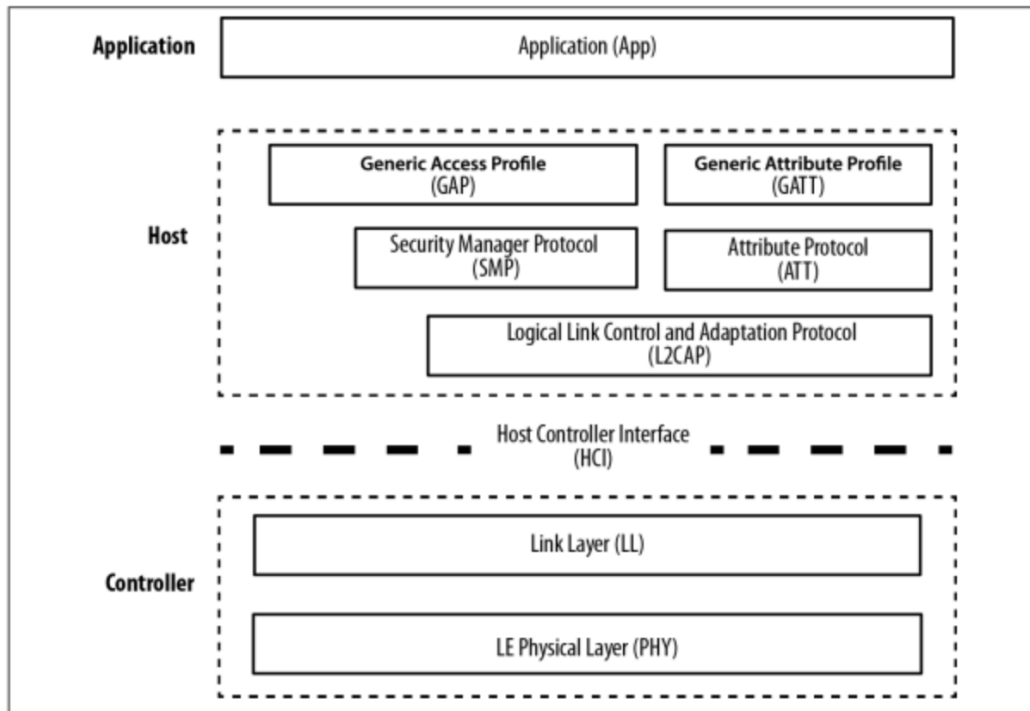
- Một trong những lợi thế lớn nhất của I2C là nó chỉ cần hai dây để truyền và nhận dữ liệu: SCL (Serial Clock Line) và SDA (Serial Data Line). Điều này giúp giảm số lượng dây cần thiết trong các mạch phức tạp, làm cho việc thiết kế và quản lý dây dễ dàng hơn.
- Lý tưởng cho truyền thông với khoảng cách ngắn và nhiều thiết bị.
- I2C sử dụng bit ACK (acknowledge) và NACK (not acknowledge) để xác nhận rằng mỗi khung dữ liệu hoặc địa chỉ được gửi hoặc nhận thành công. Điều này giúp đảm bảo tính chính xác và đáng tin cậy trong quá trình truyền dữ liệu.

Nhược điểm:

- Tốc độ truyền dữ liệu chậm hơn so với giao thức SPI, có thể là vấn đề trong các ứng dụng yêu cầu truyền dữ liệu tốc độ cao.
- Bị giới hạn về khoảng cách truyền, thường chỉ hiệu quả trong khoảng cách ngắn (thường dưới 1 mét).
- Bị giới hạn về số lượng thiết bị có thể kết nối (tối đa 128 thiết bị với địa chỉ 7-bit hoặc 1024 thiết bị với địa chỉ 10-bit).
- Kích thước của khung dữ liệu trong I2C được giới hạn ở 8 bit.
- Cần điện trở kéo lên (pull-up) trên cả hai đường tín hiệu, làm tăng mức tiêu thụ năng lượng.

3.3.2 Chuẩn giao tiếp không dây Bluetooth

BLE (Bluetooth Low Energy) là giao thức truyền thông không dây tầm ngắn, được thiết kế cho các thiết bị tiêu thụ năng lượng thấp như cảm biến, thiết bị đeo, thiết bị IoT [5].



Hình 3.3 Các thành phần bên trong bộ giao thức BLE

Đặc điểm:

- Giao tiếp không dây: BLE hoạt động ở tần số 2.4 GHz ISM, truyền dữ liệu qua sóng vô tuyến mà không cần dây nối.
- Mô hình Central – Peripheral: Một thiết bị Central (thường là điện thoại hoặc máy tính) có thể kết nối với một hoặc nhiều thiết bị Peripheral (ví dụ cảm biến, đồng hồ thông minh...). Central quản lý quá trình kết nối và giao tiếp.
- Truyền dữ liệu theo gói: Dữ liệu được chia thành các gói nhỏ và truyền qua giao thức GATT (Generic Attribute Profile).
- Tốc độ thấp: Tốc độ truyền tối đa khoảng 1 Mbps (hoặc 2 Mbps với BLE 5.0), thấp hơn nhiều so với các giao thức có dây như SPI hoặc UART.
- Tiết kiệm năng lượng: Các thiết bị BLE phần lớn thời gian ở chế độ ngủ, chỉ hoạt động khi có sự kiện truyền dữ liệu, giúp kéo dài tuổi thọ pin.

Quá trình truyền/nhận dữ liệu

Trong giao tiếp BLE, thiết bị Central sẽ chủ động tìm kiếm các thiết bị Peripheral xung quanh và gửi yêu cầu kết nối.

- Kết nối: Sau khi nhận được tín hiệu quảng bá (advertising) từ Peripheral, Central gửi yêu cầu kết nối.
- Giao tiếp: Dữ liệu được truyền thông qua các đặc tính (characteristics) trong cấu trúc dịch vụ (services) của thiết bị, dựa trên giao thức GATT.
- Notification và Indication: Peripheral có thể gửi dữ liệu đến Central bằng thông báo (notification, không yêu cầu phản hồi) hoặc chỉ báo (indication, có yêu cầu phản hồi).
- Quản lý năng lượng: Khi không có dữ liệu cần truyền, thiết bị chuyển sang chế độ tiết kiệm năng lượng.

Ưu điểm

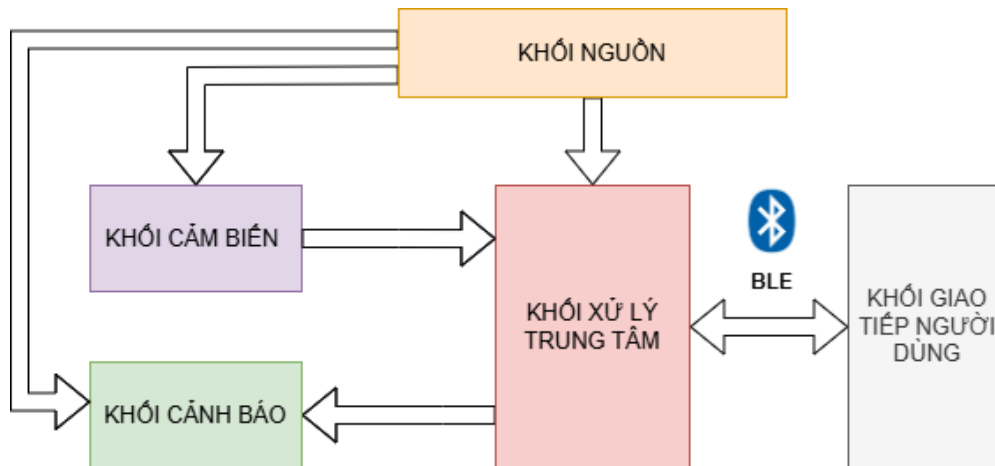
- Tiêu thụ năng lượng rất thấp, thích hợp cho các thiết bị dùng pin nhỏ.
- Không cần kết nối dây, thuận tiện cho các thiết bị di động và đeo tay.
- Cho phép một thiết bị Central kết nối với nhiều thiết bị Peripheral cùng lúc.
- Tương thích rộng rãi với hầu hết smartphone và máy tính hiện đại.

Nhược điểm

- Tốc độ truyền dữ liệu thấp, không phù hợp với các ứng dụng yêu cầu băng thông cao như truyền video hoặc âm thanh.
- Có độ trễ cao hơn giao tiếp có dây như SPI hoặc UART.
- Chất lượng kết nối có thể bị ảnh hưởng trong môi trường có nhiều nhiễu sóng vô tuyến.

4 Kiến trúc hệ thống

4.1 Sơ đồ khối hệ thống



Hình 4.1 Sơ đồ khối hệ thống

4.2 Chi tiết chức năng từng khối

Khối nguồn: Cung cấp điện áp ổn định 5V (chuyển đổi từ 12V của xe máy) cho các khối khác hoạt động ổn định. Khối nguồn cần có chức năng bảo vệ quá áp, quá dòng và giảm tiêu hao năng lượng khi xe không hoạt động.

Khối xử lý trung tâm: Là bộ não điều khiển chính của hệ thống, có nhiệm vụ thu thập, phân tích và xử lý dữ liệu từ các khối khác như cảm biến, giao tiếp không dây và người dùng. Đóng vai trò quyết định, kết nối và điều phối hoạt động của toàn bộ hệ thống.

Khối cảm biến: Chịu trách nhiệm thu thập thông tin từ môi trường. Dữ liệu từ khối cảm biến được gửi về khối xử lý trung tâm để phân tích và ra xử lý.

Khối cảnh báo: Thực hiện các chức năng cảnh báo, báo động ra môi trường xung quanh khi phát hiện các tình huống bất thường như va chạm, ngã xe, hoặc khi người dùng kích hoạt tìm xe thông qua ứng dụng điện thoại.

Khối giao tiếp người dùng: Nơi tiếp nhận và hiển thị dữ liệu đã được xử lý từ khối xử lý trung tâm, giao tiếp thông qua khối giao tiếp không dây, đồng thời tương tác để thực hiện các chức năng trên hệ thống.

Confidential Property of CCE Dept.

5 Phân tích và thiết kế các khối

5.1 Khối cảm biến

a) Lựa chọn linh kiện

MPU6050 là cảm biến 6 trục gồm 3 trục gia tốc (Accelerometer) và 3 trục con quay hồi chuyển (Gyroscope), giúp xác định phương hướng và chuyển động. Giao tiếp với vi điều khiển thông qua chuẩn I2C, sử dụng phổ biến trong drone, phát hiện va chạm, robot cân bằng...



Hình 5.1 Cảm biến gia tốc MPU6050

Bảng 5.1 Thông số kỹ thuật cảm biến gia tốc MPU6050

Thông số	Giá trị
Gia tốc kế	$\pm 2g$ đến $\pm 16g$
Con quay hồi chuyển	± 250 đến ± 2000 °/s
Tần số lấy mẫu	Tối đa 1 kHz
Giao tiếp	I2C
Điện áp hoạt động	3.3V – 5V

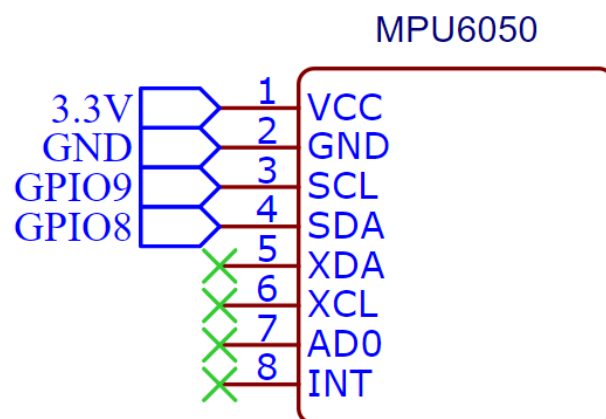
Bảng 5.2 Chức năng các chân cảm biến gia tốc MPU6050

Thông số	Đặc điểm
VCC	Cấp nguồn (3.3V hoặc 5V)
GND	Nối đất
SCL	Giao tiếp I2C
SDA	Giao tiếp I2C
XDA	Để hở hoặc nối GND
XCL	Để hở hoặc nối GND
AD0	Nối GND
INT	Chân ngắt (nối hoặc không)

Lý do chọn linh kiện: MPU6050 được chọn vì tích hợp cả gia tốc kế và con quay hồi chuyển 3 trục trong một chip, giúp phát hiện chuyển động và va chạm chính xác. Cảm biến sử dụng giao tiếp I2C, tương thích tốt với ESP32 và có nhiều thư viện hỗ trợ, dễ lập trình. MPU6050 cho phép tùy chỉnh dải đo rộng, độ nhạy cao, tiêu thụ điện thấp và giá thành rẻ, rất phù hợp để triển khai sản phẩm phần cứng đơn giản trong môi trường học thuật.

b) Sơ đồ nguyên lý khối cảm biến

Trong hình 5.2, khối cảm biến MPU6050 được kết nối với vi điều khiển ESP32-C3-SUPER-MINI thông qua giao tiếp I2C. Cụ thể, nguồn 3.3V từ ESP32 được cấp vào chân VCC của MPU6050, trong khi chân GND của cả hai thiết bị được nối chung để tạo mức điện áp tham chiếu. Để truyền nhận dữ liệu, chân GPIO8 của ESP32 được kết nối với chân SDA (Serial Data) của cảm biến, và chân GPIO9 kết nối với chân SCL (Serial Clock). Đây là hai đường tín hiệu chính của giao thức I2C, cho phép ESP32 đọc dữ liệu gia tốc và con quay hồi chuyển từ MPU6050. Việc sử dụng GPIO8 và GPIO9 cho I2C là lựa chọn phù hợp vì ESP32 hỗ trợ cấu hình linh hoạt các chân GPIO cho nhiều chức năng khác nhau, và mức điện áp 3.3V của ESP32 tương thích hoàn toàn với yêu cầu của MPU6050.

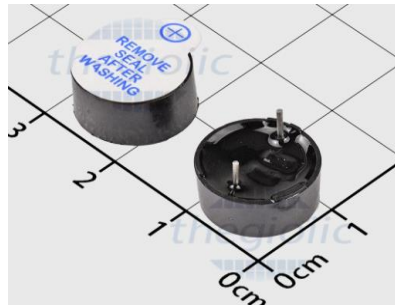


Hình 5.2 Sơ đồ nguyên lý khối cảm biến MPU6050

5.2 Khối cảnh báo

a) Lựa chọn linh kiện

Buzzer 12x9mm – 93dB là một thiết bị phát âm thanh nhỏ gọn, thường được sử dụng để tạo tín hiệu cảnh báo bằng âm thanh trong các mạch điện tử. Với kích thước tiêu chuẩn 12mm đường kính và chiều cao 9mm, thiết bị dễ dàng lắp đặt trên các bo mạch thông qua hai chân cắm xuyên lỗ. Cường độ âm thanh lên đến 93dB giúp đảm bảo tín hiệu cảnh báo rõ ràng, ngay cả trong môi trường có độ ồn vừa phải.



Hình 5.3 Hình ảnh Buzzer 12x9mm – 93dB

Bảng 5.3 Thông số kỹ thuật của Buzzer 12x9mm - 93dB

Cách gắn	Xuyên lỗ
Điện áp định mức	5 V dc
Cường độ âm thanh	93 dB
Loại Drive	Trong
Loại Tone	Liên tục
Kích thước	12 x 9mm
Nhiệt độ tối thiểu	-20°C
Nhiệt độ tối đa	+70°C
Dòng điện	22mA
Tần số	2300Hz



Hình 5.4 Sơ đồ các chân của Buzzer 12x9mm – 93dB

Chức năng các chân được thể hiện trong bảng 5.4.

Bảng 5.4 Chức năng các chân của Buzzer 12x9mm – 93dB

Chân	Chức năng
Chân 1	Chân dương (Vcc) – Nối vào nguồn hoặc tín hiệu PWM
Chân 2	Chân âm (GND) – Nối đất

Trong việc cảnh báo và hỗ trợ tìm xe, việc thông báo bằng âm thanh là phương thức trực quan và hiệu quả để thu hút sự chú ý của người dùng. Buzzer 12x9mm – 93dB được lựa chọn nhờ vào thiết kế nhỏ gọn, dễ dàng tích hợp trên các mạch in, cùng với cường độ âm thanh lớn (lên đến 93dB) – đủ để người dùng nhận biết dễ dàng trong môi trường đông người hoặc ồn ào như bãi giữ xe.

b) Sơ đồ nguyên lý khối cảnh báo

Khối cảnh báo trong sơ đồ này có chức năng phát tín hiệu cảnh báo bằng âm thanh và ánh sáng, bao gồm hai thành phần chính là đèn LED và còi buzzer.

- **LED cảnh báo**

Đối với LED cảnh báo, chân GPIO4 của ESP32 được kết nối với anốt của LED thông qua một điện trở hạn dòng 220Ω. Điện trở này có vai trò bảo vệ LED khỏi dòng điện quá lớn. Catốt của LED được nối với chân GND.

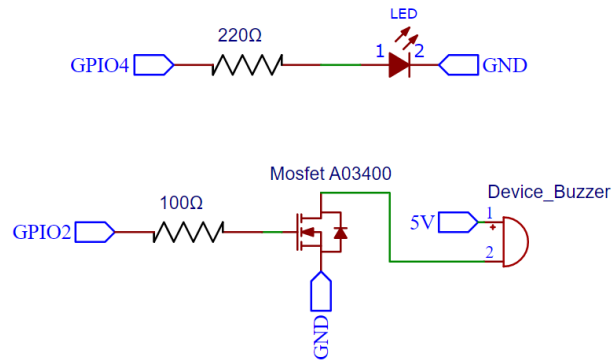
Nguyên lý hoạt động: Khi GPIO4 xuất mức cao (3.3V), dòng điện sẽ chạy qua điện trở và LED, làm LED phát sáng. Khi GPIO4 xuất mức thấp (0V), LED sẽ tắt.

- **Buzzer**

Với còi buzzer, mạch phức tạp hơn do nhóm em sử dụng MOSFET A03400 làm công tắc điện tử. Buzzer được cấp nguồn 5V trực tiếp từ chân 5V của ESP32 vào chân dương. Chân âm của buzzer nối với cực drain của MOSFET. Cực source của MOSFET nối GND. GPIO2 điều khiển cực gate của MOSFET thông qua điện trở 100Ω.

Nguyên lý hoạt động: Khi GPIO2 xuất mức cao 3.3V, điện áp này làm MOSFET dẫn điện, tạo đường dẫn từ chân âm buzzer xuống GND. Lúc này mạch điện hoàn chỉnh: 5V → buzzer → MOSFET → GND, buzzer phát âm thanh. Khi GPIO2 xuất mức thấp, MOSFET ngắt, mạch điện bị hở, buzzer im lặng.

Tóm lại, MOSFET đóng vai trò như công tắc điện tử, cho phép tín hiệu điều khiển 3.3V từ GPIO điều khiển được buzzer hoạt động ở 5V với dòng tiêu thụ cao hơn khả năng của GPIO.

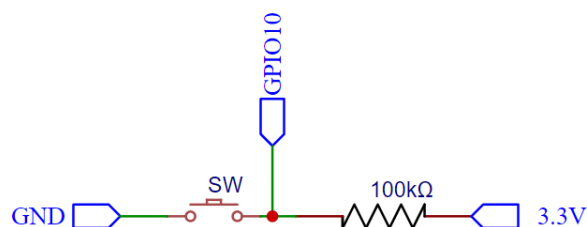


Hình 5.5 Sơ đồ nguyên lý khối cảnh báo

5.3 Nút nhấn – Reset hệ thống

Một đầu của nút nhấn được nối với GND, còn đầu kia kết nối với GPIO10 của ESP32. Đồng thời, GPIO10 cũng được nối với nguồn 3.3V thông qua một điện trở pull-up 100kΩ. Điện trở này có vai trò quan trọng trong việc đảm bảo tín hiệu ổn định.

Nguyên lý hoạt động: Khi nút nhấn không được nhấn (trạng thái mở), GPIO10 được kéo lên mức cao (3.3V) nhờ điện trở pull-up. Dòng điện rất nhỏ chảy từ nguồn 3.3V qua điện trở 100kΩ đến GPIO10, tạo ra mức logic 1. Khi người dùng nhấn nút (trạng thái đóng), GPIO10 được nối trực tiếp với GND thông qua nút nhấn. Lúc này, GPIO10 sẽ có mức thấp (0V), tạo ra mức logic 0. Điện trở 100kΩ giới hạn dòng điện từ nguồn 3.3V xuống GND, tránh gây ngắn mạch.

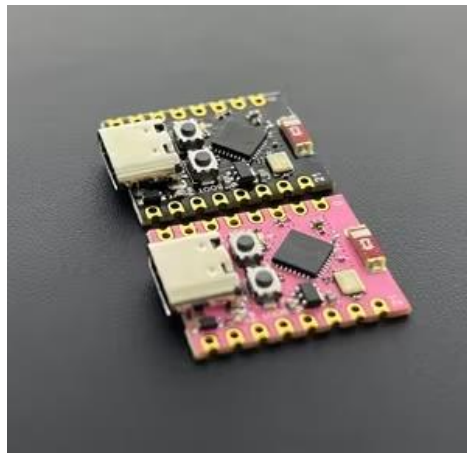


Hình 5.6 Sơ đồ nguyên lý nút nhấn

5.4 Khối xử lý trung tâm

a) Lựa chọn linh kiện

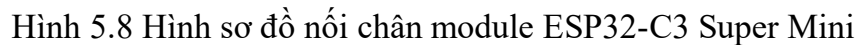
ESP32-C3 là một vi điều khiển tích hợp Wi-Fi và Bluetooth Low Energy (BLE) dựa trên kiến trúc RISC-V, phù hợp cho các ứng dụng IoT yêu cầu kết nối không dây và tiêu thụ năng lượng thấp. Phiên bản Super Mini có kích thước nhỏ gọn, dễ tích hợp vào các mô hình thiết bị phần cứng.



Hình 5.7 Module ESP32-C3 Super Mini

Bảng 5.5 Thông số kỹ thuật module ESP32-C3 Super Mini

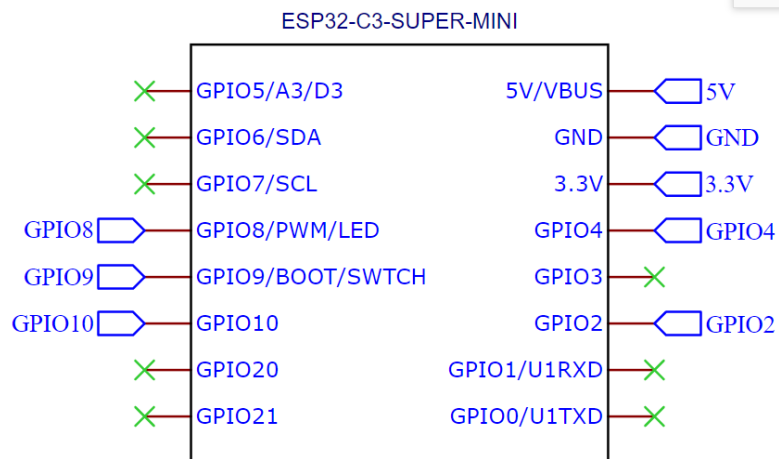
Đặc điểm	Thông số
Tốc độ xử lý	160 MHz
RAM	400 KB
Giao tiếp	I2C, PWM, UART
Số chân ADC	6 kênh 12-bit
Kết nối	Wi-Fi 2.4GHz, BLE 5.0
Điện áp hoạt động	3.0 – 3.6V



b) Sơ đồ nguyên lý khối xử lý trung tâm

Các kết nối cụ thể từ ESP32 đến các khối khác bao gồm: GPIO8 và GPIO9 được sử dụng cho giao tiếp I2C với cảm biến MPU6050, trong đó GPIO8 làm SDA (dữ liệu) và GPIO9 làm SCL (xung nhịp). GPIO4 điều khiển LED cảnh báo và GPIO2 điều khiển buzzer thông qua MOSFET. Đặc biệt, GPIO10 được thiết kế kết nối với nút nhấn trong khối giao tiếp người dùng, cho phép người dùng tương tác với hệ thống.

Tóm lại, tất cả các kết nối này cho phép ESP32 thu thập dữ liệu từ cảm biến, xử lý thông tin, phản hồi qua LED và buzzer, đồng thời nhận lệnh từ người dùng thông qua nút nhấn.

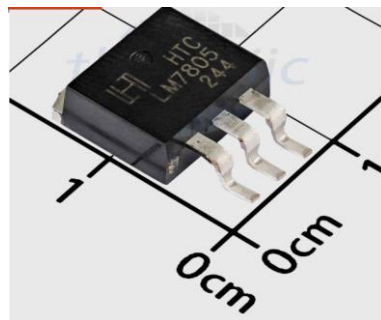


Hình 5.9 Sơ đồ nguyên lý khối xử lý trung tâm

5.5 Khối nguồn

a) Lựa chọn linh kiện

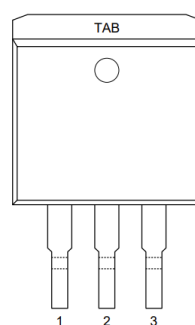
LM7805R là IC ổn áp tuyến tính đầu ra cố định 5V, dòng tối đa 1.5A, thường dùng để cấp nguồn ổn định cho vi điều khiển, cảm biến và các linh kiện điện tử khác. Nó nhận điện áp đầu vào cao hơn (thường 7–35V) và giữ đầu ra luôn ở mức 5V. LM7805R thuộc dạng vỏ TO-263 (SMD), phù hợp với mạch in dán bề mặt. Được sử dụng phổ biến trong các hệ thống nhúng, robot, mạch nguồn, thiết bị IoT...



Hình 5.10 Hình ảnh IC LM7805R

Bảng 5.6 Thông số kỹ thuật của LM7805R

Thuộc tính	Giá trị
Điện áp ngõ ra	5V
Dòng điện ngõ ra	1.5A
Điện áp ngõ vào Max	35V
Loại ổn áp	Low dropout voltage
Phân cực âm, dương	Điện áp dương
Loại ngõ ra	Cố định
Số ngõ ra	1
Nhiệt độ hoạt động	-40°C ~ 125°C
Kiểu chân	Dán bề mặt
Số chân	3
Kiểu đóng gói	TO-263



Hình 5.11 Các chân của IC LM7805R

Bảng 5.7 Chức năng các chân của LM7805R

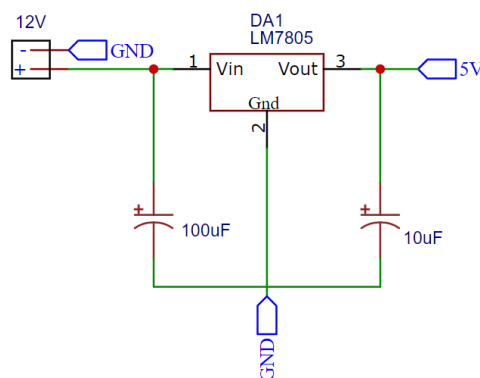
STT	Tên chân	Chức năng
1	INPUT	Input Voltage
2	GND	Ground
3	OUTPUT	Output Voltage

LM7805R được chọn vì đây là IC ổn áp 5V đơn giản, dễ dùng, cung cấp dòng lên đến 1.5A – phù hợp để cấp nguồn ổn định cho vi điều khiển và các linh kiện khác. Với dạng vỏ TO-263 nhỏ gọn, khả năng bảo vệ quá nhiệt, quá dòng, và độ nhiễu thấp, LM7805R đáp ứng tốt yêu cầu của mạch nguồn trong dự án.

b) Sơ đồ nguyên lý khối nguồn

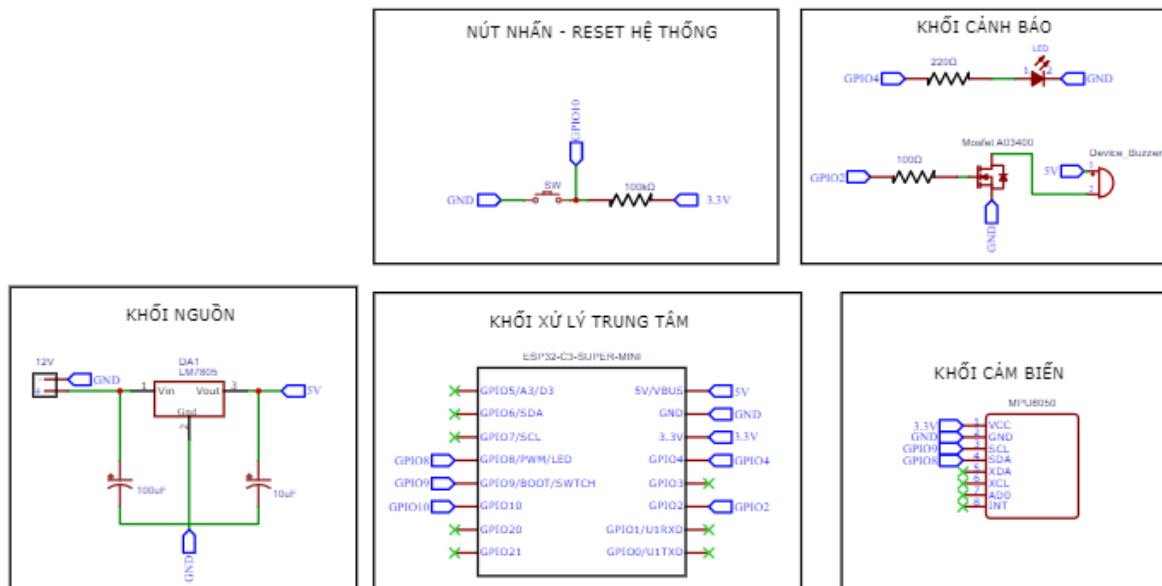
Nguồn 12V DC từ bình ắc quy được cấp vào đầu vào của mạch, với dây dương kết nối vào chân Vin của LM7805 và dây âm nối với GND chung của hệ thống. IC LM7805 là một bộ ổn áp tuyến tính ba chân, có khả năng chuyển đổi điện áp đầu vào từ 7-35V xuống điện áp đầu ra ổn định 5V.

Hai tụ điện 100 μ F được sử dụng trong mạch với vai trò khác nhau. Tụ đầu vào nối giữa chân Vin và GND giúp lọc nhiễu từ nguồn cấp, ổn định điện áp đầu vào và cung cấp dòng điện tức thời khi tải thay đổi đột ngột. Tụ đầu ra nối giữa chân Vout và GND có tác dụng lọc nhiễu đầu ra, giảm gợn sóng và cải thiện đáp ứng tức thời của mạch.



Hình 5.12 Sơ đồ nguyên lý khối nguồn

5.6 Sơ đồ khối nguyên lý phần cứng

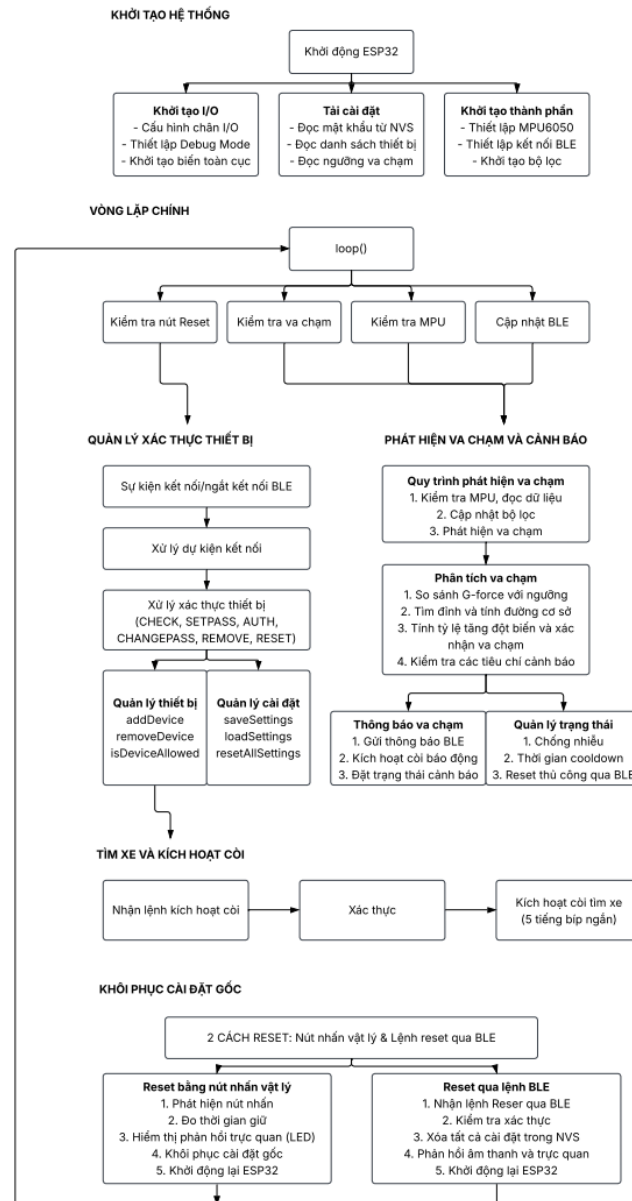


Hình 5.13 Sơ đồ nguyên lý hệ thống

Hình 5.13 trình bày sơ đồ nguyên lý hệ thống, gồm các khối chức năng chính: nút nhấn reset, khối cảnh báo, khối cảm biến, khối xử lý trung tâm và khối nguồn.

6 Lưu đồ giải thuật chức năng của hệ thống

6.1 Lưu đồ giải thuật hệ thống



Hình 6.1 Lưu đồ giải thuật toàn hệ thống

Hệ thống BikeGuard là một giải pháp bảo vệ xe máy toàn diện, được thiết kế để giải quyết hai vấn đề chính: tìm kiếm xe trong khu vực đông đúc và phát hiện va chạm khi không ở gần xe. Hệ thống được xây dựng trên nền tảng ESP32, tích hợp cảm biến gia tốc MPU6050 và giao tiếp với thiết bị di động qua Bluetooth Low Energy (BLE).

Khi khởi động, hệ thống bắt đầu với việc thiết lập các thành phần cơ bản như: ESP32 khởi tạo các chân I/O, cấu hình chế độ debug và thiết lập các biến toàn cục. Tiếp theo, hệ thống tải các cài đặt từ bộ nhớ không bay hơi (NVS), bao gồm mật khẩu, danh sách thiết bị đã được phép kết nối, và ngưỡng phát hiện va chạm. Đồng thời, ESP32 khởi tạo cảm biến MPU6050 và bộ lọc kỹ thuật số, thiết lập kết nối BLE và chuẩn bị bộ lọc tín hiệu để phát hiện va chạm chính xác.

Sau khi khởi tạo, hệ thống chuyển sang vòng lặp chính, liên tục thực hiện bốn nhiệm vụ quan trọng. Đầu tiên, kiểm tra nút reset vật lý, ghi nhận thời gian nhấn và cung cấp phản hồi trực quan qua LED. Thứ hai, kiểm tra va chạm thông qua dữ liệu từ cảm biến gia tốc. Thứ ba, giám sát trạng thái MPU và tự động khởi tạo lại nếu phát hiện lỗi. Cuối cùng, cập nhật các đặc tính BLE và duy trì kết nối với thiết bị di động.

Phần quản lý xác thực thiết bị là trọng tâm của hệ thống bảo mật BikeGuard. Khi có thiết bị kết nối, ESP32 bật LED báo hiệu và thiết lập trạng thái kết nối. Thông qua giao thức BLE, hệ thống xử lý các lệnh xác thực từ ứng dụng di động, bao gồm kiểm tra thiết bị (CHECK), đặt mật khẩu mới (SETPASS), xác thực (AUTH), thay đổi mật khẩu (CHANGEPASS), xóa thiết bị (REMOVE), và khôi phục cài đặt gốc (RESET). Hệ thống duy trì danh sách thiết bị được phép và cho phép thêm, xóa hoặc kiểm tra thiết bị, đồng thời lưu các cài đặt vào bộ nhớ không bay hơi để sử dụng sau khi khởi động lại.

Chức năng phát hiện va chạm hoạt động thông qua một quy trình ba bước. Đầu tiên, hệ thống đọc dữ liệu từ cảm biến MPU6050 và cập nhật bộ lọc trung bình di động để giảm nhiễu. Tiếp theo, thuật toán phát hiện va chạm phân tích dữ liệu bằng cách so sánh giá trị G-force với ngưỡng, tìm đỉnh và tính đường cơ sở, từ đó xác định tỷ lệ tăng đột biến. Nếu thỏa mãn các tiêu chí về cường độ, tỷ lệ tăng và thời gian, hệ thống xác nhận có va chạm và kích hoạt cảnh báo. Quá trình cảnh báo bao gồm gửi thông báo qua BLE đến thiết bị di động và kích hoạt còi

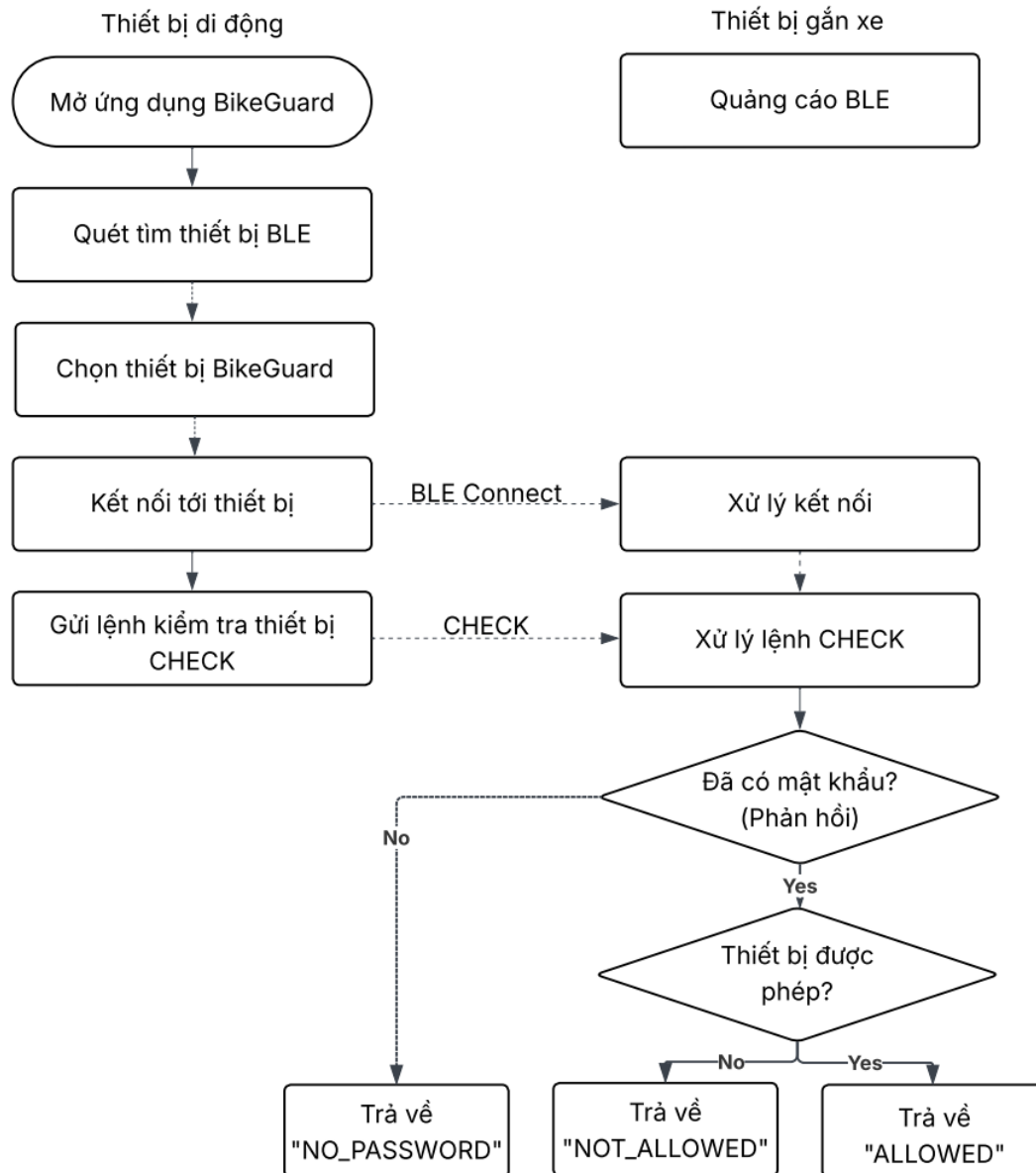
báo động tại chỗ. Hệ thống cũng thông minh trong việc tránh cảnh báo sai với cơ chế chống nhiễu và thời gian nghỉ giữa các lần cảnh báo.

Tính năng tìm xe cho phép người dùng kích hoạt còi báo từ xa thông qua ứng dụng di động. Khi nhận lệnh kích hoạt, ESP32 kiểm tra xem thiết bị đã được xác thực chưa, và nếu đã xác thực, sẽ phát ra 5 tiếng bíp ngắn để giúp người dùng định vị xe trong khu vực đông đúc.

Cuối cùng, BikeGuard hỗ trợ hai phương thức khôi phục cài đặt gốc: thông qua nút nhấn vật lý hoặc lệnh BLE. Khi người dùng giữ nút reset vật lý trên 3 giây, hoặc gửi lệnh RESET từ thiết bị đã xác thực, hệ thống sẽ xóa tất cả cài đặt trong bộ nhớ, hiển thị phản hồi âm thanh và trực quan, sau đó khởi động lại với cài đặt mặc định.

Nhờ kiến trúc rõ ràng và các thuật toán thông minh, BikeGuard cung cấp một giải pháp toàn diện để bảo vệ xe máy, dễ dàng sử dụng thông qua ứng dụng di động và đáng tin cậy trong các tình huống khẩn cấp.

6.2 Lưu đồ giải thuật tìm xe

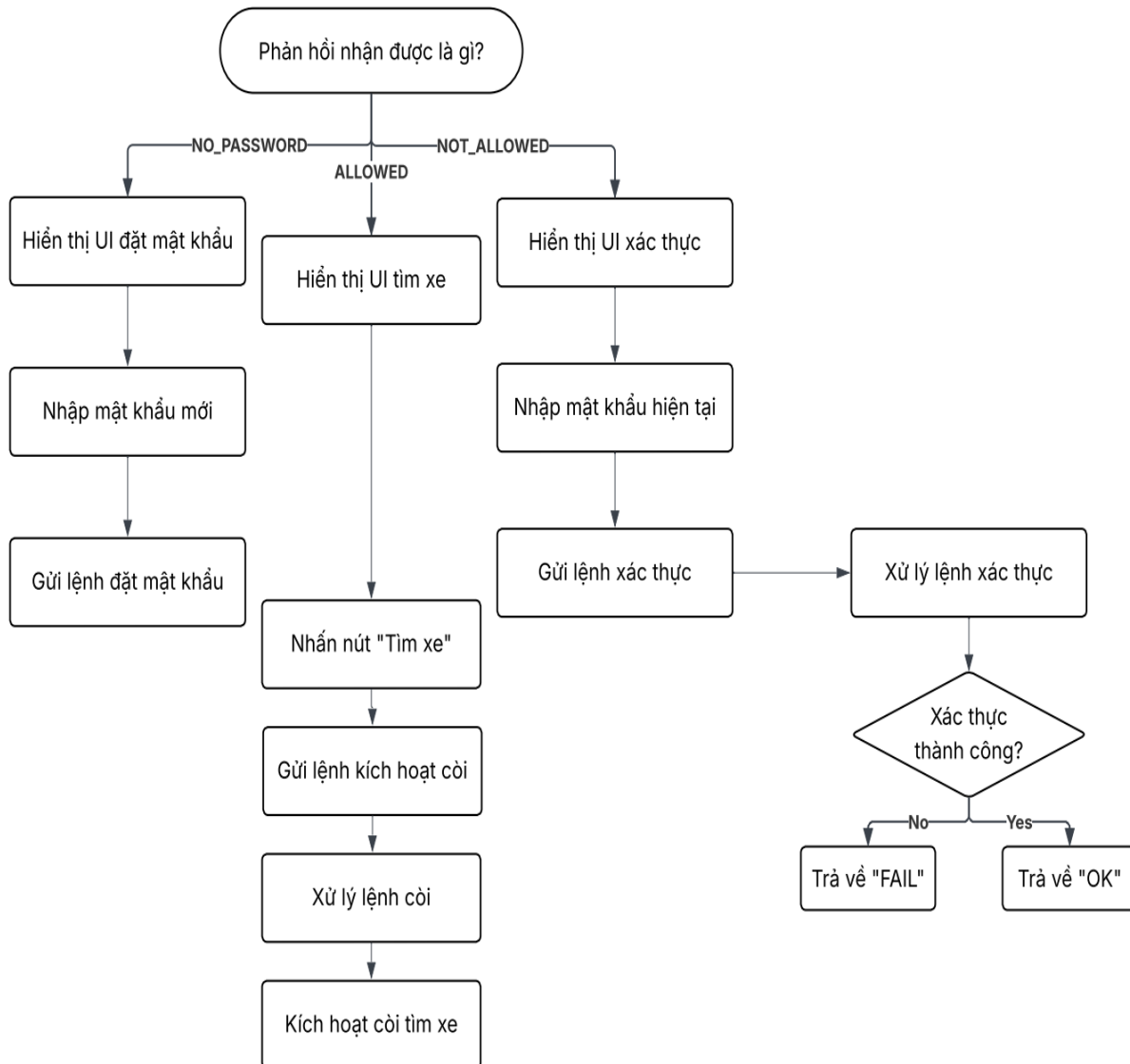


Hình 6.2 Lưu đồ giải thuật tìm xe (trước khi nhận được phản hồi)

Thiết bị di động bắt đầu bằng việc mở ứng dụng BikeGuard và quét tìm thiết bị BLE gần đó. Người dùng chọn thiết bị BikeGuard từ danh sách ứng dụng và thiết lập kết nối BLE. Sau khi kết nối, ứng dụng gửi lệnh kiểm tra (CHECK) kèm định danh thiết bị để xác minh quyền truy cập.

Dựa trên phản hồi từ thiết bị, ứng dụng sẽ thực hiện một trong ba hành động: hiển thị màn hình đặt mật khẩu (NO_PASSWORD), hiển thị màn hình xác thực

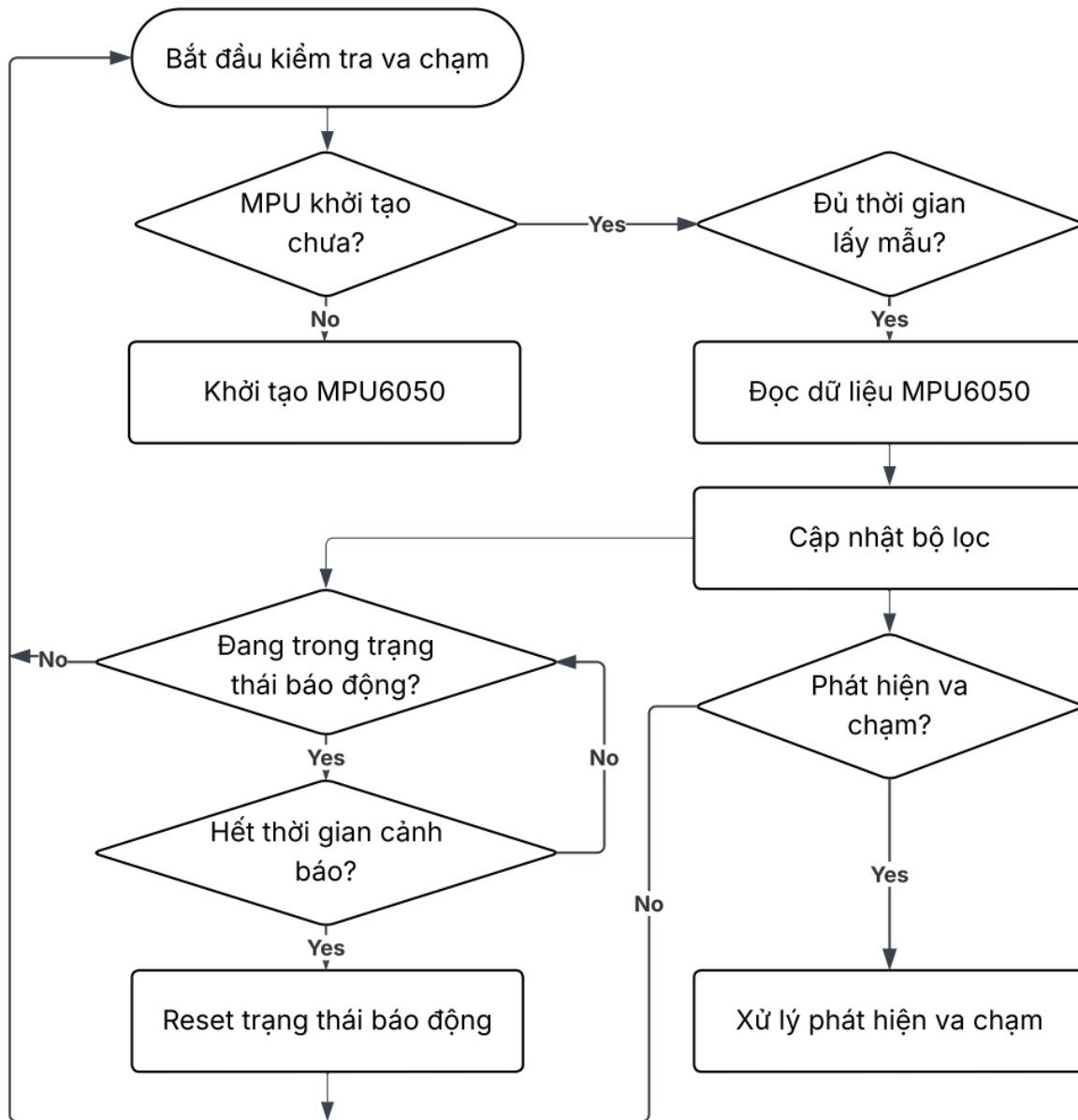
(NOT_ALLOWED) hoặc hiển thị trực tiếp giao diện tìm xe (ALLOWED). Khi người dùng nhấn nút "Tìm xe", ứng dụng sẽ gửi lệnh kích hoạt còi đến thiết bị.



Hình 6.3 Lưu đồ giải thuật tìm xe (sau khi nhận được phản hồi)

Thiết bị gắn xe liên tục quảng cáo BLE để được phát hiện. Khi có kết nối, thiết bị bật LED báo hiệu và xử lý các lệnh nhận được. Khi nhận lệnh CHECK, thiết bị kiểm tra xem đã có mật khẩu chưa và thiết bị có trong danh sách cho phép không, từ đó trả về phản hồi phù hợp. Thiết bị cũng xử lý các lệnh đặt mật khẩu (SETPASS) và xác thực (AUTH), đánh dấu thiết bị đã xác thực nếu mật khẩu đúng. Khi nhận lệnh kích hoạt còi và thiết bị đã được xác thực, thiết bị sẽ phát ra 5 tiếng bíp ngắn để giúp người dùng tìm xe.

6.3 Lưu đồ giải thuật phát hiện va chạm và gửi cảnh báo



Hình 6.4 Lưu đồ giải thuật phát hiện va chạm

Thiết bị gắn xe liên tục kiểm tra va chạm trong vòng lặp chính. Đầu tiên, hệ thống kiểm tra cảm biến MPU6050 đã được khởi tạo chưa, nếu chưa thì tiến hành khởi tạo. Tiếp theo, hệ thống kiểm tra tần số lấy mẫu để đảm bảo không đọc cảm biến quá nhanh, sau đó đọc dữ liệu gia tốc và lọc nhiễu.

Nếu hệ thống đang trong trạng thái báo động, kiểm tra xem đã hết thời gian cảnh báo chưa để reset trạng thái. Nếu không, hệ thống gọi thuật toán phát hiện va chạm để phân tích dữ liệu.



Hình 6.5 Lưu đồ giải thuật xử lý va chạm và gửi cảnh báo

Thuật toán phát hiện va chạm làm việc thông qua nhiều bước lọc. Đầu tiên, giá trị G-Force được so sánh với ngưỡng sơ bộ. Nếu vượt qua, giá trị được lưu

vào bộ đệm đỉnh và hệ thống tìm giá trị cao nhất trong cửa sổ IMPACT_WINDOW mẫu. Giá trị đỉnh này được so sánh với ngưỡng chính. Tiếp theo, hệ thống tính tỷ lệ tăng đột biến bằng cách so sánh với đường cơ sở. Chỉ khi thỏa mãn cả ba điều kiện: vượt ngưỡng, tỷ lệ tăng cao và đỉnh xảy ra gần đây, hệ thống mới xác định đó là va chạm.

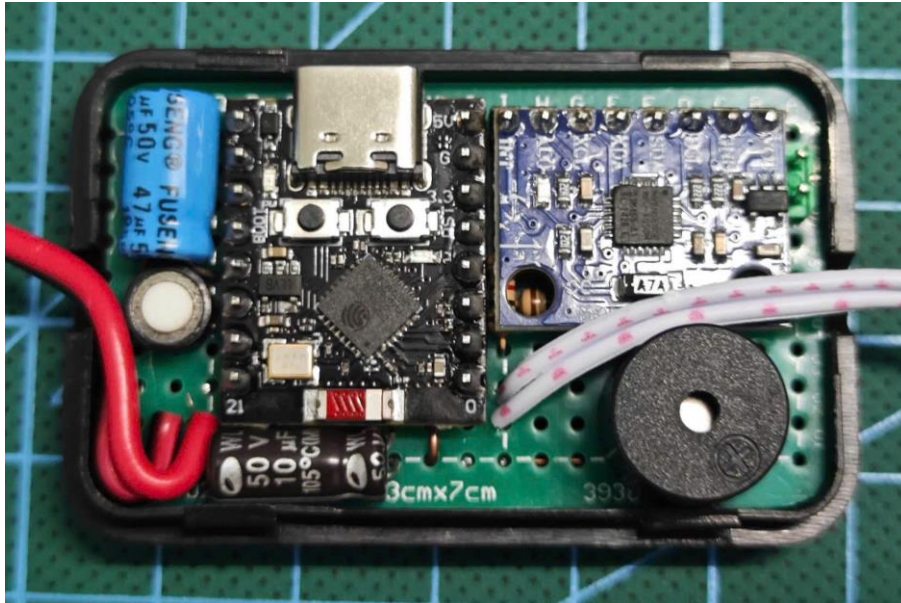
Khi phát hiện va chạm, hệ thống đánh dấu trạng thái báo động, gửi thông báo BLE nếu có kết nối và kích hoạt còi báo động với 10 tiếng bíp khẩn cấp theo mẫu 150ms/100ms.

Thiết bị di động đăng ký nhận thông báo va chạm từ thiết bị gắn xe. Khi nhận được thông báo, ứng dụng hiển thị cảnh báo, phát âm thanh, rung thiết bị và hiển thị các tùy chọn khẩn cấp như gọi số SOS, chia sẻ vị trí GPS hoặc xác nhận an toàn.

7 Kết quả thực hiện

7.1 Kết quả thực hiện phần cứng

Như hình minh họa 7.1 và 7.2, hệ thống phần cứng đã được tích hợp thành công với các thành phần chính đã nêu ở chương 5. Ngoài ra còn có hộp nhựa ABS với (55x35x1ml) để bảo vệ mạch chống tác động từ vật lý môi trường.



Hình 7.1 Thiết bị phần cứng sau khi hoàn thiện



Hình 7.2 Hộp nhựa ABS (55x35x1ml) bảo vệ phần cứng

Kết quả đạt được: Hệ thống đã được kiểm nghiệm qua 20 lần thử nghiệm và cho kết quả ổn định với tỷ lệ phát hiện va chạm là 80% và gửi tín hiệu khẩn cấp đến điện

thoại người thân thông qua gmail và telegram là 100% và định vị xe máy thông qua kết nối BLE và còi Buzzer là 100%

Phương pháp mô phỏng: Do không thể thực hiện thử nghiệm và chạm thực tế vì lý do an toàn nên nhóm đã tiến hành mô phỏng các tình huống va chạm và té ngã thông qua mô phỏng phanh gấp đột ngột.

Đánh giá hiệu quả: Mặc dù không thể thử nghiệm trực tiếp với va chạm thực tế, phương pháp mô phỏng của nhóm đã cung cấp dữ liệu đáng tin cậy về khả năng của hệ thống tuy nhiên vẫn còn nhiều hạn chế về khả năng phát hiện va chạm thực tế và thuật toán phát hiện va chạm vẫn chưa được tối ưu.

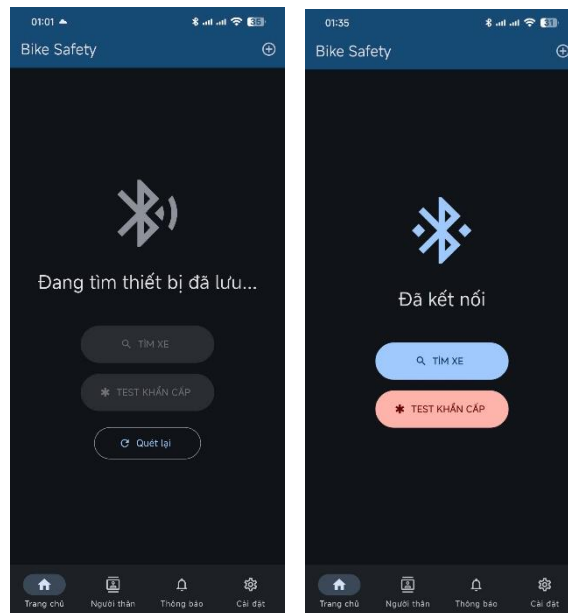
7.2 Kết quả thực hiện trên phần mềm

7.2.1 Tổng quan về kiến trúc ứng dụng

Ứng dụng Bike Safety được phát triển dựa trên Framework Flutter với kiến trúc client-server, trong đó thiết bị di động đóng vai trò client và module ESP32+MPU6050 đóng vai trò server. Hệ thống được thiết kế với các thành phần chính:

- Giao diện người dùng: Sử dụng Material Design 3 với 4 màn hình chính (Trang chủ, Người thân, Thông báo, Cài đặt)
- Kết nối BLE: Quản lý kết nối Bluetooth Low Energy giữa điện thoại và thiết bị
- Cơ sở dữ liệu: Lưu trữ thông tin người thân, lịch sử tai nạn và nhật ký thông báo
- Dịch vụ thông báo: Hỗ trợ gửi thông báo khẩn cấp qua Email và Telegram

7.2.2 Màn hình Trang chủ



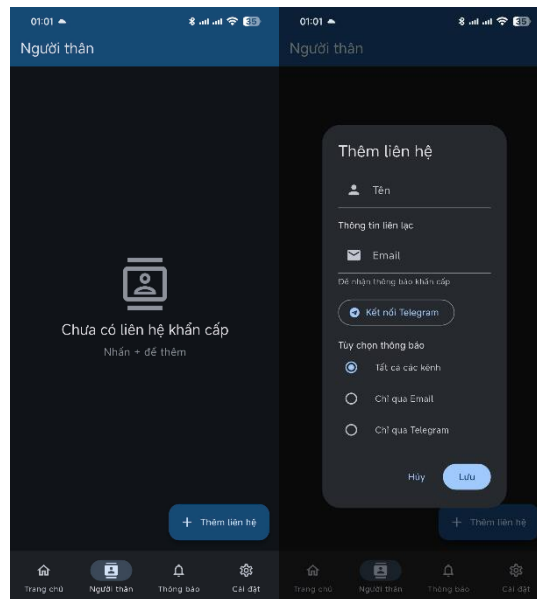
Hình 7.3 Giao diện trang chủ khi truy cập vào ứng dụng

Màn hình Trang chủ là nơi người dùng kết nối và tương tác với thiết bị bảo vệ xe đạp. Các tính năng chính bao gồm:

- Quét và kết nối thiết bị tự động: Khi mở ứng dụng, hệ thống tự động quét và kết nối với thiết bị đã lưu trước đó.
- Thêm thiết bị mới: Người dùng có thể quét và thêm thiết bị mới thông qua nút "+" trên thanh công cụ.
- Xác thực thiết bị: Khi kết nối với thiết bị mới, hệ thống yêu cầu tạo mật khẩu cho thiết bị. Với thiết bị đã kết nối trước đó, người dùng cần nhập mật khẩu để xác thực.
- Tìm xe: Kích hoạt còi báo trên thiết bị để tìm xe trong bãi đỗ.
- Kiểm tra tính năng khẩn cấp: Cho phép kiểm thử chức năng thông báo khẩn cấp, nếu chưa có danh sách thì sẽ có cảnh báo.

Giao diện hiển thị trạng thái kết nối hiện tại với biểu tượng trực quan và các nút chức năng chỉ được kích hoạt khi thiết bị được kết nối thành công.

7.2.3 Màn hình Người thân



Hình 7.4 Giao diện quản lý người thân

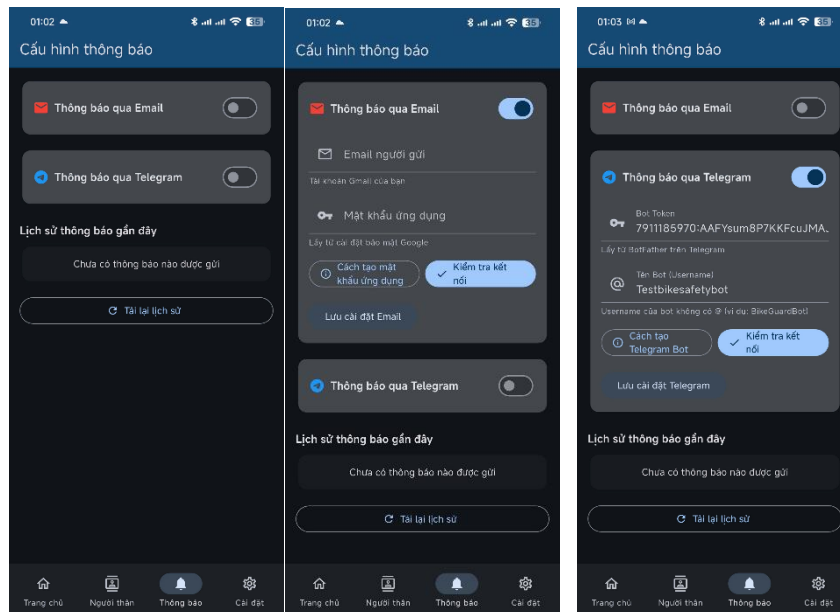
Màn hình Người thân cho phép người dùng quản lý danh sách liên hệ khẩn cấp. Mỗi người thân có thể được cấu hình để nhận thông báo qua Email, Telegram hoặc cả hai kênh, tùy theo tùy chọn mà người dùng thiết lập.

Các tính năng chính:

- Thêm người thân: Nhập tên và thông tin liên lạc (email).
- Kết nối Telegram: Tạo liên kết với người thân qua Telegram Bot để gửi thông báo khẩn cấp.
- Tùy chọn thông báo: Cho phép lựa chọn kênh thông báo (tất cả các kênh, chỉ Email, hoặc chỉ Telegram).
- Gửi thông báo thử nghiệm: Kiểm tra kết nối thông báo với từng liên hệ.

Hệ thống xác thực Telegram được thiết kế đặc biệt với cơ chế tạo mã xác minh, giúp bảo mật thông tin liên lạc. Người dùng có thể dễ dàng liên kết tài khoản Telegram của người thân với ứng dụng thông qua việc gửi mã xác thực đến bot.

7.2.4. Màn hình Thông báo



Hình 7.5 Giao diện cấu hình thông báo

Màn hình Thông báo cho phép người dùng cấu hình hệ thống thông báo khẩn cấp với hai kênh chính: Email và Telegram.

Cấu hình Email:

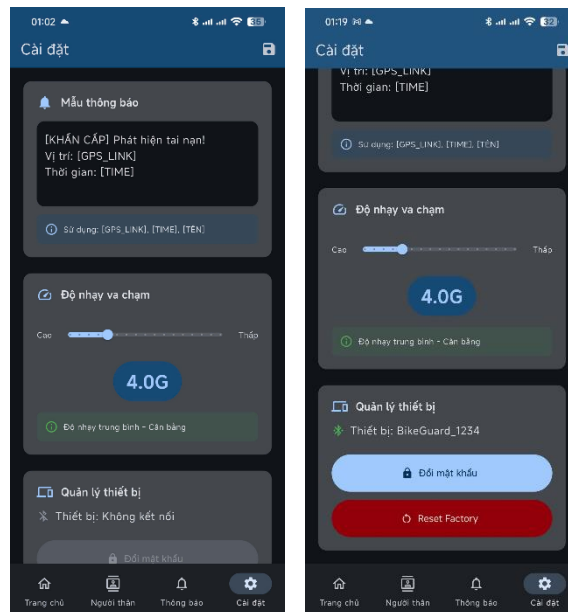
- Nhập email người gửi (Gmail)
- Nhập mật khẩu ứng dụng được tạo từ tài khoản Google
- Kiểm tra kết nối để đảm bảo cấu hình chính xác
- Hướng dẫn chi tiết về cách tạo mật khẩu ứng dụng Gmail

Cấu hình Telegram:

- Nhập Bot Token và Username Bot từ BotFather
- Kiểm tra kết nối để xác nhận Bot hoạt động
- Hướng dẫn chi tiết về cách tạo Telegram Bot

Phần lịch sử thông báo hiển thị danh sách các thông báo đã gửi gần đây, bao gồm thông tin về trạng thái gửi (thành công/thất bại), người nhận và kênh thông báo (Email/Telegram). Người dùng có thể xem chi tiết từng bản ghi để nắm rõ quá trình gửi thông báo.

7.2.5. Màn hình Cài đặt



Hình 7.6 Giao diện cài đặt

Màn hình Cài đặt cung cấp cho người dùng khả năng tùy chỉnh các thông số của hệ thống.

Mẫu thông báo:

- Tùy chỉnh nội dung thông báo khẩn cấp
- Hỗ trợ các biến động như [GPS_LINK], [TIME], [TÊN] để cá nhân hóa thông báo

Độ nhạy va chạm:

- Thanh trượt điều chỉnh độ nhạy từ Cao (2.0G) đến Thấp (10.0G)
- Hiển thị trực quan với thông tin về mức độ nhạy đang chọn
- Tự động đồng bộ với thiết bị khi điều chỉnh thay đổi

Quản lý thiết bị:

- Hiển thị trạng thái kết nối hiện tại
- Hỗ trợ đổi mật khẩu thiết bị
- Tùy chọn reset factory để khôi phục thiết bị về trạng thái ban đầu

7.2.6. Tính năng phát hiện tai nạn

Hệ thống được trang bị cảm biến MPU6050 trên module ESP32 để phát hiện tai nạn. Thuật toán phát hiện va chạm được thiết kế thông minh với các kỹ thuật lọc nhiễu và phát hiện đỉnh (peak detection). Khi phát hiện va chạm đủ lớn (vượt ngưỡng cài đặt), hệ thống sẽ kích hoạt quy trình thông báo khẩn cấp.

Quy trình gửi thông báo khẩn cấp gồm các bước:

1. Xác định vị trí GPS hiện tại
2. Tạo nội dung thông báo dựa trên mẫu người dùng đã cài đặt
3. Gửi thông báo tới danh sách người thân qua các kênh đã cấu hình
4. Lưu lịch sử thông báo và kết quả gửi

7.2.7. Cơ chế bảo mật

Hệ thống áp dụng nhiều cơ chế bảo mật để đảm bảo an toàn cho người dùng:

- Xác thực thiết bị bằng mật khẩu
- Quản lý danh sách thiết bị được phép kết nối
- Lưu trữ bảo mật thông tin nhạy cảm như mật khẩu và token API
- Cơ chế xác thực Telegram để đảm bảo thông báo chỉ gửi đến đúng người nhận

7.2.8. Tối ưu hóa hiệu suất

Ứng dụng được tối ưu hóa để tiết kiệm năng lượng và cải thiện trải nghiệm người dùng:

- Tự động quét thông minh dựa trên trạng thái kết nối
- Cơ chế lọc và xử lý dữ liệu tăng tốc độ phát hiện va chạm
- Sử dụng IndexedStack để giữ trạng thái các màn hình, giảm thời gian tải lại

Tổng kết, ứng dụng Bike Safety đã được triển khai thành công với đầy đủ tính năng đảm bảo an toàn cho người sử dụng, từ việc kết nối thiết bị, quản lý danh sách người thân, cấu hình hệ thống thông báo đến việc tùy chỉnh các thông số theo nhu cầu cá nhân.

8 Kết luận và hướng phát triển

8.1 Kết luận

Sau thời gian nghiên cứu và thực hiện, nhóm đã hoàn thành xong đề tài “Thiết kế và phát triển hệ thống phát hiện va chạm và gửi cảnh báo khẩn cấp, tích hợp tìm xe trong bãi bằng BLE”. Nhóm đã hoàn thành được mục tiêu đề ra ban đầu là phát hiện được va chạm khi đi xe máy và gửi cảnh báo khẩn cấp về điện thoại của người nhà, ngoài ra còn có thể tìm xe trong bãi đỗ xe bằng công nghệ Bluetooth thông qua ứng dụng trên điện thoại mà nhóm đã xây dựng.

Về phần cứng:

- Thiết kế thành công mạch với ESP32-C3, cảm biến MPU6050, buzzer, tụ hóa và LED
- Hiểu được các thông số của linh kiện và đặc tính kỹ thuật
- Tích hợp khối nguồn chuyển đổi 12V-5V ổn định, phù hợp với hệ thống điện xe máy
- Đóng gói thiết bị trong hộp nhựa ABS, kích thước nhỏ gọn

Về phần mềm:

- Lập trình thuật toán phát hiện va chạm với độ chính xác 80% trong điều kiện thực tế.
- Kết nối BLE ổn định trong phạm vi 85m ở bãi xe ngoài trời không có vật cản và khoảng 50m khi có vật cản.
- Phát triển ứng dụng Android với giao diện thân thiện với đầy đủ chức năng: tìm xe trong bãi, thêm được tối đa 3 tài khoản gmail và telegram để nhận được tin nhắn phát hiện khi xe máy bị va chạm và lưu lại lịch sử va chạm

Tính năng hoàn thiện:

- Gửi tin nhắn cảnh báo tự động qua gmail và telegram trong vòng 5 giây sau khi phát hiện va chạm
- Tìm xe bằng âm thanh buzzer hoạt động tốt trong môi trường ồn ào (85dB)

Mặc dù hệ thống đã đạt được những kết quả khả quan trong quá trình nghiên cứu và phát triển, nhóm nhận thấy vẫn còn có một số hạn chế cần được cải thiện:

- Hạn chế về độ chính xác cảm biến : Khó phân biệt giữa va chạm thực sự và các tình huống như phanh gấp, đánh lái mạnh
- Giới hạn về kết nối BLE: Tín hiệu có thể bị suy giảm mạnh trong bãi xe tầng hầm có nhiều vật cản bê tông
- Phụ thuộc vào điện thoại: Điện thoại người thân cần phải có wifi hoặc dữ liệu di động để nhận được tin nhắn cảnh báo
- Hạn chế về năng lượng: Không có chế độ tiết kiệm năng lượng khi xe đỗ lâu
- Giới hạn về loại va chạm: Hệ thống hiện tại mới chỉ được kiểm thử với các tình huống phanh gấp và té ngã đơn giản, chưa đánh giá được khả năng phát hiện các loại va chạm khác như va chạm bên hông, va chạm phía sau hoặc va chạm khi dừng đỗ.

8.2 Hướng phát triển

Dựa trên những hạn chế đã được nêu trên, nhóm đề xuất các hướng phát triển cụ thể cho hệ thống phát hiện va chạm và tìm xe trong bãi xe

- Sử dụng trí tuệ nhận tạo để nâng cấp thuật toán phát hiện va chạm
 - Thu thập dữ liệu từ 200 tình huống va chạm thực tế và 300 tình huống nhiễu để huấn luyện mô hình
 - Thay thế thuật toán hiện tại bằng mô hình machine learning sử dụng TensorFlow Lite
- Phát triển module eSIM cho phép thiết bị tự gửi thông báo không cần qua điện thoại
- Xây dựng hệ thống định vị cho xe
 - Phát triển thuật toán dead reckoning kết hợp GPS và MPU6050 để định vị chính xác trong hầm đỗ xe
 - Tích hợp cảm biến chuyển động để kích hoạt cảnh báo khi xe bị dịch chuyển

9 Phụ lục

9.1 Phân chia công việc

ID	Activity	Description	Duration (Days)	People
1	Phát triển hệ thống xác thực, triển khai bố cục báo cáo	Xây dựng cơ chế quản lý mật khẩu, xác thực thiết bị, danh sách thiết bị được phép	7	Đào Vĩnh Khang
2	Phát triển thuật toán phát hiện va chạm, triển khai phần cứng, thực nghiệm hệ thống	Xây dựng thuật toán lọc nhiễu, phát hiện đỉnh, tính toán ngưỡng va chạm thích ứng	8	Võ Xuân Lộc
3	Phát triển phần cứng, thực nghiệm hệ thống	Thiết kế mạch, kết nối cảm biến MPU6050, tích hợp module BLE, còi báo động và nút reset	9	Phan Nhị Hào
4	Phát triển ứng dụng di động, xây dựng giải thuật cho hệ thống	Thiết kế và xây dựng ứng dụng điều khiển trên smartphone, kết nối BLE, giao diện người dùng	7	Nguyễn Hoàng Ngân

9.2 Chi nguyên vật liệu

ID	Parts/Components	Amount	Price per Unit	Total
01	Cảm Biến Con Quay Gia Tốc GY-521 MPU-6050	1	32,000	32,000
02	Mosfet AO3400 N-Channel	1	2,800	2,800
03	Buzzer 1209	1	3,000	3,000
04	LM7805R IC Ổn Áp 5V 1.5A TO-263	1	6,000	6,000
05	ESP32-C3 Super Mini	1	59,000	59,000
08	Nút Nhấn D5 Cao 5mm 2 Chân Tròn Xuyên Lỗ	1	600	600
09	Tụ hóa 100uF 25V	1	570	570

10	Tụ Hóa 10uF 16V	1	450	450
11	LED Xanh Dương 5mm	1	160	160
12	Hộp Nhựa 55x35x15	1	8,000	8,000
13	Điện Trở 1 KOhm 1/4W	2	40	80
14	PCB Hàn Test Board	1	3,000	3,000
15	Dây Điện Đôi Đỏ Đen RVB-0.5 0.5mm2	1	6,000	6,000
				121,660.

TÀI LIỆU THAM KHẢO

- [1] A. A. Galadima, "Arduino as a learning tool," 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, 2014, pp. 1-4, doi: 10.1109/ICECCO.2014.6997577.
- [2] Google, "Flutter," [Online]. Available: <https://flutter.dev/>. [Accessed: May 15, 2025].
- [3] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins, "Global Positioning System: Theory and Practice," 5th ed., Springer, 2001.
- [4] M. Barr and A. Massa, "Programming Embedded Systems: With C and GNU Development Tools," 2nd ed., O'Reilly Media, 2006.
- [5] B. Razavi, "Design of Integrated Circuits for Optical Communications," 2nd ed., Wiley, 2012.
- [6] HTC, 3-Terminal 1.5A Positive Voltage Regulator LM78xx (Rev. 1.2.2), Jun. 2023. [Online]. Available: <https://www.htc.com/>