

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1



HỌC PHẦN IOT VÀ ỨNG DỤNG

BÁO CÁO CUỐI KỲ

TÊN ĐỀ TÀI: THIẾT BỊ ĐO NỒNG ĐỘ CÒN

Nhóm lớp: 05 – Nhóm bài tập: 15

Danh sách thành viên nhóm:

Chu Ngọc Thắng MSV: B22DCCN807

Nguyễn Hoàng Vũ MSV: B22DCCN915

Đặng Tiến Đạt MSV: B22DCCN183

Lâm Trung Hiếu MSV: B21DCCN354

Hà Nội – 2025

DANH MỤC HÌNH VẼ.....	3
DANH MỤC BẢNG BIỂU	4
LỜI CẢM ƠN	5
TÓM TẮT	6
1. Giới thiệu chung.....	7
1.1. Lý do chọn đề tài	7
1.2. Tổng quan về dự án	7
1.3. Mục đích của dự án	8
1.4. Đối tượng và phạm vi nghiên cứu	9
1.5. Phương pháp nghiên cứu	10
1.6. Công nghệ và thiết bị cần thiết cho dự án	11
2. Nền tảng lý thuyết.....	12
2.1. Tổng quan về IOT.....	12
2.2. Các kiến thức cơ bản	17
3. Phân tích yêu cầu chức năng.....	28
3.1. Chức năng đăng nhập vào hệ thống.....	28
3.2. Chức năng đo nồng độ cồn và cảnh báo vi phạm	28
3.3. Chức năng quản lý dữ liệu và truyền tải bảo mật	29
3.4. Chức năng quản lý tập trung và phân tích thống kê	30
3.5. Chức năng chuyển đổi và xử lý lỗi trạng thái.....	30
4. Thiết kế hệ thống	31
4.1. Đặc tả mô hình miền.....	31
4.2. Đặc tả các thành phần chức năng và hoạt động.....	34
4.3 Đặc tả luồng hoạt động.....	38
4.4 Tích hợp Thiết bị	41
4.5 Phát triển ứng dụng.....	42
5. Kết luận.....	49
5.1. Kết quả đạt được.....	49
5.2. Hạn chế	50
5.3. Hướng phát triển tương lai	50
BẢNG PHÂN CÔNG CÔNG VIỆC	52

DANH MỤC HÌNH VẼ

Hình 2.1 : Sơ đồ chân cảm biến MQ3

Hình 2.2 : Module ESP32

Hình 2.3: LCD 16x2

Hình 2.4: Sơ đồ chân module I2C

Hình 2.5: Giao thức UART

Hình 2.6: Giao thức I2C

Hình 2.7: Giao thức HTTP

Hình 4.1: Sơ đồ đặc tả mô hình miền

Hình 4.2: Đặc tả thành phần chức năng

Hình 4.3: Sơ đồ mạch

Hình 4.4: Giao diện đăng nhập

Hình 4.5: Giao diện kết nối

Hình 4.6: Giao diện đo nồng độ cồn

Hình 4.7: Giao diện quản lý thiết bị

Hình 4.8: Giao diện thêm thông tin

Hình 4.9: Giao diện quản lý tài khoản

Hình 4.10: Giao diện thêm tài khoản

Hình 4.11: Giao diện lịch sử đo

Hình 4.12: Giao diện thống kê

DANH MỤC BẢNG BIỂU

Bảng 2.1: So sánh ESP32 và ESP8266

Bảng 4.1: Bảng mô tả thư viện

Bảng 4.2: Bảng mô tả các hàm

Bảng 4.3 :Bảng thư viện website

Bảng 4.4: Bảng mô tả các endpoint

Bảng 4.5: Danh sách linh kiện

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn sâu sắc đến Học viện Công nghệ Bưu chính Viễn thông đã đưa môn học IoT và ứng dụng vào trong chương trình giảng dạy, tạo điều kiện cho chúng em được tiếp cận với lĩnh vực công nghệ đang phát triển mạnh mẽ này.

Chúng em xin đặc biệt gửi lời cảm ơn chân thành tới giảng viên Kim Ngọc Bách đã tận tình hướng dẫn, truyền đạt cho chúng em những kiến thức chuyên môn quý báu về IoT trong suốt thời gian học vừa qua. Thầy không chỉ giúp chúng em hiểu rõ ý nghĩa và tầm quan trọng của IoT trong thực tiễn đời sống và công việc mà còn cung cấp những ví dụ thực tế và kiến thức mới lạ để chúng em hình dung rõ hơn về nội dung môn học.

Những kiến thức này đã trang bị cho chúng em những kỹ năng cần thiết, kinh nghiệm thực tiễn và góc nhìn rộng hơn về việc ứng dụng IoT vào các vấn đề xã hội. Chúng em cũng xin cảm ơn các thành viên trong nhóm đã cùng nhau hợp tác, nỗ lực nghiên cứu và làm việc chăm chỉ để hoàn thành đề tài này.

Bài báo cáo của chúng em chắc chắn không thể tránh khỏi những thiếu sót, chúng em kính mong thầy xem xét và góp ý để chúng em có thể hoàn thiện kiến thức và sản phẩm của mình hơn nữa. Chúng em xin chân thành cảm ơn!

TÓM TẮT

Hệ thống Cảm biến đo nồng độ cồn

Hệ thống cảm biến đo nồng độ cồn là một giải pháp Internet of Things (IoT) được thiết kế để giám sát, đo lường nồng độ cồn trong hơi thở (BAC) tại hiện trường và tự động truyền tải dữ liệu vi phạm về máy chủ trung tâm.

Cảm biến khí MQ-3 sẽ thu thập dữ liệu nồng độ cồn, sau đó gửi thông tin đo được lên hệ thống quản lý (Web-App) thông qua kết nối Wi-Fi/Bluetooth/HTTPS. Kết quả đo được hiển thị tức thời trên màn hình LCD/OLED tại thiết bị, đồng thời được ghi nhận trên giao diện Web-App để giám sát từ xa.

Hệ thống giúp theo dõi nồng độ cồn theo thời gian thực, hỗ trợ cảnh báo sớm khi nồng độ cồn vượt ngưỡng pháp lý, và tự động hóa việc ghi nhận chứng cứ vi phạm, nâng cao hiệu quả và tính minh bạch trong công tác xử lý của lực lượng chức năng.

Sản phẩm sử dụng vi điều khiển ESP32 để tối ưu hóa khả năng xử lý, quản lý năng lượng (chế độ Deep Sleep), và hỗ trợ các giao thức kết nối mạng an toàn để đảm bảo tính toàn vẹn và bảo mật của dữ liệu pháp lý.

1. Giới thiệu chung

1.1. Lý do chọn đề tài

Tình trạng lái xe sau khi sử dụng rượu bia hiện nay đang là vấn đề rất nghiêm trọng, phổ biến trong xã hội. Việc này không chỉ tiềm ẩn nguy cơ cao gây ra tai nạn giao thông mà còn dẫn đến những tổn thất lớn về cả người và của. Đây là thực trạng cần phải được xử lý kịp thời và quyết liệt nhằm bảo vệ sự an toàn chung.

Để giảm thiểu tình trạng này, Nhà nước và các lực lượng như cảnh sát giao thông, cảnh sát cơ động, cùng các đơn vị bảo đảm an ninh trật tự xã hội đang tích cực và liên tục triển khai các biện pháp kiểm soát, xử lý người tham gia giao thông trong tình trạng say xỉn. Nỗ lực này nhằm đảm bảo an toàn cho người dân và ổn định trật tự xã hội.

Nhằm hỗ trợ hiệu quả cho lực lượng chức năng trong việc kiểm tra và xử phạt người vi phạm quy định về nồng độ cồn, nhóm chúng tôi đã phát triển hệ thống đo nồng độ cồn dựa trên cảm biến hơi thở. Dữ liệu đo được truyền về máy chủ qua giao thức Http, sau đó server lưu trữ thông tin vi phạm và tự động xác định mức phạt, thông báo cho cả lực lượng chức năng và người vi phạm để đảm bảo tính công khai, minh bạch. Ngoài ra, hệ thống còn cung cấp các chức năng thống kê và trực quan hóa dữ liệu vi phạm như số lượng, mức độ vi phạm trung bình,... giúp việc quản lý và phân tích trở nên thuận tiện hơn.

1.2. Tổng quan về dự án

- Dự án thiết kế Thiết bị đo nồng độ cồn là một ứng dụng điển hình của công nghệ IoT, cho phép lực lượng chức năng thu thập, giám sát, và quản lý dữ liệu vi phạm nồng độ cồn tại hiện trường theo thời gian thực.

- Hệ thống bao gồm hai thành phần chính:

- + Thiết bị phần cứng cầm tay: Là thiết bị di động, sử dụng cảm biến MQ-3 để đo nồng độ cồn và bộ vi điều khiển ESP32 để xử lý dữ liệu và kết nối không dây. Thiết bị này sẽ hiển thị kết quả đo và phát cảnh báo tại chỗ (âm thanh, đèn LED) khi phát hiện vi phạm.

- + Hệ thống phần mềm Backend/Frontend: Nền tảng quản lý tập trung (sử dụng Springboot Backend và Java Swing/Web-App Frontend) có nhiệm vụ nhận, lưu trữ, và trực quan hóa dữ liệu đo từ các thiết bị tại hiện trường. Hệ thống này cũng cung cấp các chức năng thống kê, xác định mức phạt, và quản lý thiết bị/người dùng.

1.3. Mục đích của dự án

- Vấn đề: Tình trạng người điều khiển phương tiện giao thông sau khi sử dụng rượu bia là một vấn đề nghiêm trọng, tiềm ẩn nguy cơ cao gây ra các vụ tai nạn thảm khốc, thiệt hại về người và của.

- Thách thức hiện tại: Công tác kiểm tra và xử lý vi phạm của lực lượng chức năng đôi khi còn gặp khó khăn, cần một công cụ hiệu quả hơn để quy trình diễn ra nhanh chóng, chính xác và minh bạch.

- Nhu cầu: Cần một hệ thống thông minh không chỉ để đo đạc mà còn tự động hóa việc ghi nhận, lưu trữ bằng chứng vi phạm và hỗ trợ ra quyết định xử phạt, giảm thiểu sai sót và tăng cường hiệu quả quản lý.

- Mục đích của dự án: Hệ thống được xây dựng nhằm mục đích giám sát, tự động hóa và tối ưu hóa quy trình kiểm soát nồng độ cồn, cụ thể như sau:

+ Giám sát tự động: Sử dụng cảm biến để đo nồng độ cồn (MQ-3) trong hơi thở một cách chính xác và nhanh chóng, thay thế các phương pháp ước lượng cảm tính.

+ Tự động hóa cảnh báo và ghi nhận: Hệ thống tự động kích hoạt cảnh báo (âm thanh, ánh sáng) khi phát hiện vi phạm. Dữ liệu về lần đo và thông tin người vi phạm được tự động gửi và lưu trữ trên máy chủ.

+ Tối ưu hóa quy trình xử lý: Hỗ trợ lực lượng chức năng truy xuất thông tin, xác định mức phạt và tạo biên bản xử phạt một cách nhanh chóng, đảm bảo tính minh bạch.

+ Thiết kế giao diện thân thiện, hiển thị kết quả đo trên màn hình LCD hoặc các ứng dụng di động, website, đồng thời tích hợp hệ thống cảnh báo khi phát hiện nồng độ cồn vượt mức.

+ Kết nối hệ thống với mạng IoT để giám sát dữ liệu đo nồng độ cồn từ xa theo thời gian thực, đồng thời lưu trữ và phân tích dữ liệu phục vụ quản lý.

+ Định danh duy nhất cho mỗi thiết bị đo giúp việc theo dõi và quản lý trở nên thuận tiện hơn, kèm theo chức năng giám sát trạng thái hoạt động và hiệu suất của thiết bị.

+ Hỗ trợ nâng cao an toàn giao thông bằng cách phát hiện và ngăn chặn người lái xe khi có nồng độ cồn vượt quy định.

1.4. Đối tượng và phạm vi nghiên cứu

1.4.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu chính của đề tài bao gồm:

- Về phần cứng:

+ Vi điều khiển và xử lý dữ liệu: Tập trung nghiên cứu module ESP32, tận dụng kiến trúc Lõi kép (Dual-core) để thực hiện song song các tác vụ quan trọng. Một lõi dành cho việc đọc tín hiệu Analog từ cảm biến MQ-3, tính toán nồng độ cồn (BAC), và điều khiển hiển thị; lõi còn lại dành cho việc quản lý kết nối không dây (Wi-Fi và Bluetooth/BLE) và giao tiếp với máy chủ. Nghiên cứu sâu về chế độ Deep Sleep của ESP32 để tối ưu hóa việc sử dụng năng lượng cho thiết bị di động.

+ Cảm biến đo nồng độ cồn: Nghiên cứu chi tiết nguyên lý hoạt động và đặc tính kỹ thuật của cảm biến khí MQ-3. Đây là cảm biến bán dẫn oxit kim loại có độ nhạy cao với hơi ethanol (rượu). Nghiên cứu tập trung vào việc hiệu chuẩn (Calibration) cảm biến, xây dựng hàm chuyển đổi (Mapping Function) từ tín hiệu điện áp Analog sang giá trị nồng độ cồn BAC tiêu chuẩn (mg/L hoặc % BAC) để đảm bảo kết quả đo có tính pháp lý và chính xác.

+ Cơ cấu hiển thị và cảnh báo: Nghiên cứu tích hợp màn hình LCD 16x2 thông qua giao thức I2C để tối ưu số lượng chân cần sử dụng. Nghiên cứu cơ chế điều khiển Đèn LED để kích hoạt tức thời (trong vòng < 1 giây) khi nồng độ cồn vượt ngưỡng cho phép, cung cấp cảnh báo tại chỗ.

- Về phần mềm và giao thức:

+ Kiến trúc Client-Server và Backend: Nghiên cứu cách xây dựng một Server trung gian (Backend) sử dụng nền tảng như Spring Boot (Java), đóng vai trò là trạm điều phối tập trung. Server có nhiệm vụ xác thực thiết bị, nhận dữ liệu đo, phân tích mức độ vi phạm, và lưu trữ vào cơ sở dữ liệu.

+ Giao thức Truyền tin bảo mật: Tập trung phân tích và ứng dụng giao thức HTTPS kết hợp với chứng chỉ bảo mật để mã hóa toàn bộ dữ liệu đo lường được truyền từ thiết bị lên Server.

+ Giao diện người dùng: Nghiên cứu xây dựng web tương tác cao, cho phép hiển thị trực quan các biểu đồ thống kê về tỉ lệ vi phạm, quản lý danh sách thiết bị và thống kê các loại vi phạm, phiếu phạt.

1.4.2. Phạm vi nghiên cứu

- Phạm vi về chức năng hệ thống:

+ Đề tài tập trung vào việc thực hiện trọn vẹn quy trình của một hệ thống IoT giám sát: Thu thập dữ liệu (Đo nồng độ cồn) → Xử lý sơ bộ (Tính BAC và Cảnh báo tại ESP32) → Truyền tải (Gửi dữ liệu vi phạm an toàn qua HTTPS) → Lưu trữ & Trực quan hóa (Tại Server và Dashboard).

+ Chức năng tự động hóa chỉ tập trung vào việc tự động hóa ghi nhận vi phạm và tối ưu hóa năng lượng. Hệ thống sẽ không bao gồm các tính năng phức tạp như phân tích dữ liệu bằng Machine Learning hay AI dự đoán vi phạm.

- Phạm vi về môi trường hoạt động:

+ Thiết bị phần cứng: Được thiết kế là mô hình cầm tay, hoạt động trong môi trường ngoài trời và di động, chịu được các điều kiện như nhiệt độ, bụi bẩn, và độ ẩm ở mức độ nhất định (yêu cầu vỏ bảo vệ).

+ Điều kiện kết nối: Hệ thống được giả định hoạt động dựa trên nền tảng mạng Wi-Fi hoặc Bluetooth. Phạm vi truyền tải dữ liệu Server phụ thuộc vào sự ổn định của kết nối tại hiện trường.

- Phạm vi về công nghệ và xử lý:

+ Tính toàn vẹn dữ liệu: Tốc độ phản hồi của hệ thống được tối ưu hóa để đảm bảo dữ liệu đo được ghi nhận và gửi lên Server trong vòng dưới 3 giây kể từ khi đo xong, hạn chế tối đa việc mất mát bằng chứng vi phạm.

+ Hiệu năng Server: Hệ thống Backend được giới hạn trong việc xử lý tối đa 100 yêu cầu đo/giây để đảm bảo độ tin cậy và hiệu năng ổn định trong điều kiện triển khai thực tế ban đầu.

1.5. Phương pháp nghiên cứu

1.5.1. Phương pháp nghiên cứu lý thuyết

- Nghiên cứu sâu về kiến trúc IoT (Perception Layer, Transport Layer, Processing Layer, Application Layer) và ứng dụng của nó.

- Nghiên cứu về nguyên lý hoạt động và hiệu chuẩn của cảm biến khí MQ-3.
- Nghiên cứu về đặc điểm kỹ thuật của vi điều khiển ESP32 và các giao thức truyền thông như I2C, Bluetooth, và HTTP.

1.5.2. Phương pháp thực nghiệm

- Tiến hành lắp đặt, cấu hình mạch điện giữa ESP32, cảm biến MQ-3, và màn hình LCD.
- Xây dựng và kiểm thử Firmware để đảm bảo độ chính xác của thuật toán chuyển đổi tín hiệu Analog sang nồng độ cồn (BAC).
- Phát triển và kiểm thử hệ thống Backend/Frontend để đảm bảo khả năng kết nối ổn định, tốc độ truyền dữ liệu nhanh chóng và tính toàn vẹn của dữ liệu được lưu trữ.

1.5.3. Phương pháp phân tích và đánh giá

- Phân tích các yêu cầu chức năng và phi chức năng như hiệu năng, bảo mật, và khả năng mở rộng của hệ thống.
- Đánh giá và so sánh kết quả đo lường thực tế với các tiêu chí thành công đã đặt ra để đưa ra các điều chỉnh cần thiết.
- Tiến hành chạy thử nghiệm để kiểm tra độ trễ của thao tác điều khiển gửi dữ liệu và nhận kết quả nồng độ cồn.
- Kiểm chứng độ chính xác: Đảm bảo mọi quá trình trong việc đo nồng độ cồn và nhận xác định vi phạm chính xác, minh bạch và công khai. Nếu gặp vấn đề, tiến hành điều chỉnh các vấn đề để hoàn thiện hệ thống.

1.6. Công nghệ và thiết bị cần thiết cho dự án

1.6.1. Phần cứng

- Vi điều khiển trung tâm (ESP32): Là bộ não xử lý chính, tích hợp Wi-Fi và Bluetooth. Chịu trách nhiệm nhận tín hiệu Analog từ MQ-3, tính toán BAC, điều khiển màn hình LCD và các cơ cấu cảnh báo và gửi dữ liệu lên server.
- Cảm biến nồng độ cồn (MQ-3): Là đầu dò chính, phát hiện và chuyển đổi nồng độ hơi cồn trong hơi thở thành tín hiệu điện áp Analog.

- Màn hình hiển thị (LCD 16x2 I2C): Hiển thị tức thời (real-time) kết quả nồng độ cồn đo được (BAC) và các thông báo cảnh báo/trạng thái kết nối. Sử dụng giao tiếp I2C giúp giảm số lượng chân kết nối trên ESP32.

- Cơ cấu cảnh báo (LED): Kích hoạt cảnh báo bằng ánh sáng (LED Đỏ/Xanh) ngay khi giá trị nồng độ cồn vượt ngưỡng cho phép.

- Nút nhấn (Button): Dùng để kích hoạt chức năng đo hoặc thực hiện chức năng reset hệ thống.

1.6.2. Phần mềm

- Công cụ lập trình Firmware (Arduino IDE): Môi trường phát triển tích hợp chính, được sử dụng để viết, biên dịch và nạp chương trình Firmware vào ESP32.

- Server Backend (Springboot - Java): Framework được sử dụng để xây dựng Server trung tâm.

- Giao diện người dùng (Java Swing/Web-App): Được sử dụng để phát triển ứng dụng Frontend.

- Cơ sở dữ liệu (PostgreSQL): Hệ thống quản lý cơ sở dữ liệu được sử dụng để lưu trữ dữ liệu của toàn hệ thống.

2. Nền tảng lý thuyết

2.1. Tổng quan về IOT

2.1.1 Khái niệm

IoT (Internet of Things) hay được biết với cái tên internet vạn vật, là một thuật ngữ công nghệ chỉ một mạng lưới các thiết bị vật lý được nhúng với các cảm biến, phần mềm và các công nghệ kết nối mạng, nhằm thu thập và trao đổi dữ liệu. Ý tưởng cơ bản của IoT là cho phép các thiết bị này giao tiếp và tương tác với nhau mà không cần sự can thiệp của con người.

Trong một môi trường IoT, các thiết bị từ đèn đến máy giặt, từ xe hơi đến thiết bị y tế đều có khả năng kết nối và gửi nhận dữ liệu qua Internet. Dữ liệu này có thể được sử dụng để tự động hóa quy trình, cung cấp thông tin phản hồi hoặc tạo ra các hệ thống thông minh để cải thiện hiệu suất và tiện ích.

2.1.2. Cấu trúc và cách thức hoạt động của IoT

Cấu trúc: Gồm 4 tầng chính

- Tầng thu thập dữ liệu (Perception Layer)

Tầng này đóng vai trò như là cánh cửa đầu tiên kết nối hệ thống IoT với thế giới thực. Nó là những thiết bị vật như thiết bị cảm biến, máy móc và thiết bị điện tử khác. Các thiết bị tại tầng này có khả năng thu thập dữ liệu trực tiếp từ môi trường xung quanh, bao gồm các thông tin như nhiệt độ, độ ẩm, ánh sáng, chất lượng không khí và áp suất. Nhờ vào các cảm biến tích hợp, chúng có thể cảm nhận các thay đổi và chuyển đổi chúng thành dữ liệu số để xử lý.

Một điểm đặc biệt quan trọng của tầng này là khả năng xử lý dữ liệu gần trường. Điều này có nghĩa là một số thiết bị trường có thể tự mình xử lý dữ liệu một cách nhanh chóng và hiệu quả ngay tại nơi thu thập, thay vì phải chờ đợi để truyền dữ liệu lên các tầng cao hơn trong hệ thống. Việc này giúp giảm bớt lượng dữ liệu cần truyền đi qua mạng và giảm độ trễ trong việc phản ứng với thông tin.

Hơn nữa, các thiết bị trường cũng có khả năng giao tiếp với nhau và với các nút mạng hoặc cổng giao tiếp trung gian khác như gateway hoặc nút cơ sở. Nhờ vào tính linh hoạt này, dữ liệu được thu thập từ các thiết bị trường có thể được truyền tải đến các tầng cao hơn trong hệ thống IoT để xử lý hoặc lưu trữ một cách hiệu quả.

- Tầng truyền tải thông tin (Gateways)

Tầng truyền tải thông tin được đặt ở tầng trung gian nằm giữa tầng thu thập dữ liệu và tầng xử lý, tầng truyền tải thông tin có vai trò chuyển đổi và truyền tải dữ liệu từ các thiết bị trường đến các hệ thống xử lý và lưu trữ.

Về cấu trúc, Gateways thường được trang bị các phần cứng và phần mềm đặc biệt để xử lý dữ liệu, thực hiện các giao thức truyền tải và bảo mật thông tin. Đồng thời, chúng có khả năng kết nối với nhiều thiết bị trường khác nhau qua các giao thức mạng khác nhau như Wifi, bluetooth, zigbee, hoặc LoRa.

Tầng này hoạt động như một cầu nối giữa thế giới vật lý của các thiết bị trường và thế giới kỹ thuật số của các hệ thống xử lý dữ liệu. Các Gateway có khả năng thu thập dữ liệu từ nhiều nguồn và gửi nó đến các hệ thống trung tâm để xử lý và phân tích. Đồng thời, chúng cũng có thể gửi lệnh điều khiển từ hệ thống trung tâm xuống các thiết bị trường để thực hiện các hành động cụ thể.

Một ví dụ điển hình về tầng truyền tải thông tin là một Gateway được sử dụng trong một hệ thống quản lý thông minh của một tòa nhà hoặc một khu vực công nghiệp. Gateway này có thể thu thập dữ liệu từ các cảm biến nhiệt độ, đèn, camera an ninh và các thiết bị khác, sau đó chuyển tiếp dữ liệu này đến hệ thống trung tâm để xử

lý và giám sát. Đồng thời, Gateway cũng có khả năng nhận lệnh từ hệ thống trung tâm và điều khiển các thiết bị trường để thực hiện các chức năng như điều chỉnh nhiệt độ, bật/tắt đèn, hoặc kích hoạt hệ thống bảo mật.

- Tầng xử lý thông tin (Processing Layer)

Tầng này đóng vai trò trung tâm trong việc xử lý và phân tích dữ liệu thu thập từ các thiết bị trường thông qua Gateway, cũng như lưu trữ dữ liệu cho các mục đích sau này.

Cấu trúc của tầng này thường bao gồm các máy chủ, hệ thống xử lý trung tâm, cơ sở dữ liệu và các công cụ phân tích dữ liệu. Các máy chủ được cấu hình mạnh mẽ để có thể xử lý lượng lớn dữ liệu từ hàng ngàn thiết bị IoT cùng một lúc.

Cơ sở dữ liệu thường được thiết kế để lưu trữ dữ liệu lớn và đa dạng, có khả năng mở rộng linh hoạt để phục vụ cho việc lưu trữ dữ liệu trong thời gian dài và cho các mục đích phân tích sau này.

Tầng xử lý và lưu trữ dữ liệu thực hiện nhiều chức năng quan trọng như sau:
Xử lý dữ liệu: Tầng này tiếp nhận dữ liệu từ các Gateway và thực hiện các quy trình xử lý như lọc dữ liệu, chuyển đổi định dạng, và phân tích dữ liệu để trích xuất thông tin quan trọng và đưa ra các quyết định thông minh.

Phân tích dữ liệu: Các công cụ phân tích dữ liệu được sử dụng để khám phá thông tin tiềm ẩn từ dữ liệu thu thập, từ việc tạo ra các báo cáo tự động đến việc dự đoán xu hướng và sự cố trong tương lai.

Lưu trữ dữ liệu: Cơ sở dữ liệu được sử dụng để lưu trữ dữ liệu thu thập từ các thiết bị IoT, cho phép truy cập nhanh chóng và linh hoạt vào dữ liệu lịch sử để phục vụ cho các mục đích như phân tích, giám sát và báo cáo.

Tầng xử lý và lưu trữ dữ liệu là trung tâm của hệ thống IoT, nơi mà dữ liệu được biến đổi thành thông tin hữu ích và có giá trị, đồng thời cũng là nơi lưu trữ tri thức để hỗ trợ quyết định và tối ưu hóa các quy trình trong thời gian thực và trong tương lai.

- Tầng ứng dụng (Application Layer)

Tầng này tập trung vào việc phát triển các ứng dụng và dịch vụ IoT để đáp ứng nhu cầu cụ thể của người dùng và doanh nghiệp. Các ứng dụng này có thể bao gồm giám sát và điều khiển tự động của các hệ thống thông minh như nhà thông minh, xe tự lái,...

Tầng ứng dụng cung cấp các giao diện người dùng và API cho phép người dùng cuối và các nhà phát triển tương tác với hệ thống IoT, truy cập vào dữ liệu, điều khiển các thiết bị và tận dụng những tính năng thông minh.

2.1.3. Lợi ích mà IoT mang lại

- Kết nối và điều khiển thiết bị thông minh: IoT cho phép kết nối một loạt các thiết bị thông minh như đèn, máy giặt, điều hòa, từ xa thông qua Internet. Điều này giúp người dùng có thể điều khiển và quản lý các thiết bị từ bất kỳ đâu, tạo ra sự thuận tiện và linh hoạt trong cuộc sống hàng ngày.

- Thu thập và phân tích dữ liệu: Thu thập dữ liệu từ các thiết bị và môi trường xung quanh một cách tự động và liên tục. Dữ liệu này sau đó có thể được phân tích để đưa ra thông tin quan trọng, xu hướng và dự đoán, giúp tối ưu hóa quyết định và quản lý hiệu quả.

- Tự động hóa quy trình: Tự động hóa các quy trình và hoạt động, giảm thiểu sự can thiệp của con người và tăng cường hiệu suất. Ví dụ, trong sản xuất, các hệ thống IoT có thể tự động điều chỉnh dây chuyền sản xuất dựa trên dữ liệu thu thập được từ các cảm biến.

- An toàn và an ninh: IoT cung cấp các giải pháp bảo mật tiên tiến để bảo vệ dữ liệu và thiết bị khỏi các mối đe dọa mạng. Việc sử dụng mã hóa, xác thực và giám sát bảo mật giúp ngăn chặn các cuộc tấn công và đảm bảo tính toàn vẹn của hệ thống.

- Cải thiện trải nghiệm người dùng: Tạo ra các trải nghiệm người dùng tốt hơn thông qua việc cung cấp các sản phẩm và dịch vụ thông minh và tùy chỉnh. Ví dụ, các hệ thống nhà thông minh có thể tự động điều chỉnh ánh sáng và nhiệt độ theo sở thích cá nhân của người dùng.

- Tạo ra các mô hình kinh doanh mới: Mở ra cơ hội cho các mô hình kinh doanh mới và sáng tạo. Việc kết hợp dữ liệu từ các thiết bị thông minh và phân tích thông tin có thể dẫn đến việc phát triển các sản phẩm và dịch vụ mới, cũng như tạo ra các mô hình kinh doanh dựa trên dịch vụ và sản phẩm kết nối.

2.1.4. Ứng dụng của IoT

- Nhà và văn phòng thông minh:

- + Tự động hóa và điều khiển các thiết bị như đèn, điều hòa, khóa cửa, camera an ninh, hệ thống giám sát.

- + Giúp tăng tiện nghi, đảm bảo an ninh và tiết kiệm năng lượng.

- Thiết bị đeo thông minh:
 - + Bao gồm đồng hồ thông minh, vòng tay sức khỏe, cảm biến thể chất.
 - + Thu thập dữ liệu về hoạt động, sức khỏe và vị trí của người dùng.
 - + Dữ liệu được gửi đến ứng dụng di động hoặc hệ thống y tế để giám sát sức khỏe và lối sống.
- Ô tô tự lái:
 - + IoT cung cấp dữ liệu từ các cảm biến và hệ thống điều khiển cho xe.
 - + Hệ thống tự lái sử dụng dữ liệu này để nhận biết môi trường, giao thông và vị trí.
 - + Đảm bảo an toàn và hỗ trợ lái xe tự động hiệu quả.
- Phân tích dữ liệu lớn (Big Data Analytics):
 - + IoT tạo ra lượng dữ liệu khổng lồ từ các thiết bị và cảm biến.
 - + Dữ liệu được phân tích để phát hiện xu hướng, mô hình dự đoán và tối ưu hóa quyết định kinh doanh.
- Công nghiệp sản xuất:
 - + Ứng dụng IoT trong dây chuyền sản xuất thông minh và tự động hóa.
 - + Cảm biến và hệ thống điều khiển giúp giám sát, phân tích và tối ưu quy trình.
 - + Nâng cao hiệu suất, giảm thiểu lãng phí và chi phí vận hành.
- Truyền tin và cảnh báo sự cố:
 - + IoT giúp giám sát và phát hiện sớm các sự cố trong hệ thống hoặc thiết bị.
 - + Dữ liệu cảm biến được gửi đến máy chủ để cảnh báo cho người dùng hoặc cơ quan chức năng (như PCCC).
 - + Hỗ trợ xử lý nhanh chóng, giảm thiểu thiệt hại và rủi ro.

2.1.5. Tầm quan trọng của IoT

Tầm quan trọng của IoT được thể hiện ở các khía cạnh sau:

- **Tự động hóa quy trình:** IoT giúp tự động hóa các quy trình và hoạt động trong nhiều lĩnh vực, từ sản xuất đến quản lý tài nguyên và dịch vụ công cộng.

- **Thu thập và phân tích dữ liệu:** IoT cung cấp một nguồn lượng lớn dữ liệu từ các thiết bị và cảm biến, giúp các tổ chức thu thập và phân tích thông tin để đưa ra quyết định thông minh và dự đoán xu hướng tương lai. Điều này giúp cải thiện quản lý, tối ưu hóa quy trình và nâng cao hiệu suất.

- **Giảm thiểu chi phí:** Bằng cách tối ưu hóa sử dụng tài nguyên và quy trình, cùng việc giám sát và phát hiện sớm sự cố, IoT giúp giảm thiểu chi phí hoạt động. Ví dụ, trong sản xuất, việc sử dụng hệ thống IoT có thể dẫn đến giảm chi phí vận hành và bảo trì, cũng như giảm lãng phí và thất thoát.

2.2. Các kiến thức cơ bản

2.2.1. Phần mềm Arduino

2.2.1.1. Khái niệm

Arduino là nền tảng mã nguồn mở giúp người dùng xây dựng các ứng dụng điện tử có khả năng liên kết và tương tác với nhau một cách linh hoạt. Arduino được xem như một máy tính thu nhỏ, cho phép người dùng lập trình và thực hiện các dự án điện tử mà không cần công cụ chuyên biệt để nạp code.

2.2.1.2. Ứng dụng

Arduino được ứng dụng rộng rãi trong nhiều lĩnh vực nhờ khả năng dễ lập trình và tích hợp với nhiều loại cảm biến, động cơ và thiết bị khác.

- Tự động hóa nhà thông minh

+ Điều khiển đèn, quạt, cửa, thiết bị gia dụng qua Internet hoặc Bluetooth.

+ Cảm biến nhiệt độ, độ ẩm giúp điều chỉnh môi trường tự động.

- Dự án robot

+ Lập trình robot di chuyển, tránh vật cản, quét dọn, di chuyển đồ vật.

+ Điều khiển động cơ, servo để tạo chuyển động linh hoạt.

- Điều khiển LED và đèn RGB

+ Tạo hiệu ứng ánh sáng nghệ thuật, biểu diễn ánh sáng, hệ thống chiếu sáng thông minh.

+ Đồng bộ ánh sáng với âm thanh hoặc thời gian thực.

- Thiết bị đo lường và cảm biến

+ Đo độ ẩm, nhiệt độ, ánh sáng, khí gas,... để giám sát môi trường.

+ Cảnh báo khi nồng độ khí nguy hiểm vượt ngưỡng.

- Ứng dụng y tế và sức khỏe

+ Đo nhịp tim, huyết áp, theo dõi hoạt động cơ thể.

+ Máy đo nồng độ cồn trong hơi thở.

- Hệ thống bảo mật và an ninh

+ Khóa cửa bằng RFID, vân tay, nhận diện khuôn mặt.

+ Cảm biến chuyển động phát hiện và cảnh báo đột nhập.

- Giáo dục và học tập

+ Dạy lập trình, điện tử, tự động hóa trong giáo dục STEM.

+ Các bộ kit Arduino hỗ trợ học sinh thực hành sáng tạo.

- Ứng dụng nông nghiệp

+ Hệ thống tưới tiêu tự động theo cảm biến độ ẩm và thời tiết.

+ Giám sát và điều chỉnh môi trường nhà kính.

2.2.1.3. Phần mềm Arduino IDE

Arduino IDE là môi trường phát triển tích hợp giúp viết, biên dịch và nạp code cho bo mạch Arduino. IDE hỗ trợ nhiều loại bo mạch như Uno, Nano, Mega, Pro Mini,...

Ngôn ngữ lập trình sử dụng là C/C++, có các tính năng như làm nổi bật cú pháp, tự động thụt dòng, cùng thư viện mẫu phong phú.

Phần mềm được cung cấp miễn phí và hiện có phiên bản mới nhất là Arduino 2.3.3.

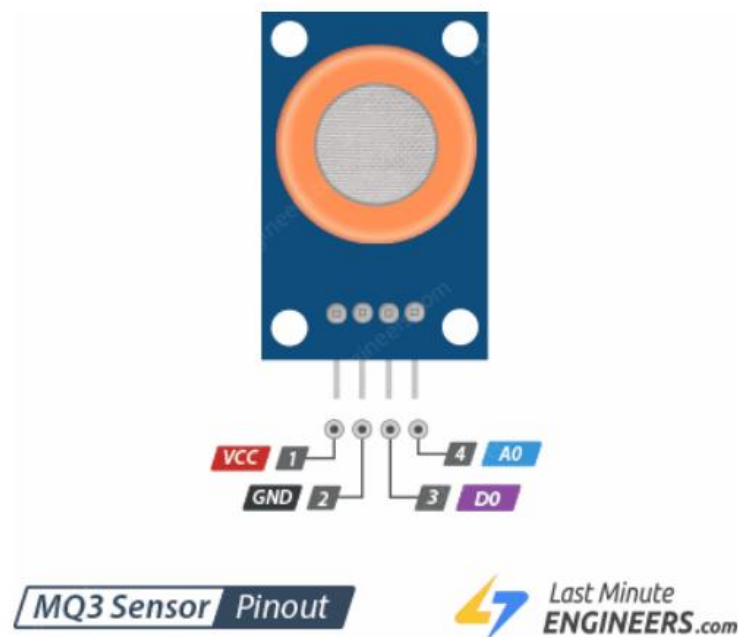
2.2.2. Giới thiệu về cảm biến khí rượu MQ-3

2.2.2.1. Khái niệm

Cảm biến MQ-3 là loại cảm biến chuyên dụng để phát hiện nồng độ rượu (ethanol) trong không khí, thường dùng trong các dự án máy đo nồng độ cồn.

2.2.2.2. Đặc điểm

- Độ nhạy cao với ethanol (C_2H_5OH).
- Thời gian phản hồi nhanh, phù hợp với ứng dụng đo theo thời gian thực.
- Ngõ ra tương tự (analog) tỉ lệ với nồng độ cồn và ngõ ra số (digital) kích hoạt khi vượt ngưỡng.



Hình 2.1 : Sơ đồ chân cảm biến MQ3

2.2.2.3. Cách hoạt động

Cảm biến sử dụng lớp màng oxit thiếc (SnO_2) để phát hiện ethanol. Khi có khí cồn, điện trở lớp này thay đổi, dẫn đến thay đổi tín hiệu ngõ ra.

2.2.2.4. Ứng dụng

- Máy đo nồng độ cồn trong hơi thở.
- Hệ thống giám sát nồng độ cồn trong môi trường dân dụng hoặc công nghiệp.

2.2.3. Giới thiệu về ESP32

2.2.3.1. Khái niệm

ESP32 là vi điều khiển SoC (System on Chip) của Espressif Systems, kế thừa từ ESP8266, tích hợp Wi-Fi và Bluetooth.



Hình 2.2 : Module ESP32

2.2.3.2. Đặc điểm nổi bật

- Vi xử lý mạnh mẽ: Lõi kép Tensilica Xtensa LX6 (240MHz), 520KB RAM.
- Kết nối không dây: Wi-Fi 802.11 b/g/n, Bluetooth 4.2 và BLE.
- Giao tiếp đa dạng: UART, SPI, I2C, I2S, PWM, DAC, ADC.
- Tiêu thụ điện năng thấp, có các chế độ Deep Sleep và Light Sleep.
- Bảo mật cao: Hỗ trợ mã hóa phần cứng và mạng.

2.2.3.3. Ứng dụng

- Nhà thông minh: Điều khiển thiết bị qua ứng dụng hoặc web.
- Dự án IoT: Giám sát và điều khiển thiết bị từ xa.
- Thiết bị đeo thông minh: Theo dõi sức khỏe, kết nối Bluetooth.

2.2.3.4. Ưu điểm so với Arduino Uno

- Mạnh hơn, có Wi-Fi/Bluetooth tích hợp.

- Bộ nhớ lớn hơn, xử lý đa nhiệm tốt.

2.2.3.5. So sánh ESP32 và ESP8266

Tiêu chí	ESP32	ESP8266
Số chân GPIO	34	16
Vi xử lý	Lõi kép Xtensa 160MHz	Lõi đơn
Tính năng	Có ADC, DAC, cảm biến chạm, bảo mật cao	Không có cảm biến chạm, bảo mật hạn chế
Kết nối	Wi-Fi + Bluetooth	Chỉ Wi-Fi
Ứng dụng	IoT, nhà thông minh, điều khiển từ xa	Thiết bị Wi-Fi giá rẻ

Bảng 2.1: So sánh ESP32 và ESP8266

2.2.4. LCD 16x2

- LCD (Liquid Crystal Display) là màn hình tinh thể lỏng dùng để hiển thị ký tự hoặc số liệu.

- LCD 16×2 có 16 chân, gồm 8 chân dữ liệu (D0–D7) và 3 chân điều khiển (RS, RW, EN).

- Có thể hoạt động ở chế độ 4-bit hoặc 8-bit.

- Được dùng rộng rãi trong các dự án Arduino để hiển thị thông tin.



Hình 2.3: LCD 16x2

2.2.5. Module I2C Arduino

LCD có nhiều chân nên việc kết nối phức tạp; Module I2C giúp giảm số chân kết nối chỉ còn 2 chân (SCL, SDA).

- Tương thích với các LCD sử dụng driver HD44780.

Thông số kỹ thuật:

- Điện áp: 2.5–6V DC
- Giao tiếp: I2C
- Địa chỉ mặc định: 0x27
- Có biến trở điều chỉnh độ tương phản, jumper điều khiển đèn nền.

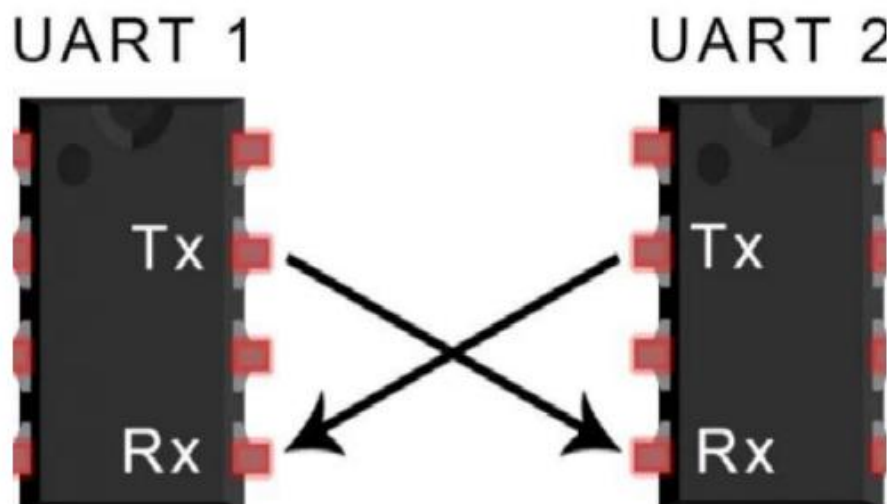


Hình 2.4: Sơ đồ chân module I2C

2.2.6. Các giao thức truyền thông dữ liệu và kết nối

2.2.6.1. Giao thức UART (Universal Asynchronous Receiver-Transmitter)

UART là giao thức truyền thông nối tiếp, gồm 2 dây: TX (truyền) và RX (nhận). Không cần đồng hồ chung (asynchronous), dữ liệu truyền theo từng byte gồm các bit Start, Data, Parity, Stop.



Hình 2.5: Giao thức UART

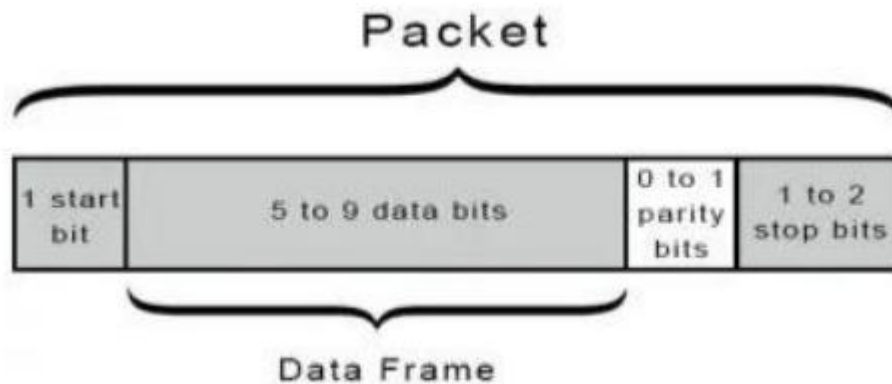
Ưu điểm:

- Đơn giản, dễ sử dụng.
- Ứng dụng rộng rãi trong IoT giữa vi điều khiển và cảm biến.

Nhược điểm:

- Chỉ kết nối 1-1 giữa hai thiết bị.
- Không tự sửa lỗi, phụ thuộc phần mềm hoặc giao thức khác như RS-232, RS-485, USB.

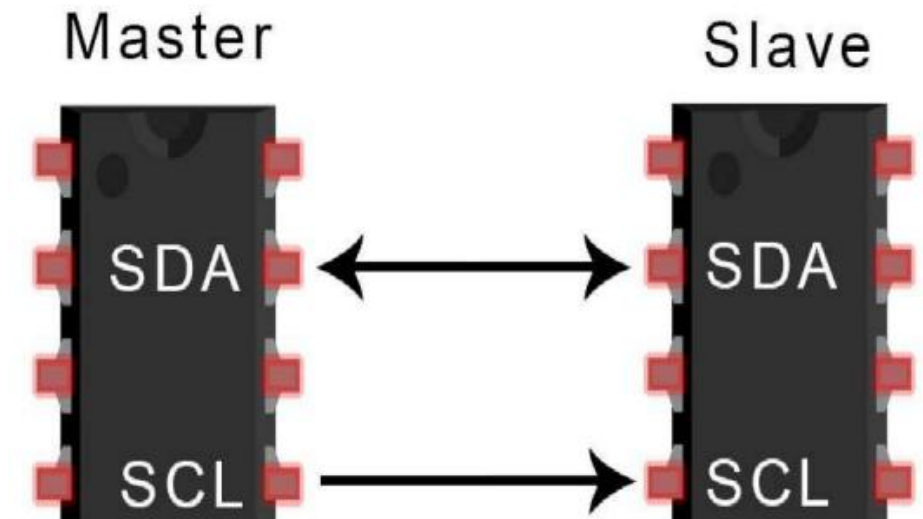
Cách thức hoạt động



Cách hoạt động của UART là thông qua việc truyền và nhận các khối dữ liệu từng byte theo từng chu kỳ. Dữ liệu được truyền qua dây TX dưới dạng các tín hiệu điện mức cao và thấp để biểu diễn các bit 0 và 1. Dây RX nhận các tín hiệu này và giải mã chúng để nhận dữ liệu.

UART sử dụng cơ chế bất đồng bộ (asynchronous) để đồng bộ truyền nhận dữ liệu giữa các thiết bị. Điều này có nghĩa là không có một tín hiệu đồng hồ chung giữa hai thiết bị để đồng bộ hoạt động. Thay vào đó, các thiết bị UART sử dụng các bit Start (bắt đầu), Data (dữ liệu), Parity (kiểm tra chẵn lẻ) và Stop (kết thúc) để đồng bộ truyền nhận dữ liệu.

2.2.6.2. Giao thức I2C (Inter-Integrated Circuit)



Hình 2.6: Giao thức I2C

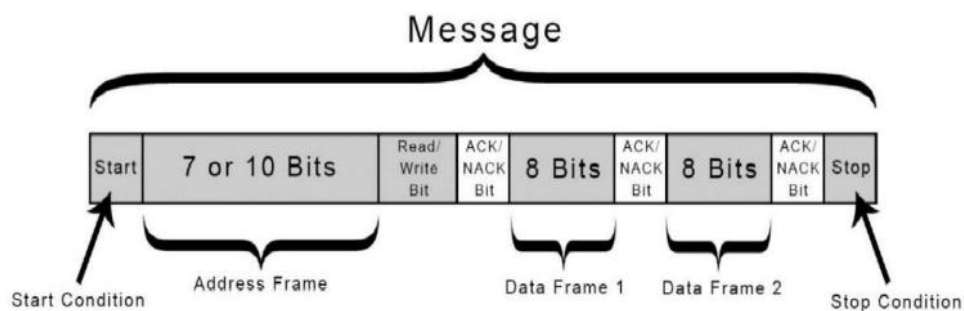
Giao thức I2C cho phép nhiều thiết bị giao tiếp qua 2 dây:

- SDA (Serial Data)
- SCL (Serial Clock)

Thành phần chính:

- Master: Thiết bị điều khiển truyền thông.
- Slave: Các thiết bị được Master điều khiển.

Cách hoạt động:



1. Khởi tạo: Thiết bị Master khởi tạo và kiểm tra mạng I2C để xác định xem các thiết bị Slave nào có sẵn trong mạch

2. Gửi địa chỉ: Master chọn một thiết bị Slave bằng cách gửi địa chỉ của nó qua dây dữ liệu SDA. Mỗi thiết bị Slave có một địa chỉ duy nhất để phân biệt chúng

3. Giao tiếp dữ liệu: Sau khi thiết bị Slave được chọn, Master và Slave giao tiếp qua dây dữ liệu SDA. Master gửi hoặc yêu cầu nhận dữ liệu từ thiết bị Slave thông qua các tín hiệu điều khiển trên dây đồng hồ SCL.

4. Có thể chuyển đổi vai trò khi cần. Điều này cho phép các thiết bị Slave truyền dữ liệu hoặc yêu cầu dữ liệu từ Master

Ưu điểm:

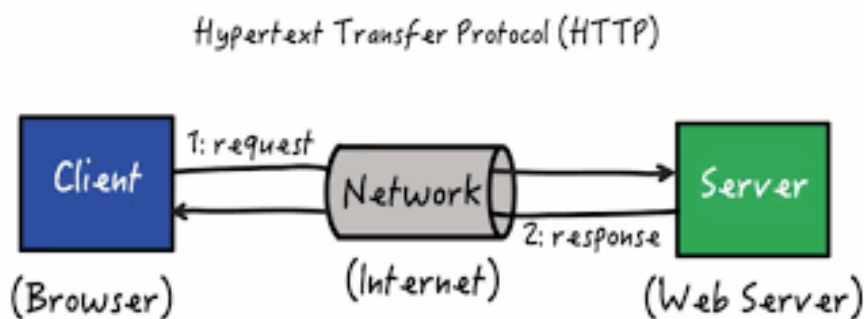
- Sử dụng ít chân GPIO.
- Hỗ trợ nhiều thiết bị.
- Dễ sử dụng và triển khai.

Nhược điểm:

- Tốc độ truyền thấp hơn SPI/UART.
- Khoảng cách truyền ngắn.
- Có độ trễ do cơ chế xác nhận dữ liệu.

2.2.6.3. Giao thức HTTP (HyperText Transfer Protocol)

HTTP là giao thức truyền thông ở tầng ứng dụng, được dùng rộng rãi trong các hệ thống IoT có kết nối Internet. Dữ liệu trao đổi theo mô hình **Client – Server**, trong đó Client gửi yêu cầu (request) và Server trả về kết quả (response).



Hình 2.7: Giao thức HTTP

- Ưu điểm:

- + Phổ biến và tiêu chuẩn hóa, dễ tích hợp với web, mobile, API.

- + Hoạt động qua WiFi/Internet, phù hợp cho IoT kết nối từ xa.
- + Hỗ trợ nhiều định dạng dữ liệu như JSON, XML, text.
- + Tương thích với nhiều nền tảng (ESP32, Raspberry Pi, Java, Python...)
- Nhược điểm:
 - + Độ trễ cao hơn so với giao thức nhúng như UART/I2C/SPI.
 - + Tiêu tốn năng lượng, không phù hợp cho thiết bị siêu tiết kiệm pin.
 - + Yêu cầu mạng Internet hoặc LAN, không dùng được ngoại tuyến.
- Cách thức hoạt động:
 - + Client gửi yêu cầu HTTP (GET/POST/PUT/DELETE) đến Server.
 - + Server xử lý yêu cầu, truy vấn dữ liệu hoặc thực thi nghiệp vụ.
 - + Server trả về phản hồi HTTP, gồm mã trạng thái (status code) và nội dung (body).
 - + Dữ liệu thường được truyền dưới dạng JSON, dễ parse trên ESP32 hoặc hệ thống backend.

2.2.6.4. Giao thức Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) là giao thức truyền thông không dây tầm ngắn, tối ưu hóa cho **tiết kiệm năng lượng**, rất phù hợp cho các thiết bị IoT cầm tay như cảm biến, đồng hồ thông minh, thiết bị đo sức khỏe, đo nồng độ cồn cầm tay.

- Ưu điểm:
 - + Rất tiết kiệm năng lượng, phù hợp thiết bị dùng pin.
 - + Tốc độ chấp nhận được cho dữ liệu cảm biến.
 - + Kết nối nhanh, hỗ trợ quảng bá và tự động kết nối lại.
 - + Không cần Internet, hoạt động offline.

- Nhược điểm:
 - + Phạm vi ngắn (5–15m).
 - + Không phù hợp truyền file lớn hoặc dữ liệu tốc độ cao.

- + Phải hiểu cấu trúc Service/Characteristic để lập trình.
- Cách hoạt động
 - + Peripheral phát quảng bá (advertising): gửi thông tin để Central tìm thấy.
 - + Central kết nối tới Peripheral thông qua kênh BLE.
 - + Trao đổi dữ liệu:
 - + Central đọc (Read) dữ liệu qua đặc tính (Characteristic).
 - + Peripheral gửi thông báo (Notify) khi dữ liệu thay đổi.
 - + Hoặc Central ghi (Write) dữ liệu vào đặc tính.
 - + Ngắt kết nối khi không còn trao đổi để tiết kiệm pin

3. Phân tích yêu cầu chức năng

3.1. Chức năng đăng nhập vào hệ thống

- Giao diện: Giao diện đăng nhập phải bao gồm hai ô điền thông tin: Tên tài khoản (username) và Mật khẩu (password), cùng với nút Đăng nhập để xác thực truy cập Web.
- Cơ sở Dữ liệu: Dữ liệu về tài khoản người dùng phải được lưu trữ trong CSDL (Database) với các thuộc tính cần thiết để quản lý nhân sự: id (Khóa chính), username, password (đã được mã hóa), hoten.
- Luồng xử lý:
 - + Thành công: Sau khi kiểm tra username và password khớp với dữ liệu đã mã hóa, Server sẽ tạo và trả về một JSON Web Token (JWT). Web sẽ chuyển hướng người dùng tới Giao diện chính (Dashboard).
 - + Thất bại: Hệ thống sẽ hiển thị thông báo chung là Sai tên tài khoản hoặc mật khẩu và giữ người dùng tại giao diện đăng nhập để bảo mật thông tin tài khoản.

3.2. Chức năng đo nồng độ cồn và cảnh báo vi phạm

- Đây là chức năng tương tác cốt lõi tại hiện trường, cho phép lực lượng chức năng thực hiện đo lường và nhận phản hồi ngay lập tức.

- Luồng dữ liệu: Hơi thở → MQ-3 → ESP32 (Xử lý/Cảnh báo) → LCD/LED.
- Các thao tác đo lường:
 - + Kích hoạt đo: Thiết bị phải được kích hoạt đo lường bằng thao tác vật lý (nhấn nút Button) để đảm bảo quá trình lấy mẫu hơi thở chủ động.
 - + Đo và xử lý: ESP32 phải đọc tín hiệu Analog từ MQ-3, thực hiện thuật toán chuyển đổi sang BAC (mg/L) và hiển thị kết quả lên LCD trong vòng ≤ 3 giây.
 - + Cơ chế cảnh báo nhanh: Khi giá trị BAC vượt quá ngưỡng quy định, ESP32 phải ngay lập tức kích hoạt đồng thời đèn LED Đỏ nhấp nháy để thông báo về hành vi vi phạm.
 - + Yêu cầu phi chức năng: Độ trễ từ lúc kết thúc lấy mẫu hơi thở đến khi cảnh báo (LED) được kích hoạt phải nhỏ hơn 1 giây để đảm bảo tính kịp thời tại hiện trường.

3.3. Chức năng quản lý dữ liệu và truyền tải bảo mật

- Chức năng này chịu trách nhiệm thu thập, xác thực và lưu trữ dữ liệu đo lường, biến nó thành bằng chứng pháp lý.
- Luồng dữ liệu: ESP32 → BLE → Client Backend (Json) → HTTPS → Server Backend → Database.
- Quy trình truyền tải:
 - + ESP32 thực hiện đo dữ liệu và gửi dữ liệu đo đến hệ thống client.
 - + Gói Dữ liệu: Bên hệ thống phía client phải đóng gói kết quả đo, thông tin người được đo, và device ID thành gói JSON.
 - + Mã hóa truyền tải: Gói dữ liệu phải được gửi đến Server thông qua giao thức HTTPS để đảm bảo dữ liệu không bị can thiệp (man-in-the-middle) hoặc đọc trộm.
 - + Xác thực thiết bị: Server phải xác thực device ID và key đi kèm trong Header của gói tin để đảm bảo chỉ dữ liệu từ thiết bị hợp lệ mới được chấp nhận.
 - + Lưu trữ bằng chứng: Server phải ghi nhận bản ghi vi phạm vào Database, bao gồm cả mức vi phạm đã được phân loại tự động và device ID để truy vết nguồn gốc của bằng chứng.

3.4. Chức năng quản lý tập trung và phân tích thống kê

- Đây là các chức năng chạy trên Server và Giao diện Web (Dashboard), phục vụ cho cấp quản lý.

- Quản lý tài khoản và phân quyền:

+ Cho phép Quản trị viên thực hiện các thao tác: Thêm, Sửa, Xóa tài khoản người dùng và thiết lập Role-Based Access Control (RBAC).

- Quản lý thiết bị (Device Management):

+ Cho phép đăng ký thiết bị mới (gán ID duy nhất), theo dõi trạng thái hoạt động (Online/Offline) và tình trạng của từng thiết bị.

+ Cung cấp chức năng xem lịch sử hiệu chuẩn của mỗi thiết bị để đảm bảo tính pháp lý của kết quả đo.

- Thống kê và trực quan hóa dữ liệu:

+ Cung cấp Dashboard hiển thị tổng quan về tình hình vi phạm theo thời gian thực.

+ Cho phép truy vấn dữ liệu vi phạm theo các tham số: Device ID, thời gian, người đo.

+ Hiển thị biểu đồ phân bố vi phạm theo thời gian để hỗ trợ việc phân bổ lực lượng tuần tra.

3.5. Chức năng chuyển đổi và xử lý lỗi trạng thái

- Hệ thống cần đảm bảo việc chuyển đổi trạng thái và xử lý lỗi diễn ra mượt mà và an toàn.

- Quy trình Hiệu chuẩn (Calibration Process):

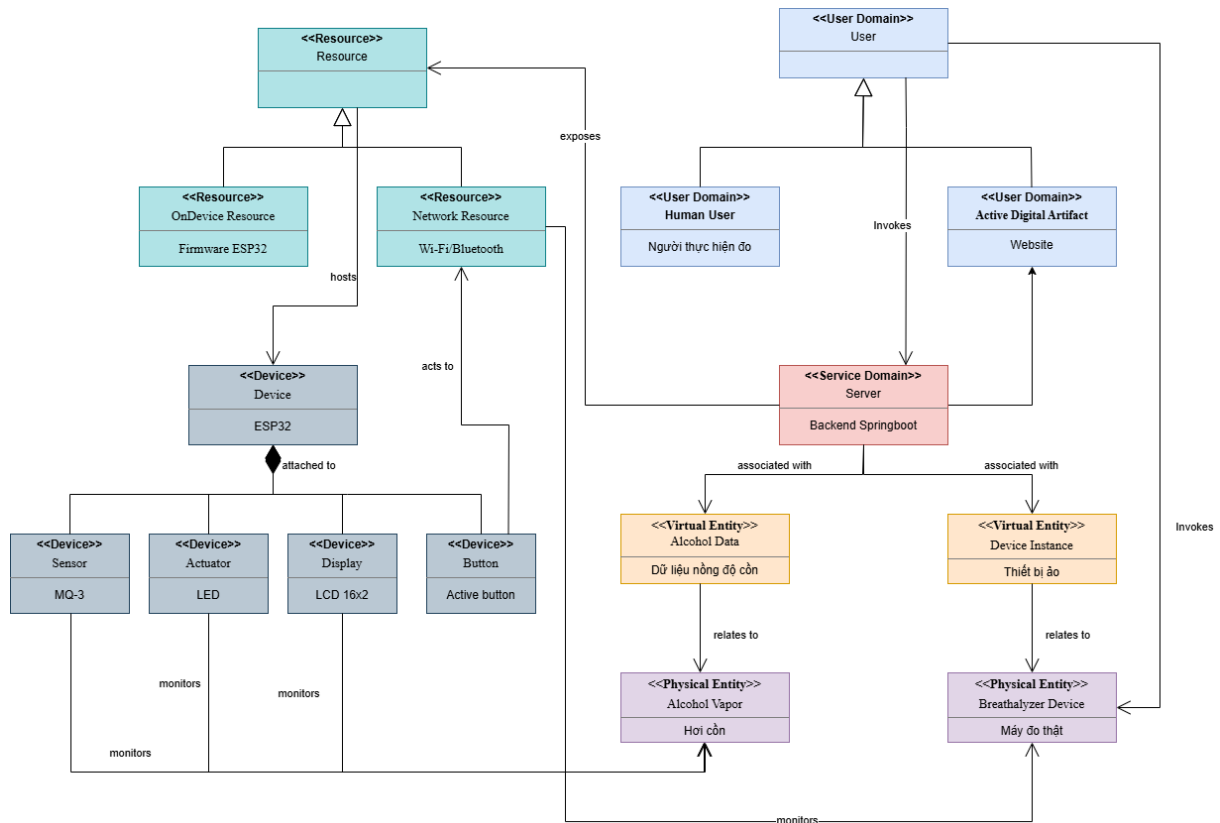
+ Thiết bị phải có một chế độ Hiệu chuẩn (Calibration Mode) để thiết lập lại giá trị R (Điện trở cơ sở trong không khí sạch) và cập nhật các tham số hiệu chuẩn.

- Xử lý lỗi kết nối:

+ Nếu thiết bị không thể kết nối Internet (Wi-Fi Offline), nó phải chuyển sang chế độ lưu trữ cục bộ (Local Storage). Khi kết nối được phục hồi, dữ liệu phải được tự động đồng bộ lên Server theo thứ tự thời gian.

4. Thiết kế hệ thống

4.1. Đặc tả mô hình miền



Hình 4.1: Sơ đồ đặc tả mô hình miền

4.1.1. Miền người dùng (User Domain)

- User (Người dùng): Là thực thể tương tác với hệ thống để thực hiện đo lường hoặc giám sát dữ liệu vi phạm.

- Human User: Là lực lượng chức năng thực hiện kiểm tra nồng độ cồn tại hiện trường.

- Active Digital Artifact: Là giao diện Web Dashboard (JavaSwing) dùng để: đăng nhập, quản lý thiết bị, xem dữ liệu vi phạm, và trực quan hóa kết quả. Đây là cầu nối giữa người dùng và máy chủ dữ liệu

4.1.2. Miền ảo (Virtual Entity)

Đây là vùng của các thực thể ảo đại diện cho những đối tượng vật lý trong thế giới thực, giúp số hóa và xử lý logic nghiệp vụ.

- Alcohol Data (Dữ liệu cồn): Đại diện cho giá trị BAC đo được và Violation Level (Mức vi phạm). Đây là các biến số trong phần mềm

- Device Instance (Thiết bị ảo): Đại diện cho trạng thái số hóa của thiết bị cầm tay, bao gồm các thuộc tính như Device ID, Trạng thái, và lần đo cuối

4.1.3. Miền vật lý (Physical Entity)

Đây là các thực thể vật lý tương ứng với các thực thể ảo, là các đối tượng thực mà hệ thống có thể giám sát và điều khiển

- Alcohol Vapor (Hơi cồn): Là thành phần vật lý được cảm biến MQ-3 thu nhận, đại diện cho nồng độ cồn trong hơi thở.

- Breathalyzer Device: Chiếc máy đo nồng độ cồn cầm tay, là đối tượng vật lý được quản lý và theo dõi

4.1.4. Miền dịch vụ (Service Domain)

- Đại diện cho Server Backend và các dịch vụ chạy nền.

- + Dịch vụ xử lý logic nghiệp vụ, bao gồm: Xác thực, phân loại mức vi phạm, và lưu trữ bản ghi vào CSDL.

- + Dịch vụ nhận dữ liệu từ thiết bị, xử lý logic, và cung cấp API truy vấn thống kê dữ liệu cho ứng dụng

4.1.5. Miền tài nguyên (Resource)

Đây là miền có các tài nguyên mạng hoặc tài nguyên thiết bị cung cấp cho dịch vụ.

- Network Resource: Bao gồm kết nối Wi-Fi/Bluetooth và giao thức REST API (HTTP/HTTPS) giúp truyền tải dữ liệu vi phạm từ thiết bị lên Server.

- OnDevice Resource: Là mã nguồn (Firmware C++) chạy trên ESP32. Đây là nơi chứa các thuật toán quan trọng như: logic hiệu chuẩn, logic cảnh báo cục bộ và quản lý kết nối Bluetooth/Wi-Fi

4.1.6. Miền thiết bị (Device)

Đây là phần cứng cốt lõi của thiết bị cầm tay, được tổ chức theo vi điều khiển làm trung tâm:

- Device (Thiết bị chính): Là vi điều khiển ESP32. Nó đóng vai trò là bộ não trung tâm, chịu trách nhiệm xử lý dữ liệu, kết nối mạng và điều phối tất cả các thành phần khác.

- Các thành phần gắn kèm:

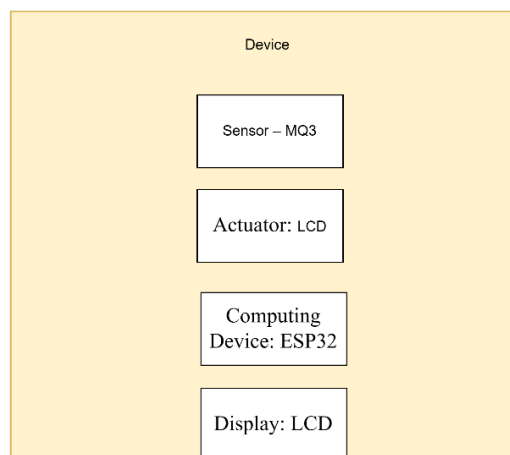
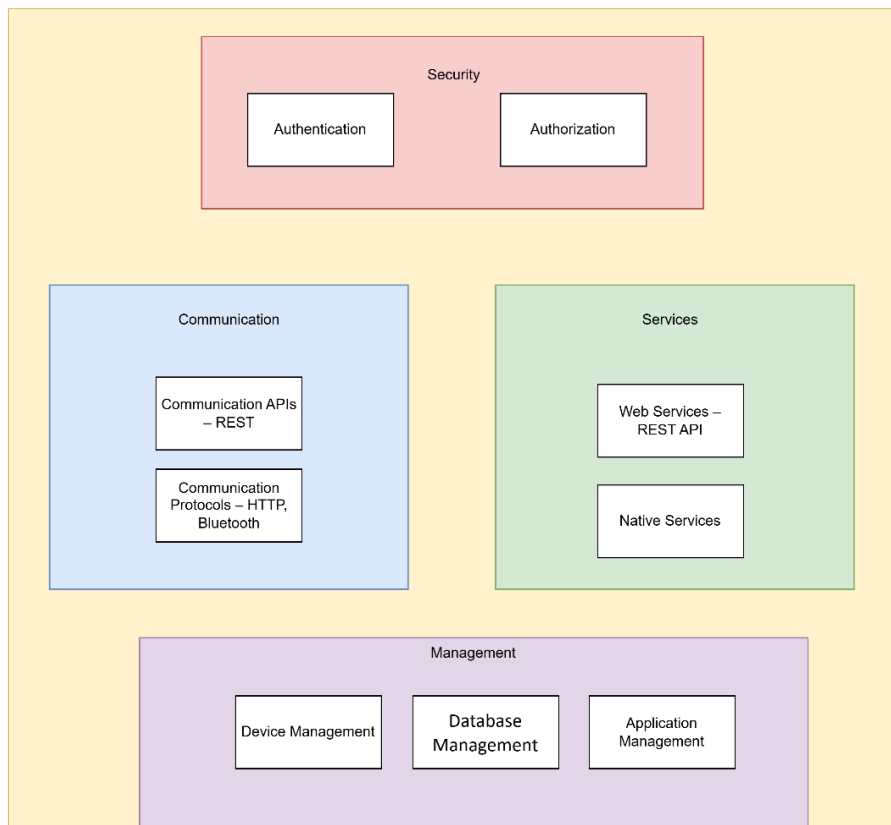
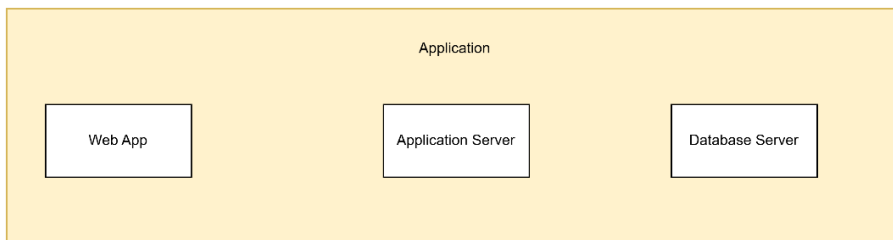
+ Sensor (MQ-3): Cảm biến khí MQ-3 được gắn vào ESP32 để đo nồng độ cồn.

+ Actuator (Indicator): LED Đỏ/Xanh. ESP32 gửi tín hiệu điều khiển để thực hiện hành động vật lý (cảnh báo) khi vượt ngưỡng.

+ Display (LCD 16x2): Màn hình hiển thị kết quả đo trực tiếp tại thiết bị.

+ Activate Button: Nút nhấn được gắn vào ESP32 để khởi động hoặc reset hệ thống.

4.2. Đặc tả các thành phần chức năng và hoạt động



Hình 4.2: Đặc tả thành phần chức năng

4.2.1 Thành phần chức năng

Phần này mô tả sự ánh xạ từ các thiết bị vật lý, tài nguyên phần mềm và các chức năng trong hệ thống đo nồng độ cồn vào các khối chức năng trong kiến trúc tổng quát.

- **Thiết bị IoT (IoT Devices)** : Được ánh xạ tới **khối Device**. Bao gồm các thành phần phần cứng chính của hệ thống:

- + Cảm biến: **MQ-3** dùng để đo nồng độ cồn trong hơi thở.
- + Vi điều khiển: **ESP32** đóng vai trò xử lý trung tâm và truyền thông Bluetooth.
- + Thiết bị hiển thị và cảnh báo: **LCD 16x2, LED đỏ/xanh**.
- + Nút nhấn (**Button**) để kích hoạt hoặc chuyển chế độ hoạt động.

- **Quản lý thiết bị (Device Management)**: Được ánh xạ tới **khối Management**.

- + Server thực hiện quản lý danh sách thiết bị, định danh thiết bị, ghi log dữ liệu đo.
- + ESP32 được lập trình và cập nhật thông qua Arduino IDE.

- **Tài nguyên (Resources)**: được ánh xạ tới **khối Management**

+ **OnDevice Resource (thiết bị)**: ESP32 xử lý tín hiệu từ MQ-3, điều khiển LCD/LED/Buzzer và gửi dữ liệu qua Bluetooth.

+ **Network Resource (mạng)**: REST API truyền dữ liệu giữa Web App và Server; Bluetooth giao tiếp giữa ESP32 và ứng dụng.

- **Dịch vụ điều khiển – Controller Services**: Được ánh xạ vào **khối Services**.

- + Thuật toán xử lý tín hiệu từ MQ-3 trên ESP32.
- + Dịch vụ phân tích và lưu trữ dữ liệu BAC chạy trên Server (Flask + PostgreSQL).
- + Chức năng cảnh báo mức vi phạm.

- **Dịch vụ Web (Web Services)**: Ánh xạ tới **khối Services**

- + RESTful API phục vụ Web App và các module quản lý.
- + Dịch vụ xử lý yêu cầu của thiết bị (gửi dữ liệu đo, đăng ký thiết bị, lấy lịch sử).
- **Cơ sở dữ liệu (Database):** Được ánh xạ vào **Management** (quản lý dữ liệu) và **Security** (quản lý thông tin người dùng):
 - + PostgreSQL lưu thông tin người dùng, thiết bị và lịch sử đo nồng độ cồn.
- **Ứng dụng Web – Web App:** Ánh xạ vào khối **Application**
 - + Desktop Web App được phát triển bằng **Java Swing**.
 - + Tương tác với Server qua REST API.
 - + Kết nối với thiết bị qua Bluetooth.

4.2.2 Thành phần hoạt động

Phần này liệt kê công nghệ, phần mềm và giao thức giúp hiện thực hóa các thành phần chức năng của hệ thống.

Application (Tầng Ứng dụng)

- Web
 - + Giao diện người dùng được xây dựng bằng **Java Swing** (Desktop Application).
 - + Hiện thị dữ liệu đo, cảnh báo, biểu đồ thống kê.
- Application Server
 - + Xây dựng bằng SpringBoot (Java)
 - + Chịu trách nhiệm xử lý API, lưu trữ dữ liệu và xác thực người dùng.
- Database Server
 - + Sử dụng **PostgreSQL**, lưu trữ log đo BAC, thông tin người dùng, thiết bị.

Management (Khối Quản lý)

- Application Management

- + Được hiện thực bằng Java, quản lý luồng nghiệp vụ backend
- Database Management: Quản lý kết nối truy vấn với cơ sở dữ liệu
- Device Management
 - + Quản lý kết nối thiết bị ESP32.
 - + Xác minh thiết bị, ghi log đo lường.

Services (Khối dịch vụ)

- Native Services
 - + Dịch vụ xử lý dữ liệu cảm biến MQ-3 trên **ESP32**
 - + Dịch vụ cảnh báo (LED) ngay trên thiết bị
- Web Services
 - + RESTful API: đăng nhập, đăng ký thiết bị, ghi nhận dữ liệu đo, truy vấn lịch sử.

Communication (Khối giao tiếp)

- Communication APIs
 - + REST API giữa Web App ↔ Server.
- Communication Protocols
 - + **Bluetooth**: ESP32 → Java Swing App.
 - + **HTTP/REST**: Web App ↔ Server.
 - + **UART/I2C**: giao tiếp nội bộ giữa ESP32 – LCD – cảm biến.

Security (Khối bảo mật)

- Authentication
 - + Xác thực người dùng qua REST API (Login/Register).
- Authorization

+ Phân quyền quản trị viên / người dùng thường.

Device (Tầng Thiết bị)

- Sensors: Cảm biến MQ-3.
- Actuator: LED đỏ/xanh cho cảnh báo
- Computing Device: ESP32 xử lý dữ liệu và truyền Bluetooth.
- Display: LCD 16x2 hiển thị BAC và trạng thái

4.3 Đặc tả luồng hoạt động

4.3.1 Thiết bị cảm biến và vi xử lý

a. Thư viện sử dụng

Thư viện	Chức năng
Arduino.h	Thư viện chính điều khiển vi điều khiển
MQSensor.h / MQ-3 library	Đọc giá trị từ cảm biến nồng độ cồn MQ-3
Wire.h	Giao tiếp I2C cho màn hình LCD hoặc module RTC
LiquidCrystal_I2C.h	Hiển thị dữ liệu đo trên LCD
ESP8266WiFi.h	Kết nối mạng WiFi
HTTPClient.h	Gửi dữ liệu lên máy chủ qua API

Bảng 4.1: Bảng mô tả thư viện

b. Các hàm chức năng

thư viện	Chức năng
setupSensor()	Khởi tạo cảm biến nồng độ cồn và các chân ADC
readAlcohol()	Đọc giá trị từ MQ-3, quy đổi sang mg/L hoặc %BAC

displayValue()	Hiển thị kết quả đo lên LCD hoặc LED
connectWiFi()	Kết nối WiFi và kiểm tra trạng thái
sendDataToServer()	Gửi dữ liệu đo (thời gian, giá trị, ID thiết bị) lên website quản lý
alarmHandler()	Kích hoạt còi/buzzer khi nồng độ vượt ngưỡng cho phép
loopFlow()	Vòng lặp chính: đo → hiển thị → gửi dữ liệu → cảnh báo

Bảng 4.2: Bảng mô tả các hàm

c. Luồng hoạt động

- Vi điều khiển khởi động và chạy hàm setup().
- Thiết bị kết nối mạng WiFi.
- Cảm biến MQ-3 được nung nóng và vào trạng thái ổn định.
- Người dùng thổi vào cảm biến → MQ-3 tạo tín hiệu analog.
- Vi điều khiển đọc tín hiệu ADC và chuyển đổi thành nồng độ cồn.
- Giá trị được hiển thị trên LCD.
- Nếu vượt ngưỡng: còi cảnh báo bật.
- Dữ liệu được gửi lên máy chủ qua API.
- Máy chủ lưu trữ và hiển thị trên website quản lý

4.3.2 Website quản lý

a. Thư viện sử dụng

Thư Viện	Chức năng
Springboot	Framework Java giúp xây dựng ứng dụng web nhanh chóng, tự động cấu hình (auto-configuration), hỗ trợ tạo API RESTful đơn giản và dễ mở rộng
Java Swing	Thư viện được sử dụng để phát triển giao diện người dùng trên website, hiển thị thông tin đo lường, thống kê vì phạm...

Bảng 4.3 :Bảng thư viện website

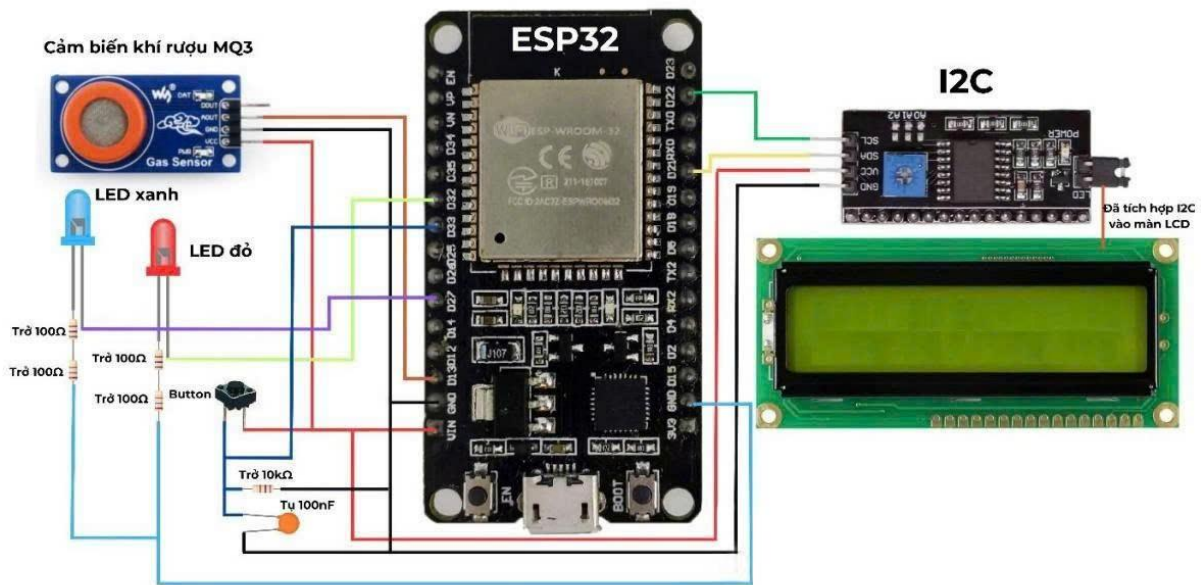
b. Các API sử dụng

stt	Mô tả	Endpoint	Phương thức	Tham số	Trả về
1	đăng nhập	api/auth/login	POST	username(String) , password(String)	token xác thực jwt để truy cập hệ thống
2	lấy danh sách thông tin thiết bị	/api/devices/{deviceId}	GET	không có	danh sách các thiết bị
3	lấy danh sách đo lường	/api/measurements	GET	phân trang từ trang 1-10	trả về succes hoặc fail
4	thống kê thiết bị	/api/devices/statistics	GET	không có	message (“success” hoặc “fail”)

5	kiểm tra thiết bị	/api/devices /check/{dev iceId}	GET	không có	message ("success"/"fail"), deviceInfo
---	-------------------	---------------------------------------	-----	----------	--

Bảng 4.4: Bảng mô tả các endpoint

4.4 Tích hợp Thiết bị



Hình 4.3: Sơ đồ mạch

Danh sách các linh kiện đã được tích hợp vào ESP32

Linh kiện	Tên chân (PIN)	Nối vào ESP32	Chức năng
Cảm biến khí rượu MQ3	AO	GPIO 13	Gửi dữ liệu nồng độ cồn sang ESP32
	GND	GND	Chân nguồn âm
	VCC	VIN	Chân nguồn dương
I2C (đã được tích hợp với màn LCD)	VCC	VIN	Chân nguồn dương
	GND	GND	Chân nguồn âm

	SDA	GPIO 21	Dây xung nhịp
	SCL	GPIO 22	Gửi/nhận dữ liệu từ ESP32
LED đỏ	Cực dương	GPIO 32	
	Cực âm	GND	
LED xanh	Cực dương	GPIO 27	
	Cực âm	GND	

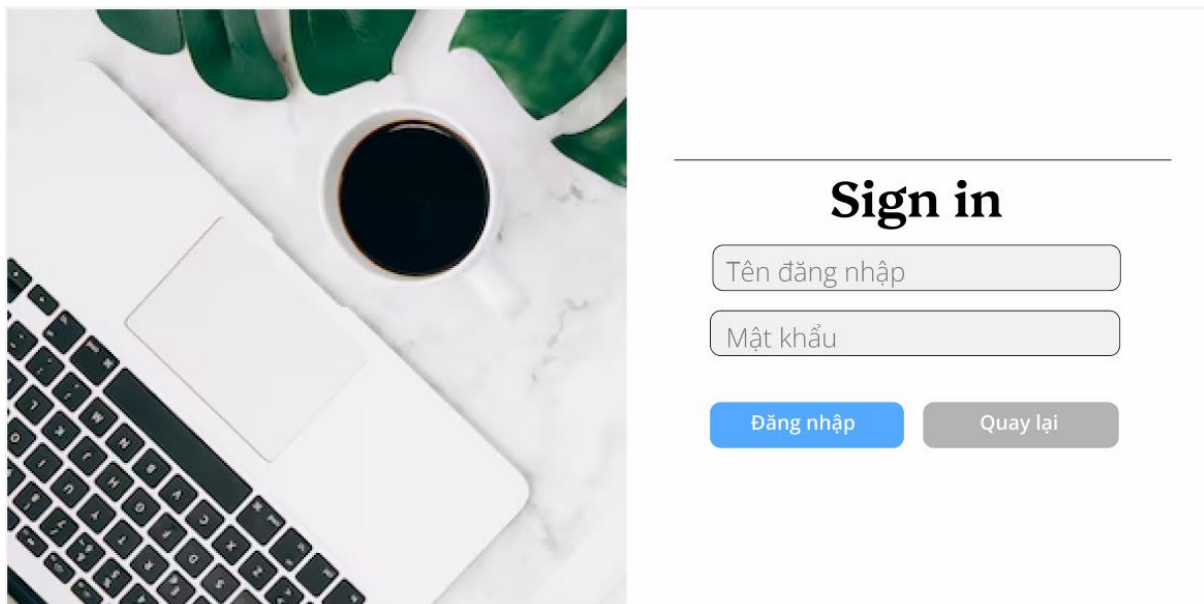
Bảng 4.5: Danh sách linh kiện

4.5 Phát triển ứng dụng

4.5.1. Giao diện đăng nhập chung

Trước khi sử dụng hệ thống cần thực hiện đăng nhập để có thể truy cập vào hệ thống quản lý, nếu sai tài khoản hoặc mật khẩu hệ thống sẽ kiểm tra và gửi thông báo sai tài khoản hoặc mật khẩu.

Nếu đăng nhập thành công, hệ thống đưa người dùng đến trang thích hợp dựa theo quyền hạn của người dùng.



Hình 4.4: Giao diện đăng nhập

4.5.2. Giao diện người trực tiếp đo

a. Giao diện kết nối thiết bị

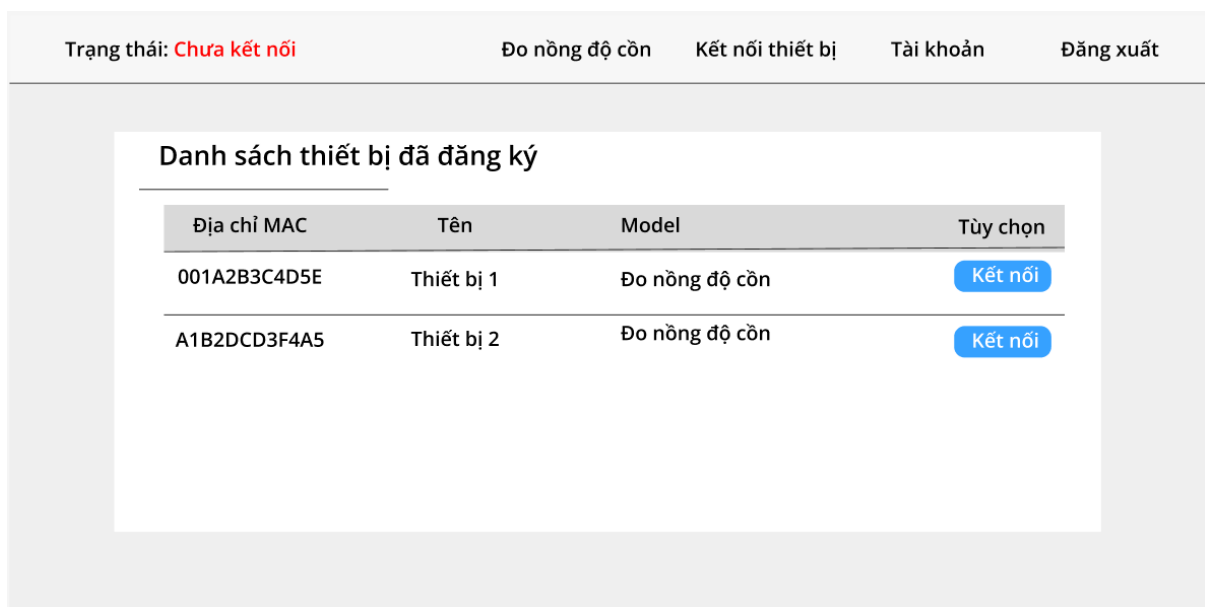
- Sau khi đăng nhập, để thực hiện đo nồng độ cồn, người dùng cần bật bluetooth máy tính và click kết nối thiết bị.

- Hệ thống quét một lượt các thiết bị đang chờ kết nối và hiện danh sách các thiết bị được quét và đã được đăng ký trước đó. Giao diện bao gồm:

+ Danh sách các thiết bị bao gồm các thuộc tính: địa chỉ MAC, Tên, Model

+ Nút kết nối cho mỗi thiết bị

- Người dùng ấn kết nối để thực hiện kết nối với thiết bị tương ứng.



Hình 4.5: Giao diện kết nối

b. Giao diện đo nồng độ cồn

- Để thực hiện đo nồng độ cồn, người dùng ấn vào chức năng đo nồng độ cồn sau đó ấn nút trên thiết bị để thực hiện đo nồng độ cồn.

- Sau 5 giây từ khi nhả nút, thiết bị gửi kết quả cho máy tính và hệ thống hiện form để điền dữ liệu bao gồm:

+ Kết quả đo được và mức vi phạm

- + Form điền thông tin gồm các trường thông tin: Họ tên, tuổi, CCCD, quê quán
- + Nút: Gửi dữ liệu, đo lại, quay lại
- Người dùng có thể click gửi để gửi dữ liệu lên hệ thống hoặc click đo lại để thực hiện đo lại nồng độ cồn.

Hình 4.6: Giao diện đo nồng độ cồn

4.5.3. Giao diện quản lý

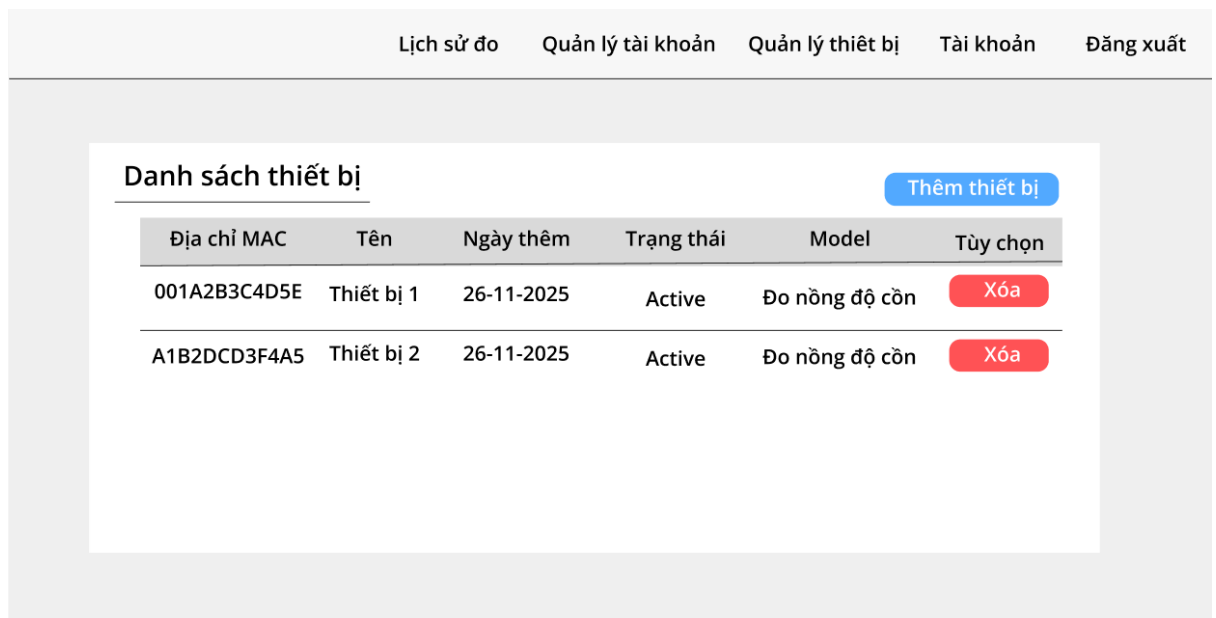
a. Giao diện quản lý thiết bị

- Sau khi đăng nhập, người quản lý click quản lý thiết bị. Hệ thống hiển thị giao diện bao gồm:

- + Danh sách thiết bị bao gồm các thuộc tính: Địa chỉ MAC, tên thiết bị, ngày thêm, trạng thái, model.

- + Nút thêm thiết bị và xóa.

- Người quản lý có thể ấn thêm thiết bị để thực hiện đăng ký cho thiết bị mới hoặc xóa thông tin thiết bị hiện tại.



Hình 4.7: Giao diện quản lý thiết bị

- Để thực hiện đăng ký thêm thiết bị, người quản lý phải bật bluetooth trên máy tính và click thêm thiết bị. Hệ thống hiển thị danh sách các thiết bị chưa được đăng ký quét được

- Người quản lý chọn một thiết bị, hệ thống hiển thị giao diện đăng ký cho thiết bị. Giao diện bao gồm:

- + Địa chỉ MAC của thiết bị được cố định
 - + Các trường điền thông tin: tên, model
 - + Nút thêm, quay lại
- Người quản lý điền thông tin thiết bị và click thêm để hoàn tất đăng ký thiết bị.

Lịch sử đo Quản lý tài khoản Quản lý thiết bị Tài khoản Đăng xuất

Điền thông tin

Địa chỉ MAC

AB14ACD525

Tên

Thiết bị 3

Model

Đo nồng độ cồn

Thêm

Quay lại

Hình 4.8: Giao diện thêm thông tin

b. Giao diện quản lý tài khoản

- Sau khi người quản lý click quản lý tài khoản, hệ thống hiển thị giao diện quản lý tài khoản bao gồm:

+ Danh sách tài khoản bao gồm các trường thông tin: ID, họ tên, email, role, ngày tạo, username.

+ Các nút: thêm tài khoản, sửa/xóa tài khoản

Lịch sử đo Quản lý tài khoản Quản lý thiết bị Tài khoản Đăng xuất

Danh sách tài khoản

Thêm tài khoản

ID	Tên	Email	Role	Created at	Username	Tùy chọn
1	Nguyễn Văn A	admin@gmail.com	admin	26-11-2025	acount	<div>SửaXóa</div>
2	Nguyễn Văn B	police1@gmail.com	police	26-11-2025	abc	<div>SửaXóa</div>

Hình 4.9: Giao diện quản lý tài khoản

- Để thực hiện đăng ký tài khoản cho người dùng mới, người quản lý click vào thêm tài khoản. Hệ thống hiển thị giao diện bao gồm:

+ Các trường điền thông tin: Username, Password, confirm password, họ tên, email, role

+ Nút thêm, quay lại

- Người quản lý điền thông tin tài khoản và ấn nút thêm để hoàn tất việc tạo tài khoản.

Lịch sử đo Quản lý tài khoản Quản lý thiết bị Tài khoản Đăng xuất

Điền thông tin

Username

Họ tên

Password

Email

Confirm Password

Role

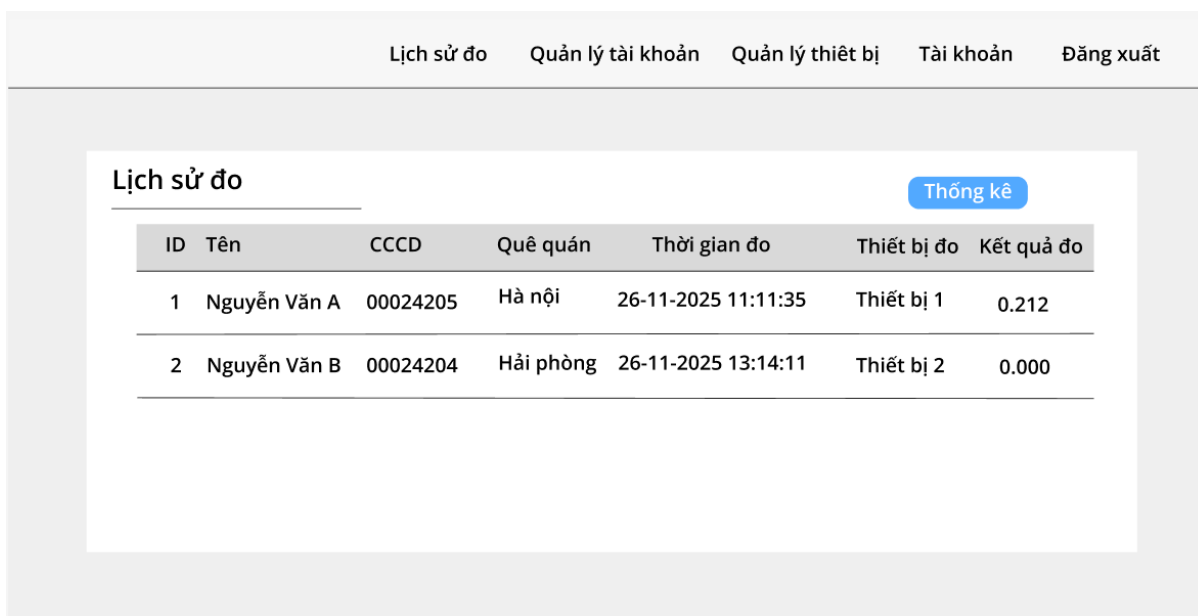
Hình 4.10: Giao diện thêm tài khoản

c. Giao diện lịch sử đo

- Người quản lý click lịch sử đo để xem lịch sử các lần đo. Hệ thống hiển thị giao diện bao gồm:

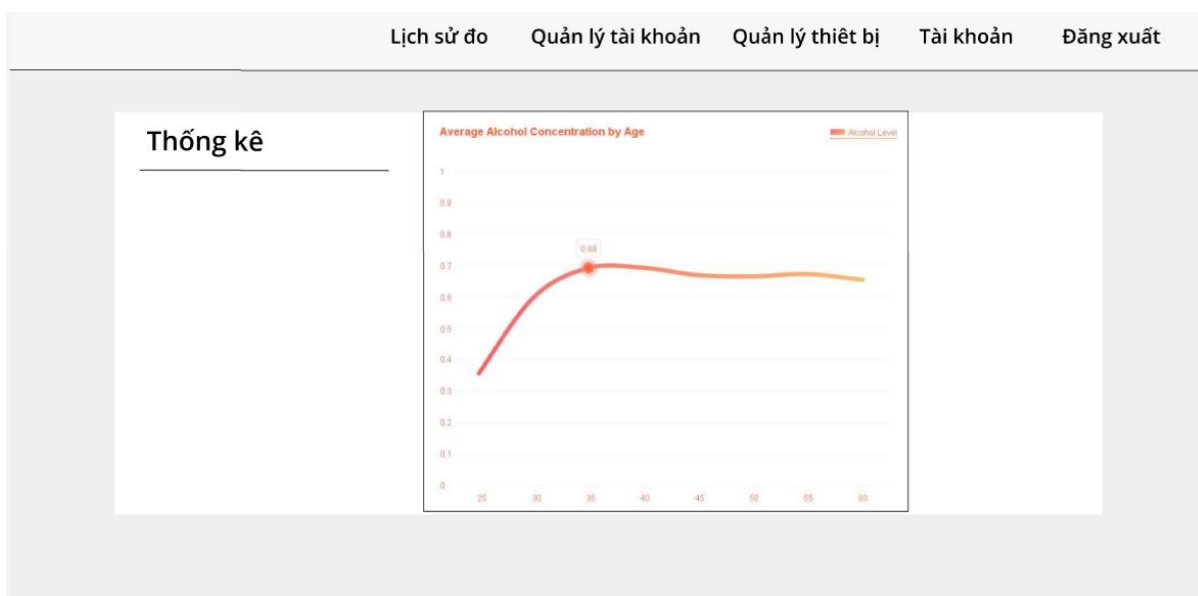
+ Danh sách lịch sử đo bao gồm các trường thông tin: ID, Tên, CCCD, quê quán, thời gian đo, thiết bị đo, kết quả đo.

+ Nút thống kê lịch sử đo



Hình 4.11: Giao diện lịch sử đo

- Người dùng click thống kê, hệ thống hiển thị giao diện bao gồm biểu đồ đường thống kê trung bình các lần đo dựa theo độ tuổi.



Hình 4.12: Giao diện thống kê

5. Kết luận

5.1. Kết quả đạt được

5.1.1. Về mặt lý thuyết

- Xây dựng Kiến trúc IoT: Đã nghiên cứu và xây dựng thành công kiến trúc hệ thống IoT (bao gồm cảm biến, bộ xử lý, giao thức truyền thông, và nền tảng đám mây/máy chủ) cho việc thu thập và quản lý dữ liệu nồng độ cồn.
- Tích hợp Cảm biến: Đã phân tích và lựa chọn thành công cảm biến cồn (ví dụ: cảm biến bán dẫn oxit kim loại hoặc tế bào nhiên liệu) và tích hợp nó vào vi điều khiển để số hóa dữ liệu
- Phát triển Thuật toán: Đã thiết lập thuật toán hiệu chuẩn và xử lý tín hiệu để đảm bảo độ chính xác và tốc độ đo, đáp ứng các tiêu chuẩn quy định về nồng độ cồn
- Xử lý dữ liệu: Xây dựng được quy trình lý thuyết về việc chuyển hóa từ điện áp sang nồng độ cồn trong hơi thở, và quy trình Backend xử lý, lưu trữ biên bản vi phạm và xác định mức phạt.

5.1.2. Về mặt thực tiễn

- Giám sát Từ xa: Thiết bị hoạt động ổn định và cho phép giám sát nồng độ cồn theo thời gian thực từ xa thông qua ứng dụng di động hoặc trang web (dashboard)
- Khả năng Kết nối: Thiết lập thành công kết nối Bluetooth giữa thiết bị ESP32 và ứng dụng Desktop (JavaSwing) , cho phép truyền dữ liệu BAC và nhận lệnh điều khiển từ xa.
- Lưu trữ Dữ liệu Lịch sử: Đã tạo được cơ sở dữ liệu để lưu trữ lịch sử các lần đo (bao gồm nồng độ cồn, thời gian, và vị trí đo), phục vụ cho việc thống kê và truy vết
- Cảnh báo Tự động: Hệ thống đã tự động gửi cảnh báo đèn LED khi nồng độ cồn đo được vượt quá ngưỡng an toàn đã cài đặt
- Hệ thống Backend Vận hành: Xây dựng thành công hệ thống Backend (sử dụng Springboot/PostgreSQL) với các API hoàn chỉnh để đăng ký thiết bị, ghi lại thông tin đo, xác định mức vi phạm và quản lý người dùng

5.2. Hạn chế

- Tính di động/Năng lượng: Hệ thống cần tối ưu hóa về mặt tiêu thụ năng lượng, đặc biệt khi sử dụng trong thời gian dài (cần tối ưu hóa chế độ Deep Sleep trên ESP32).

- Độ chính xác Cảm biến: Độ chính xác của cảm biến MQ-3 có thể bị ảnh hưởng bởi các yếu tố môi trường (nhiệt độ, độ ẩm) và cần quy trình hiệu chuẩn định kỳ để đảm bảo tính pháp lý của kết quả đo.

- Phụ thuộc Kết nối: Khả năng truyền dữ liệu phụ thuộc vào kết nối Bluetooth và Wi-Fi. Nếu mất kết nối, việc cập nhật dữ liệu liên tục bị gián đoạn.

- Bảo mật Truyền tải: Cơ chế bảo mật hiện tại hoạt động nhưng chưa thật sự tối ưu; kết nối Bluetooth và HTTP/REST API cần được mã hóa đầy đủ bằng các giao thức như TLS/SSL để chống giả mạo lệnh điều khiển và đánh cắp thông tin.

5.3. Hướng phát triển tương lai

5.3.1. Về hệ thống

- Mở rộng Phạm vi: Hỗ trợ thêm các giao thức truyền thông khác như 3G/4G/LoRa để mở rộng phạm vi hoạt động của thiết bị, không còn phụ thuộc vào Wi-Fi cục bộ.

- Tích hợp GPS: Thêm module GPS để ghi lại tọa độ chính xác của vụ vi phạm, làm tăng tính pháp lý và khả năng truy vết dữ liệu.

- Phân tích dữ liệu: Phát triển các thuật toán phân tích dữ liệu nâng cao, thậm chí áp dụng AI để dự đoán xu hướng vi phạm theo thời gian hoặc khu vực.

- Cảnh báo đa kênh: Bổ sung chức năng cảnh báo qua SMS hoặc ứng dụng di động (Push Notification) thay vì chỉ hiển thị trên giao diện giám sát.

5.3.2. Về bảo mật

- Mã hóa kênh truyền: Mã hóa toàn bộ kênh truyền dữ liệu, bao gồm cả Bluetooth và API RESTful, bằng các giao thức bảo mật như TLS hoặc SSL để chống nghe lén và giả mạo lệnh điều khiển.

- Bảo vệ dữ liệu pháp lý: Nghiên cứu ứng dụng Blockchain để ghi lại và bảo vệ dữ liệu đo nồng độ cồn một cách minh bạch và không thể thay đổi, đảm bảo tính hợp pháp.

- Xác thực hai lớp (2FA): Bổ sung xác thực hai lớp khi đăng nhập vào hệ thống để tăng cường bảo mật tài khoản người dùng và quản trị viên

BẢNG PHÂN CÔNG CÔNG VIỆC

Họ tên	Nhiệm vụ
Chu Ngọc Thắng - B22DCCN807	Lập trình BackEnd Thiết kế database
Nguyễn Hoàng Vũ - B22DCCN915	Lập trình BackEnd Lập trình Arduino
Đặng Tiến Đạt - B22DCCN183	Lập trình Arduino Hỗ trợ các API liên quan
Lâm Trung Hiếu - B21DCCN354	Lập trình FrontEnd