

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN CUỐI KÌ  
MÔN HỌC: HỆ THỐNG NHÚNG  
MÃ HỌC PHẦN: ESYS431080**

**ĐỀ TÀI:  
CAR OVERSPEEDING DETECTION PROJECT**

**Giảng viên hướng dẫn: Thầy Đinh Công Đoan**

**Nhóm: 18**

**Danh sách sinh viên thực hiện:**

1. Xín Lợi Huy – 23110231
2. Bùi Nhật Thành – 23162091
3. Nguyễn Quốc Tĩnh – 23162101

***TP. Hồ Chí Minh, tháng 12 năm 2025***

# MỤC LỤC

## DANH MỤC HÌNH ẢNH

<b>PHẦN MỞ ĐẦU .....</b>	<b>1</b>
<b>1.1. Tóm tắt .....</b>	<b>1</b>
<b>1.2. Đặt vấn đề .....</b>	<b>1</b>
1.2.1. Tóm lược những nghiên cứu trong và ngoài nước liên quan đến đề tài .....	1
1.2.2. Tính cấp thiết cần nghiên cứu của đề tài .....	2
1.2.3. Một số tài liệu có liên quan .....	2
1.2.4. Lý do chọn đề tài .....	3
1.2.5. Mục tiêu đề tài .....	3
1.2.6. Đối tượng và phạm vi nghiên cứu .....	4
1.2.7. Phương pháp nghiên cứu .....	4
1.2.8. Nội dung đề tài .....	4
<b>PHẦN NỘI DUNG .....</b>	<b>5</b>
<b>CHƯƠNG 1. GIỚI THIỆU VI XỬ LÝ.....</b>	<b>5</b>
<b>1.1. Tổng quan về vi xử lý/vi điều khiển .....</b>	<b>5</b>
<b>1.2. Tổng quan về vi điều khiển ESP32.....</b>	<b>5</b>
1.2.1. Tổng quan về dòng ESP32 .....	5
1.2.2. Các đặc điểm nổi bật của ESP32 .....	6
1.2.3. Lý do lựa chọn ESP32 cho đề tài .....	6
<b>1.3. Khả năng ứng dụng của ESP32 trong hệ thống phát hiện vượt tốc độ.....</b>	<b>7</b>
<b>CHƯƠNG 2. CẤU TRÚC CHUNG CỦA KIT .....</b>	<b>8</b>
<b>2.1. Tổng quan về kit sử dụng trong đề tài.....</b>	<b>8</b>
<b>2.2. Các thành phần phần cứng của kit .....</b>	<b>8</b>
2.2.1. Vi điều khiển ESP32 / ESP32-CAM .....	8
2.2.2. Màn hình LCD giao tiếp I2C.....	10
2.2.3. Buzzer cảnh báo .....	11
2.2.4. MH Sensor.....	12
2.2.5. Module cảm biến ánh sáng & laser .....	12
2.2.6. Các công tắc, nút bấm .....	14
<b>2.3. Sơ đồ khối hệ thống .....</b>	<b>15</b>
<b>CHƯƠNG 3. NHỮNG KIẾN THỨC LIÊN QUAN .....</b>	<b>16</b>
<b>3.1. Tổng quan về bài toán phát hiện xe vượt quá tốc độ .....</b>	<b>16</b>
3.1.1. Trường hợp thiết bị lắp đặt ở ngoài đường .....	16
3.1.2. Trường hợp thiết bị được lắp đặt bên trong xe .....	16
<b>3.2. Nguyên lý đo và tính toán tốc độ.....</b>	<b>17</b>

3.2.1. Trường hợp bên ngoài đường .....	17
3.2.2. Trường hợp bên trong xe .....	17
<b>3.3. Nguyên lý hoạt động của cảm biến và thiết bị sử dụng.....</b>	<b>18</b>
3.3.1. Khối cảm biến quang (Laser Speed Trap) .....	18
3.3.2. Khối cảm biến Hall (Speedometer) .....	19
<b>3.4. Xử lý tín hiệu và thuật toán trong hệ thống nhúng .....</b>	<b>19</b>
3.4.1. Cơ chế ngắt (Interrupt Service Routine - ISR) .....	19
3.4.2. Quản lý đa nhiệm và Máy trạng thái (Finite State Machine) .....	20
<b>CHƯƠNG 4. ỨNG DỤNG.....</b>	<b>21</b>
<b>4.1. Sơ đồ khối và nguyên tắc hoạt động của hệ thống.....</b>	<b>21</b>
4.1.1. Sơ đồ khối của hệ thống .....	21
4.1.2. Nguyên tắc hoạt động của hệ thống .....	21
<b>4.2. Thiết kế và kết nối phần cứng ứng dụng.....</b>	<b>23</b>
<b>4.3. Xây dựng chương trình, nạp code.....</b>	<b>25</b>
<b>4.4. Triển khai giao diện kết quả .....</b>	<b>26</b>
<b>4.5. Kết quả thực hiện, khó khăn và hướng khắc phục.....</b>	<b>31</b>
4.5.1. Kết quả thực hiện.....	31
4.5.2. Các khó khăn trong quá trình thực hiện .....	32
4.5.3. Hướng khắc phục và phát triển tương lai .....	33
<b>PHẦN KẾT LUẬN .....</b>	<b>34</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>35</b>

## DANH MỤC HÌNH ẢNH

Hình 1. Vi điều khiển ESP32/ESP32-CAM .....	8
Hình 2. Màn hình LCD giao tiếp I2C.....	10
Hình 3. Còi Buzzer .....	11
Hình 4. MH Sensor .....	12
Hình 5. Module cảm biến ánh sáng.....	12
Hình 6. Laser .....	13
Hình 7. Nút bấm .....	14
Hình 8. Công tắc.....	14
Hình 9. Khối cảm biến .....	15
Hình 10. Khối xử lý trung tâm.....	15
Hình 11. Khối hiển thị và cảnh báo .....	15
Hình 12. Khối điều khiển .....	15
Hình 13. Sơ đồ kết nối của hệ thống.....	21
Hình 14. Kết nối các phần cứng với nhau.....	23
Hình 15. Kết nối các cảm biến, mạch với ESP32 .....	24
Hình 16. Các module được kết nối .....	24
Hình 17. Giao diện tổng quan của Dashboard .....	27
Hình 18. Giao diện danh sách xe .....	27
Hình 19. Giao diện thông tin xe vi phạm .....	28
Hình 20. Hình ảnh xe vi phạm .....	28
Hình 21. Biểu số xe được nhận diện .....	29
Hình 22. Biểu đồ thống kê vi phạm .....	29
Hình 23. Popup cảnh báo vi phạm .....	29
Hình 24. ChatBot AI hỗ trợ giải đáp .....	30
Hình 25. Xuất file pdf cho việc lưu trữ .....	30
Hình 26. Cơ sở dữ liệu được tổ chức .....	31
Hình 27. Mô hình hoàn thiện .....	31
Hình 28. Ảnh hưởng bởi điều kiện ánh sáng.....	32
Hình 29. Chất lượng hình ảnh thu được chưa cao .....	33

## **LỜI CẢM ƠN**

Với lòng biết ơn sâu sắc, chúng em xin gửi lời cảm ơn chân thành nhất đến thầy Đinh Công Doan, giảng viên Khoa Công nghệ Thông tin, đã tận tình hướng dẫn, truyền đạt kiến thức quý báu và tạo mọi điều kiện thuận lợi để chúng em hoàn thành đồ án môn học Hệ thống nhúng.

Trong suốt quá trình thực hiện đồ án, thầy đã luôn theo sát, chỉ bảo chi tiết, giúp chúng em vượt qua những khó khăn về mặt chuyên môn và định hướng nghiên cứu. Những ý kiến đóng góp kịp thời và sự khích lệ của thầy là nguồn động lực to lớn để đồ án của chúng em được hoàn thành đúng tiến độ và đạt được kết quả tốt nhất.

Chúng em xin kính chúc thầy dồi dào sức khỏe và tiếp tục gặt hái nhiều thành công trong sự nghiệp trồng người.

Nhóm xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày... tháng... năm 2025

Nhóm Sinh viên thực hiện

Xín Lợi Huy

Bùi Nhật Thành

Nguyễn Quốc Tĩnh

ĐIỂM SỐ

TIÊU CHÍ	NỘI DUNG	TRÌNH BÀY	TỔNG
ĐIỂM			

NHẬN XÉT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ký tên

## BẢNG PHÂN CÔNG NHIỆM VỤ

THỨ TỰ	HỌ TÊN	NHIỆM VỤ	KẾT QUẢ	KÝ TÊN
1	Xín Lợi Huy	<ul style="list-style-type: none"><li>- Xử lý hệ thống nhúng phần bên trong speedometer đo tốc độ với cảm biến MH sensor.</li><li>- Xử lý ảnh biển số xe.</li></ul>	Hoàn thành tốt	
2	Bùi Nhật Thành	<ul style="list-style-type: none"><li>- Xử lý hệ thống nhúng phần mạch cảnh báo tốc độ bên ngoài.</li><li>- Hỗ trợ code phần bên trong và hỗ trợ xử lý giao thức gửi qua http.</li></ul>	Hoàn thành tốt	
3	Nguyễn Quốc Tĩnh	<ul style="list-style-type: none"><li>- Xử lý code tín hiệu gửi và kết nối Wifi cho phần cứng.</li><li>- Xử lý dữ liệu thu được và làm giao diện trang Dashboard.</li></ul>	Hoàn thành tốt	

## PHẦN MỞ ĐẦU

### 1.1. Tóm tắt

Đề tài “Car Overspeeding Detection Project” của nhóm tập trung nghiên cứu và xây dựng hệ thống phát hiện xe vượt quá tốc độ cho phép trong hai trường hợp: bên trong xe và ngoài đường giao thông. Hệ thống được thiết kế dựa trên nền tảng vi xử lý/vi điều khiển, kết hợp với các cảm biến đo tốc độ và các mô-đun hỗ trợ nhằm thu thập, xử lý và đánh giá thông tin vận tốc của phương tiện.

Nội dung báo cáo trình bày cơ sở lý thuyết về các cấu trúc phần cứng được sử dụng trong đề tài, bao gồm vi xử lý, cảm biến, các mô-đun giao tiếp và mạch hỗ trợ. Bên cạnh đó, báo cáo mô tả chi tiết quy trình triển khai hệ thống, từ khâu thiết kế sơ đồ khối, xây dựng phần cứng, lập trình điều khiển cho đến tích hợp và kiểm thử thực tế.

Kết quả thực nghiệm cho thấy hệ thống có khả năng phát hiện và cảnh báo khi xe vượt quá tốc độ cài đặt, đáp ứng được yêu cầu đặt ra trong cả hai môi trường sử dụng. Thông qua đề tài, nhóm không chỉ củng cố kiến thức về vi xử lý và hệ thống nhúng mà còn rèn luyện kỹ năng thiết kế, triển khai và đánh giá một ứng dụng thực tiễn trong lĩnh vực an toàn giao thông.

### 1.2. Đặt vấn đề

#### 1.2.1. Tóm lược những nghiên cứu trong và ngoài nước liên quan đến đề tài

Trong nhiều năm trở lại đây, vấn đề phát hiện và kiểm soát tốc độ phương tiện đã được nghiên cứu rộng rãi trên thế giới nhằm giảm thiểu các vi phạm giao thông và tai nạn do chạy quá tốc độ. Một số nghiên cứu tiêu biểu trong lĩnh vực này tập trung vào việc phát triển các hệ thống phát hiện tốc độ xe hiệu quả và chính xác bằng nhiều phương pháp khác nhau.

Ở ngoài nước, các nghiên cứu đã phát triển nhiều hệ thống giám sát tốc độ dựa trên công nghệ hiện đại. Ví dụ, hệ thống *Automatic Vehicle Over Speed Detection Alert and Controlling System*<sup>1</sup> sử dụng camera và cảm biến để phát hiện xe vượt quá tốc độ và ghi lại hình ảnh phương tiện vi phạm nhằm hỗ trợ quản lý giao thông trên đường cao tốc. Khi tốc độ vượt giới hạn cài đặt, thiết bị sẽ chụp hình xe và truyền dữ liệu đến máy tính để xử lý và lưu trữ thông tin vi phạm.

Một hướng tiếp cận khác là tích hợp radar đo tốc độ với camera, giúp tăng tốc độ phản hồi và cải thiện độ chính xác trong việc chụp ảnh xe khi phát hiện vượt tốc độ, đồng thời sử dụng xử lý ảnh và nhận dạng biển số để xác định phương tiện vi phạm.

---

<sup>1</sup> Kanavi, P., Chaithra, K. B., Chaitra, K. T., Bhoomika, M. G., & Lakshmi, C. T. (2022). *Automatic Vehicle Over Speed Detection Alert and Controlling System on Highway*. Published online 30-08-2022. Available at: <https://www.ijert.org/automatic-vehicle-over-speed-detection-alert-and-controlling-system-on-highway> (Accessed: 03 December 2025)



Các giải pháp dựa trên công nghệ IoT cũng được đề xuất như hệ thống *Smart Vehicle Overspeeding Detector*<sup>2</sup> sử dụng GPS và công nghệ IoT để theo dõi tốc độ xe trong thời gian thực và cảnh báo khi vượt quá giới hạn an toàn, giúp giảm nguy cơ tai nạn tại các khu vực đô thị.

Ở trong nước, mặc dù chưa có nhiều nghiên cứu công bố chính thức như ở ngoài nước, nhưng các ứng dụng thực tế như các hệ thống camera giám sát giao thông và thiết bị đo tốc độ cố định đã được triển khai tại nhiều tuyến đường lớn để hỗ trợ lực lượng chức năng phát hiện và xử lý vi phạm tốc độ. Ví dụ, các camera giám sát tại nhiều quốc gia đã chứng minh hiệu quả trong việc giảm tai nạn và các vụ vi phạm về tốc độ khi lắp đặt hệ thống phát hiện tự động.

Nhìn chung, các nghiên cứu và ứng dụng hiện tại đều nhấn mạnh tầm quan trọng của việc sử dụng cảm biến, camera, IoT và thuật toán xử lý tín hiệu/ảnh nhằm xây dựng hệ thống phát hiện tốc độ hiệu quả, phục vụ mục tiêu an toàn giao thông và giám sát vi phạm một cách tự động.

### 1.2.2. Tính cấp thiết cần nghiên cứu của đề tài

Chạy quá tốc độ là một trong những nguyên nhân chính gây tai nạn giao thông, đặc biệt trong điều kiện mật độ phương tiện ngày càng tăng. Các biện pháp kiểm soát tốc độ hiện nay còn hạn chế, chi phí cao và chưa bao phủ đầy đủ cả giám sát ngoài đường lẫn cảnh báo trực tiếp bên trong xe.

Do đó, việc nghiên cứu và xây dựng hệ thống phát hiện xe vượt quá tốc độ dựa trên vi xử lý, có khả năng ứng dụng trong nhiều môi trường, là cần thiết nhằm nâng cao an toàn giao thông, hỗ trợ công tác giám sát và nâng cao ý thức chấp hành luật của người điều khiển phương tiện.

### 1.2.3. Một số tài liệu có liên quan

Trong quá trình nghiên cứu và thực hiện đề tài, nhóm đã tham khảo một số tài liệu khoa học, báo cáo kỹ thuật và bài nghiên cứu có nội dung liên quan đến việc phát hiện và đo tốc độ xe, hệ thống giám sát tốc độ xe tự động cũng như các ứng dụng vi xử lý/embedded systems trong giám sát giao thông:

1. **Automatic Vehicle Over Speed Detection and Alert System on Highway** – Báo cáo nghiên cứu hệ thống phát hiện vượt tốc độ thông minh sử dụng cảm biến và IoT để giám sát tốc độ phương tiện trên đường cao tốc. Hệ thống tự động phát hiện và gửi

---

<sup>2</sup> Sreekanth, N., Spandana, I., Shravya, J., & Sravani, K. (2024). *Vehicle Over Speed Indication Based IoT*. *International Journal of Advanced Research in Science and Technology (IJARST)*, Vol. 14, Issue 10, pp. 170–174. Available at: <https://www.ijarst.in/public/uploads/paper/582441729063704.pdf> (Accessed: 03 December 2025)

cảnh báo khi phương tiện vượt quá giới hạn tốc độ. Tài liệu tham khảo: <https://technologyjournal.net/wp-content/uploads/8-JOT1482.pdf>

2. **Vehicle Detection and Speed Tracking – IJERT** – Nghiên cứu mô tả các phương pháp phát hiện xe và theo dõi vận tốc trong video, sử dụng kỹ thuật xử lý ảnh như *Background Subtraction* và *Feature Extraction* để ước tính tốc độ phương tiện. Tài liệu tham khảo: <https://www.ijert.org/research/vehicle-detection-and-speed-tracking-IJERTV10IS050428.pdf>
3. **Development & Implementation of Smart Vehicle Over Speeding Detector using IoT** – Nghiên cứu ứng dụng IoT và GPS để xây dựng hệ thống phát hiện vượt tốc độ, thử nghiệm thực tế trong xe sử dụng phần cứng Raspberry Pi và GPS. Tài liệu tham khảo: [https://www.academia.edu/104606204/Development\\_and\\_Implementation\\_of\\_Smart\\_Vehicle\\_Over\\_Speeding\\_Detector\\_using\\_IoT](https://www.academia.edu/104606204/Development_and_Implementation_of_Smart_Vehicle_Over_Speeding_Detector_using_IoT)

#### 1.2.4. Lý do chọn đề tài

Trong bối cảnh giao thông ngày càng phức tạp, tình trạng phương tiện chạy quá tốc độ cho phép vẫn diễn ra phổ biến và là nguyên nhân chính dẫn đến nhiều vụ tai nạn nghiêm trọng. Việc kiểm soát tốc độ hiện nay chủ yếu dựa vào các thiết bị đo chuyên dụng ngoài đường hoặc ý thức tự giác của người điều khiển phương tiện, tuy nhiên hiệu quả vẫn còn hạn chế. Do đó, nhu cầu về một hệ thống phát hiện và cảnh báo vượt tốc độ mang tính chủ động, linh hoạt và dễ triển khai là rất cần thiết.

Bên cạnh tính thực tiễn, đề tài còn phù hợp với định hướng học tập và nghiên cứu của nhóm trong lĩnh vực vi xử lý và hệ thống nhúng. Thông qua đề tài, nhóm có cơ hội vận dụng kiến thức lý thuyết đã học vào việc thiết kế phần cứng, lập trình điều khiển và tích hợp các mô-đun cảm biến. Việc nghiên cứu hệ thống phát hiện vượt tốc độ trong cả môi trường bên trong xe và ngoài đường giao thông giúp mở rộng khả năng ứng dụng, đồng thời nâng cao giá trị học thuật và thực tiễn của đề tài.

#### 1.2.5. Mục tiêu đề tài

Mục tiêu chính của đề tài là thiết kế và xây dựng một hệ thống phát hiện xe vượt quá tốc độ cho phép dựa trên nền tảng vi xử lý/vi điều khiển. Hệ thống có khả năng thu thập dữ liệu tốc độ từ các cảm biến, xử lý và so sánh với ngưỡng tốc độ được thiết lập trước, từ đó đưa ra cảnh báo khi phát hiện vi phạm. Hệ thống hướng tới sự ổn định, độ chính xác phù hợp và khả năng hoạt động trong các điều kiện khác nhau.

Ngoài mục tiêu chính, đề tài còn nhằm giúp nhóm củng cố và nâng cao kiến thức chuyên môn về cấu trúc phần cứng, nguyên lý hoạt động của cảm biến, phương pháp đo và xử lý tín

hiệu tốc độ. Đồng thời, đề tài góp phần rèn luyện kỹ năng phân tích, thiết kế hệ thống nhúng hoàn chỉnh từ lý thuyết đến thực nghiệm, tạo nền tảng cho việc nghiên cứu và phát triển các ứng dụng giao thông thông minh trong tương lai.

#### **1.2.6. Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu của đề tài là hệ thống phát hiện xe vượt quá tốc độ cho phép dựa trên vi xử lý/vi điều khiển và các cảm biến đo tốc độ. Cụ thể, đề tài tập trung nghiên cứu nguyên lý đo vận tốc của phương tiện, quá trình thu thập và xử lý dữ liệu tốc độ, cũng như cơ chế so sánh và đưa ra cảnh báo khi xe vượt quá ngưỡng tốc độ cài đặt.

Phạm vi nghiên cứu của đề tài giới hạn ở việc xây dựng mô hình thử nghiệm (prototype) với quy mô nhỏ, có khả năng ứng dụng trong hai trường hợp: bên trong xe và ngoài đường giao thông. Hệ thống được triển khai với các linh kiện phổ biến, chi phí thấp và thuật toán xử lý đơn giản. Đề tài không đi sâu vào các hệ thống thương mại quy mô lớn hay các thuật toán xử lý ảnh, trí tuệ nhân tạo phức tạp.

#### **1.2.7. Phương pháp nghiên cứu**

Đề tài được thực hiện dựa trên phương pháp nghiên cứu lý thuyết kết hợp với thực nghiệm. Trước hết, nhóm tiến hành khảo sát và phân tích các tài liệu liên quan đến vi xử lý, cảm biến đo tốc độ và các hệ thống phát hiện vượt tốc độ đã được nghiên cứu trước đó nhằm xây dựng cơ sở lý thuyết cho đề tài.

Tiếp theo, nhóm áp dụng phương pháp thiết kế – triển khai – kiểm thử, bao gồm thiết kế sơ đồ khối hệ thống, xây dựng mạch phần cứng, lập trình điều khiển và tích hợp các mô-đun. Cuối cùng, hệ thống được kiểm tra, đánh giá kết quả hoạt động, phân tích các hạn chế và đề xuất hướng cải tiến nhằm hoàn thiện mô hình.

#### **1.2.8. Nội dung đề tài**

Nội dung đề tài tập trung vào việc nghiên cứu và xây dựng hệ thống phát hiện xe vượt quá tốc độ cho phép dựa trên vi xử lý/vi điều khiển. Trước hết, đề tài trình bày cơ sở lý thuyết về vi xử lý được sử dụng, cấu trúc phần cứng của hệ thống và các linh kiện, mô-đun liên quan như cảm biến đo tốc độ, mạch giao tiếp và thiết bị cảnh báo.

Tiếp theo, đề tài đi sâu vào quá trình thiết kế và triển khai hệ thống, bao gồm xây dựng sơ đồ khối, sơ đồ kết nối phần cứng, lập trình điều khiển và tích hợp các chức năng đo tốc độ, so sánh ngưỡng và cảnh báo vượt tốc độ trong cả hai trường hợp: bên trong xe và ngoài đường giao thông. Cuối cùng, đề tài đánh giá kết quả thực nghiệm, phân tích ưu nhược điểm của hệ thống và đề xuất hướng phát triển trong tương lai.

## PHẦN NỘI DUNG

### CHƯƠNG 1. GIỚI THIỆU VI XỬ LÝ

#### 1.1. Tổng quan về vi xử lý/vi điều khiển

Vi xử lý (Microprocessor) là một thiết bị số có thể lập trình, có khả năng nhận dữ liệu đầu vào, thực hiện các phép toán số học và logic, sau đó tạo ra kết quả đầu ra mong muốn. Nói một cách đơn giản, vi xử lý là một chip kỹ thuật số có thể lấy lệnh từ bộ nhớ, giải mã và thực thi các lệnh đó, đồng thời trả về kết quả xử lý. Vi xử lý là thành phần cốt lõi của kiến trúc máy tính, giữ vai trò trung tâm trong việc điều khiển và xử lý mọi hoạt động của hệ thống.<sup>3</sup>

Vi xử lý thực hiện chương trình thông qua các lệnh ở dạng mã máy. Trong quá trình thực thi lệnh, vi xử lý đảm nhiệm ba chức năng cơ bản:

- Thực hiện các phép toán số học và logic thông qua ALU
- Truyền và xử lý dữ liệu giữa các thanh ghi và bộ nhớ
- Điều khiển luồng chương trình thông qua Program Counter (PC), cho phép nhảy lệnh và ra quyết định

Các kiến trúc vi xử lý phổ biến

- **CISC (Complex Instruction Set Computer):** Tập lệnh phức tạp, mỗi lệnh có thể thực hiện nhiều thao tác; sử dụng ít thanh ghi. Ví dụ: Intel 386, 486, Pentium
- **RISC (Reduced Instruction Set Computer):** Tập lệnh đơn giản, thực thi nhanh, tối ưu pipeline; sử dụng nhiều thanh ghi. Ví dụ: IBM RS6000, DEC Alpha
- **EPIC (Explicitly Parallel Instruction Computing):** Cho phép thực thi song song các lệnh nhờ trình biên dịch. Ví dụ: IA-64

#### 1.2. Tổng quan về vi điều khiển ESP32

##### 1.2.1. Tổng quan về dòng ESP32

ESP32 là dòng vi điều khiển SoC (System on Chip) do Espressif Systems phát triển, được xem là thế hệ kế nhiệm và mở rộng của ESP8266. Mặc dù ESP32 có giá thành cao hơn ESP8266, nhưng điều này là hoàn toàn hợp lý do ESP32 được tích hợp nhiều tính năng hơn, hiệu năng cao hơn và khả năng mở rộng tốt hơn, đặc biệt phù hợp cho các ứng dụng IoT và hệ thống nhúng hiện đại.

---

<sup>3</sup> GeeksforGeeks. (2025). *Introduction of Microprocessor*. GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/electronics-engineering/introduction-of-microprocessor/> (Accessed: 08 December 2025).

ESP32 tích hợp đầy đủ bộ xử lý, bộ nhớ, các khối giao tiếp không dây và ngoại vi trên cùng một chip, giúp giảm số lượng linh kiện bên ngoài và tăng độ tin cậy của hệ thống.

### **1.2.2. Các đặc điểm nổi bật của ESP32**

So với ESP8266, ESP32 có nhiều cải tiến đáng kể về phần cứng và chức năng, nổi bật gồm:

Tốc độ xử lý cao: ESP32 sử dụng vi xử lý Xtensa dual-core 32-bit LX6, hiệu năng lên đến 600 DMIPS, tần số hoạt động tối đa 240 MHz, cho khả năng xử lý nhanh và ổn định, vượt trội so với ESP8266 chỉ có lõi đơn.

Kết nối không dây mạnh mẽ: ESP32 hỗ trợ WiFi chuẩn 802.11 b/g/n (HT40) và Bluetooth 4.2 cùng Bluetooth Low Energy (BLE), trong khi ESP8266 chỉ hỗ trợ WiFi. Điều này giúp ESP32 dễ dàng giao tiếp với nhiều thiết bị và nền tảng khác nhau.

Bộ nhớ lớn và linh hoạt: ESP32 được tích hợp ROM, SRAM, RTC SRAM và hỗ trợ flash bộ nhớ trong hoặc mở rộng flash ngoài lên đến 16 MB, cho phép phát triển các ứng dụng phức tạp mà không bị giới hạn bộ nhớ.

Hệ thống GPIO và ngoại vi phong phú: ESP32 có lên đến 34 chân GPIO, hỗ trợ nhiều giao thức như SPI, I2C, UART, I2S, CAN, cùng với ADC độ phân giải 12-bit, DAC, PWM lên đến 16 kênh, cảm biến chạm điện dung, cảm biến Hall và RTC.

Tiêu thụ năng lượng thấp: ESP32 tích hợp bộ đồng xử lý ULP (Ultra Low Power) cho phép hệ thống hoạt động trong chế độ ngủ sâu mà vẫn duy trì một số chức năng, rất phù hợp cho các hệ thống dùng pin.

Bảo mật phần cứng: ESP32 hỗ trợ các bộ tăng tốc mã hóa phần cứng như AES, SHA, RSA, giúp nâng cao mức độ an toàn cho dữ liệu và chương trình.

### **1.2.3. Lý do lựa chọn ESP32 cho đề tài**

ESP32 được lựa chọn cho đề tài vì các lý do chính sau:

- Hiệu năng xử lý cao, đáp ứng tốt yêu cầu tính toán và xử lý dữ liệu thời gian thực
- Tích hợp đồng thời WiFi và Bluetooth, thuận lợi cho việc truyền dữ liệu và mở rộng hệ thống
- Nhiều GPIO và ngoại vi, dễ dàng kết nối với cảm biến và thiết bị chấp hành

- Hỗ trợ tiết kiệm năng lượng, phù hợp cho các hệ thống hoạt động liên tục hoặc dùng pin
- Bảo mật tốt và khả năng mở rộng cao cho các hướng phát triển trong tương lai

Nhờ những ưu điểm trên, ESP32 là một vi điều khiển mạnh, linh hoạt và đáng tin cậy, rất phù hợp để sử dụng trong các đề tài hệ thống nhúng và IoT hiện đại.

### **1.3. Khả năng ứng dụng của ESP32 trong hệ thống phát hiện vượt tốc độ**

ESP32 có hiệu năng xử lý cao nhờ bộ vi xử lý 32-bit tốc độ lớn, cho phép thu thập và xử lý dữ liệu từ các cảm biến một cách nhanh chóng và chính xác. Với bộ chuyển đổi ADC 12-bit, ESP32 có thể đọc tín hiệu tương tự từ các cảm biến đo khoảng cách, cảm biến hồng ngoại hoặc các cảm biến tốc độ khác, từ đó tính toán vận tốc của đối tượng theo thời gian thực.

Khả năng xử lý song song và thời gian đáp ứng nhanh của ESP32 giúp hệ thống phát hiện chính xác các trường hợp vượt ngưỡng tốc độ đã cài đặt, đảm bảo độ tin cậy của kết quả đo.

ESP32 hỗ trợ nhiều chân GPIO và ngoại vi, cho phép điều khiển linh hoạt các thiết bị trong hệ thống phát hiện vượt tốc độ như:

- Buzzer: phát tín hiệu cảnh báo khi phát hiện phương tiện vượt tốc độ cho phép
- Màn hình LCD/OLED: hiển thị tốc độ đo được, ngưỡng tốc độ và trạng thái hệ thống
- Nút nhấn và công tắc: cho phép người dùng thiết lập ngưỡng tốc độ, khởi động hoặc dừng hệ thống

Việc hỗ trợ PWM, I2C, SPI và UART giúp ESP32 giao tiếp và điều khiển các thiết bị ngoại vi một cách ổn định và hiệu quả.

Nhờ tích hợp WiFi và Bluetooth, ESP32 cho phép mở rộng hệ thống phát hiện vượt tốc độ theo nhiều hướng khác nhau, như:

- Truyền dữ liệu tốc độ và cảnh báo về máy chủ hoặc ứng dụng di động
- Lưu trữ và thống kê dữ liệu vi phạm để phục vụ phân tích
- Kết hợp thêm các cảm biến khác nhằm nâng cao độ chính xác
- Phát triển thành hệ thống giám sát giao thông thông minh trong mô hình IoT

Với khả năng mở rộng linh hoạt và hiệu năng mạnh mẽ, ESP32 đáp ứng tốt các yêu cầu phát triển và nâng cấp hệ thống trong tương lai.

## CHƯƠNG 2. CẤU TRÚC CHUNG CỦA KIT

### 2.1. Tổng quan về kit sử dụng trong đề tài

Kit được sử dụng trong đề tài nhằm mục đích xây dựng một mô hình hệ thống phát hiện xe vượt quá tốc độ cho phép dựa trên vi điều khiển. Kit đóng vai trò là nền tảng phần cứng để tích hợp các cảm biến, mô-đun và thiết bị ngoại vi, phục vụ cho việc thu thập dữ liệu, xử lý tín hiệu và đưa ra cảnh báo khi phát hiện xe vượt tốc độ.

Bên cạnh đó, kit còn giúp nhóm dễ dàng thử nghiệm, hiệu chỉnh và đánh giá hoạt động của hệ thống trong các điều kiện khác nhau, phù hợp cho việc học tập, nghiên cứu và phát triển mô hình thử nghiệm.

Kit có chức năng chính là điều khiển và xử lý toàn bộ hoạt động của hệ thống. Cụ thể, kit đảm nhiệm việc nhận tín hiệu từ các cảm biến, tính toán vận tốc của phương tiện, so sánh với ngưỡng tốc độ đã cài đặt và kích hoạt các thiết bị cảnh báo như buzzer hoặc hiển thị thông tin lên màn hình LCD. Ngoài ra, kit còn hỗ trợ người dùng điều chỉnh chế độ hoạt động thông qua nút bấm và công tắc.

### 2.2. Các thành phần phần cứng của kit

#### 2.2.1. Vi điều khiển ESP32 / ESP32-CAM



*Hình 1. Vi điều khiển ESP32/ESP32-CAM*

ESP32-CAM là một module vi điều khiển tích hợp camera có kích thước nhỏ gọn, sử dụng chip ESP32-S, được thiết kế cho các ứng dụng IoT, giám sát và xử lý hình ảnh. Module

này tích hợp sẵn camera OV2640, khe cắm thẻ nhớ microSD (TF card) và các chân GPIO để kết nối với thiết bị ngoại vi.<sup>4</sup>

Với chi phí thấp nhưng hiệu năng cao, ESP32-CAM rất phù hợp cho các hệ thống cần ghi nhận hình ảnh kết hợp xử lý và truyền dữ liệu không dây.

ESP32-CAM sử dụng CPU 32-bit công suất thấp, có thể hoạt động như bộ xử lý ứng dụng chính của hệ thống. Các đặc điểm nổi bật gồm:

- Tốc độ xung nhịp lên đến 160 MHz
- Hiệu năng tính toán đạt khoảng 600 DMIPS
- 520 KB SRAM tích hợp
- Hỗ trợ PSRAM ngoài 4 MB, rất quan trọng cho xử lý ảnh và video
- Hỗ trợ nhiều chế độ tiết kiệm năng lượng (sleep modes)

Nhờ đó, ESP32-CAM có khả năng xử lý dữ liệu hình ảnh và truyền thông không dây một cách hiệu quả.

ESP32-CAM hỗ trợ:

- Camera OV2640 (chính) và OV7670
- Đèn flash LED tích hợp
- Truyền hình ảnh qua WiFi
- Khe cắm thẻ nhớ microSD, dùng để:
  - Lưu ảnh chụp
  - Lưu video
  - Lưu file dữ liệu phục vụ hệ thống

Đây là ưu điểm lớn giúp ESP32-CAM có thể ghi nhận bằng chứng hình ảnh khi xảy ra sự kiện (ví dụ vượt tốc độ).

ESP32-CAM là một module vi điều khiển mạnh, nhỏ gọn và chi phí thấp, tích hợp đầy đủ xử lý, camera, lưu trữ và kết nối không dây. Những đặc điểm này giúp ESP32-CAM trở thành lựa chọn phù hợp cho các hệ thống nhúng hiện đại, đặc biệt là các ứng dụng IoT và giám sát thông minh.

---

<sup>4</sup> Random Nerd Tutorials. (2025). *ESP32-CAM Video Streaming and Face Recognition with Arduino IDE*. Available at: <https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/> (Accessed: 08 December 2025).



### 2.2.2. Màn hình LCD giao tiếp I2C



*Hình 2. Màn hình LCD giao tiếp I2C*

Thông số kỹ thuật:

- + LCD 16x2 có 16 chân trong đó 8 chân dữ liệu (D0 - D7) và 3 chân điều khiển (RS, RW, EN).
- + 5 chân còn lại dùng để cấp nguồn và đèn nền cho LCD 16x2.
- + Các chân điều khiển giúp ta dễ dàng cấu hình LCD ở chế độ lệnh hoặc chế độ dữ liệu.
- + Chúng còn giúp ta cấu hình ở chế độ đọc hoặc ghi.

LCD 16x2 có thể sử dụng ở chế độ 4 bit hoặc 8 bit tùy theo ứng dụng ta đang làm.

Màn hình LCD giao tiếp I2C là một module hiển thị ký tự 16×2 (16 ký tự trên mỗi dòng, 2 dòng) được trang bị giao tiếp I2C, giúp truyền dữ liệu giữa vi điều khiển và màn hình chỉ qua hai dây dữ liệu SDA và SCL, thay vì cần nhiều chân như giao tiếp song song truyền thống. Việc chuyển từ giao tiếp song song sang I2C giúp giảm đáng kể số chân I/O cần sử dụng, đơn giản hóa mạch điện và dễ dàng lập trình hơn.<sup>5</sup>

Việc sử dụng chuẩn giao tiếp I2C (Inter-Integrated Circuit) mang lại nhiều ưu điểm so với giao tiếp song song truyền thống. Màn hình LCD I2C chỉ cần hai đường tín hiệu chính là SDA và SCL, ngoài ra là nguồn cấp và chân GND, giúp giảm số lượng chân kết nối với vi điều khiển ESP32. Điều này không chỉ đơn giản hóa sơ đồ mạch, mà còn giúp tiết kiệm chân GPIO cho các cảm biến và thiết bị ngoại vi khác.

<sup>5</sup> Handson Technology. (n.d.). *I2C Serial Interface 1602 LCD Module User Guide*. Available at: [https://www.handsontec.com/dataspecs/module/I2C\\_1602\\_LCD.pdf](https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf) (Accessed: 08 December 2025).

Bên cạnh đó, LCD I2C có khả năng hoạt động ổn định, tiêu thụ điện năng thấp và dễ dàng lập trình thông qua các thư viện hỗ trợ sẵn, do đó rất phù hợp để sử dụng trong các hệ thống nhúng và IoT.

### 2.2.3. Buzzer cảnh báo



*Hình 3. Còi Buzzer*

Thông số kỹ thuật:

- + Điện áp hoạt động: 3 ~ 24VDC
- + Dòng điện tiêu thụ: 30mA 12V, 40mA 24V.
- + Biên độ âm thanh: lên đến 100dB.
- + Kích thước: 27 x 15mm.
- + Khối lượng: 10g.

Buzzer là thiết bị phát âm thanh được sử dụng trong hệ thống nhằm cảnh báo khi phát hiện phương tiện vượt quá tốc độ cho phép. Khi vi điều khiển ESP32 xác định vận tốc đo được lớn hơn ngưỡng đã cài đặt, buzzer sẽ được kích hoạt để phát ra tín hiệu âm thanh cảnh báo.

Thiết bị này giúp cảnh báo trực tiếp, nhanh chóng và dễ nhận biết, hỗ trợ người sử dụng hoặc người giám sát kịp thời nắm bắt trạng thái vi phạm tốc độ. Buzzer có cấu tạo đơn giản, dễ điều khiển thông qua chân GPIO hoặc tín hiệu PWM từ ESP32, tiêu thụ điện năng thấp và hoạt động ổn định, phù hợp cho các hệ thống nhúng và ứng dụng cảnh báo thời gian thực.

#### 2.2.4. MH Sensor

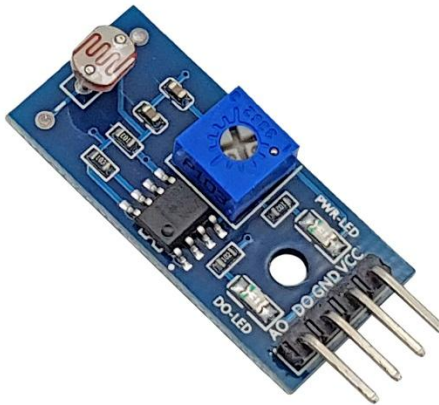


*Hình 4. MH Sensor*

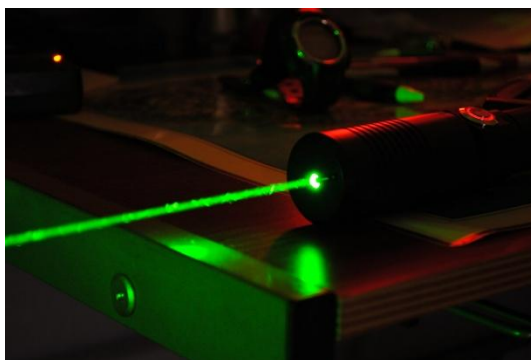
MH Sensor là cảm biến được sử dụng trong hệ thống nhằm hỗ trợ phát hiện chuyển động của phương tiện hoặc các thay đổi vật lý liên quan trong quá trình xe di chuyển qua khu vực đo. Cảm biến có khả năng tạo ra tín hiệu đầu ra khi phát hiện sự thay đổi do tác động của phương tiện, từ đó cung cấp thông tin cần thiết cho hệ thống.

Tín hiệu thu được từ MH Sensor được đưa về vi điều khiển ESP32 thông qua các chân GPIO để xử lý. Dữ liệu này được kết hợp với thông tin từ các cảm biến khác trong hệ thống nhằm phục vụ cho việc xác định thời điểm xe đi qua, tính toán vận tốc và phát hiện xe vượt quá tốc độ cho phép, góp phần nâng cao độ chính xác và độ tin cậy của hệ thống.

#### 2.2.5. Module cảm biến ánh sáng & laser



*Hình 5. Module cảm biến ánh sáng*



*Hình 6. Laser*

Thông số kỹ thuật Module ánh sáng:

- + Điện áp hoạt động 3.3 – 5 V
- + Kết nối 4 chân với 2 chân cấp nguồn (VCC và GND) và 2 chân tín hiệu ngõ ra (AO và DO).
- + Hỗ trợ cả 2 dạng tín hiệu ra Analog và TTL. Ngõ ra Analog 0 – 5V tỷ lệ thuận với cường độ ánh sáng, ngõ TTL tích cực mức thấp.
- + Độ nhạy cao với ánh sáng được tùy chỉnh bằng biến trở .
- + Kích thước 32 x 14mm

Hai cảm biến ánh sáng được sử dụng để nhận tín hiệu từ các module laser trong hệ thống. Ở trạng thái bình thường, cảm biến liên tục nhận được ánh sáng từ tia laser chiếu tới. Khi phương tiện di chuyển và cắt ngang tia laser, cường độ ánh sáng tại cảm biến thay đổi đột ngột, từ đó tạo ra tín hiệu đầu ra tương ứng.

Tín hiệu từ các cảm biến ánh sáng được đưa về vi điều khiển ESP32 để ghi nhận thời điểm phương tiện đi qua từng vị trí. Dựa trên thời gian phát hiện tại hai cảm biến, hệ thống xác định khoảng thời gian xe di chuyển giữa hai điểm đo, làm cơ sở cho việc tính toán vận tốc của phương tiện.

Hai module laser có nhiệm vụ tạo ra hai chùm tia laser song song tại hai vị trí xác định trên đường đo. Các tia laser này được bố trí cố định với khoảng cách xác định trước, đảm bảo tính chính xác trong quá trình đo vận tốc.

Khi phương tiện di chuyển qua khu vực đo, xe sẽ lần lượt cắt ngang từng tia laser. Khoảng thời gian giữa hai lần cắt tia được ESP32 ghi nhận thông qua tín hiệu từ các cảm biến ánh sáng. Dựa vào khoảng cách giữa hai module laser và thời gian xe di chuyển, hệ thống tiến hành tính toán tốc độ của phương tiện, từ đó xác định xe có vượt quá tốc độ cho phép hay không.

### 2.2.6. Các công tắc, nút bấm



*Hình 7. Nút bấm*



*Hình 8. Công tắc*

Nút bấm 3 chân được sử dụng trong hệ thống nhằm thay đổi logic hoạt động của chương trình. Cụ thể, nút bấm dùng để kích hoạt hoặc chuyển đổi các trạng thái xử lý như bắt đầu đo, dừng đo, đặt lại quá trình tính toán hoặc xác nhận thao tác của người dùng. Nút hoạt động theo nguyên lý đóng/ngắt, tạo ra sự thay đổi mức logic tại chân tín hiệu.

Tín hiệu từ nút bấm được đưa vào vi điều khiển ESP32 thông qua chân GPIO. Khi nút được nhấn, ESP32 sẽ nhận sự thay đổi mức logic và xử lý theo thuật toán đã được lập trình, giúp hệ thống hoạt động linh hoạt và phù hợp với từng yêu cầu sử dụng.

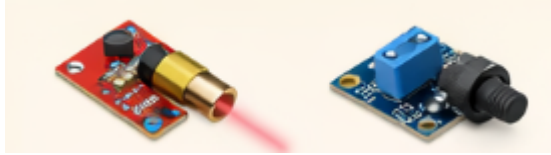
Công tắc gạt 3 chân được sử dụng để lựa chọn chế độ (mode) hoạt động của hệ thống. Thông qua công tắc, người dùng có thể chuyển đổi giữa chế độ đo tốc độ bên trong xe và chế độ đo tốc độ ngoài đường giao thông.

Công tắc được kết nối với vi điều khiển ESP32 thông qua các chân GPIO. Dựa trên trạng thái của công tắc, ESP32 xác định mode hoạt động tương ứng và điều chỉnh thuật toán xử lý, ngưỡng tốc độ cũng như cách thức cảnh báo cho phù hợp. Việc sử dụng công tắc gạt giúp thao tác đơn giản, trực quan và đảm bảo hệ thống hoạt động ổn định trong các môi trường khác nhau.

### 2.3. Sơ đồ khối hệ thống

Hệ thống được chia thành các khối chức năng chính gồm:

- Khối cảm biến: bao gồm module laser, cảm biến ánh sáng và MH Sensor, có nhiệm vụ thu thập dữ liệu.



*Hình 9. Khối cảm biến*

- Khối xử lý trung tâm: vi điều khiển ESP32/ESP32-CAM, thực hiện xử lý tín hiệu và tính toán tốc độ.



*Hình 10. Khối xử lý trung tâm*

- Khối hiển thị và cảnh báo: LCD I2C và buzzer, dùng để hiển thị thông tin và cảnh báo vượt tốc độ.



*Hình 11. Khối hiển thị và cảnh báo*

- Khối điều khiển: nút bấm và công tắc, cho phép người dùng tương tác với hệ thống.



*Hình 12. Khối điều khiển*

- Khối nguồn: cung cấp năng lượng cho toàn bộ hệ thống.

## **CHƯƠNG 3. NHỮNG KIẾN THỨC LIÊN QUAN**

### **3.1. Tổng quan về bài toán phát hiện xe vượt quá tốc độ**

#### **3.1.1. Trường hợp thiết bị lắp đặt ở ngoài đường**

Trong bối cảnh giao thông đô thị ngày càng phức tạp, tình trạng phương tiện di chuyển vượt quá tốc độ cho phép là một trong những nguyên nhân chính dẫn đến tai nạn giao thông, đặc biệt tại các khu vực đông dân cư, khu dân sinh, trường học và các tuyến đường có mật độ phương tiện cao. Do đó, nhu cầu xây dựng các hệ thống hỗ trợ lực lượng chức năng trong việc giám sát và kiểm soát tốc độ phương tiện là hết sức cần thiết.

Bài toán phát hiện phương tiện vượt quá tốc độ ở bên ngoài được đề xuất nhằm phục vụ công tác quản lý và xử lý vi phạm giao thông một cách tự động, chính xác và liên tục. Hệ thống được thiết kế để hoạt động độc lập tại hiện trường, cho phép phát hiện kịp thời các trường hợp vi phạm tốc độ mà không cần sự can thiệp thường xuyên của con người. Khi phát hiện phương tiện có dấu hiệu vượt quá tốc độ cho phép, hệ thống sẽ ghi nhận thông tin liên quan, bao gồm hình ảnh phương tiện và biển số, từ đó hỗ trợ cảnh sát giao thông trong công tác giám sát, xử phạt và thống kê.

Ngoài chức năng ghi nhận vi phạm, hệ thống còn cung cấp khả năng hiển thị tốc độ phương tiện theo thời gian thực thông qua màn hình LCD, giúp người vận hành hoặc lực lượng chức năng dễ dàng theo dõi tình trạng giao thông tại khu vực lắp đặt. Với các đặc điểm trên, bài toán phát hiện phương tiện vượt quá tốc độ đặc biệt phù hợp để triển khai tại các khu vực có nguy cơ tai nạn cao, góp phần nâng cao ý thức chấp hành luật giao thông và đảm bảo an toàn cho cộng đồng.

#### **3.1.2. Trường hợp thiết bị được lắp đặt bên trong xe**

Bên cạnh hệ thống giám sát tốc độ từ bên ngoài phục vụ công tác quản lý giao thông, bài toán giám sát tốc độ bên trong phương tiện được xây dựng nhằm hỗ trợ trực tiếp cho người điều khiển trong quá trình tham gia giao thông. Hệ thống này hướng đến việc cung cấp thông tin tốc độ một cách liên tục và trực quan, giúp người lái xe chủ động nhận biết khi phương tiện có nguy cơ vượt quá tốc độ cho phép, từ đó kịp thời điều chỉnh hành vi điều khiển cho phù hợp.

Bài toán bên trong phương tiện tập trung vào việc theo dõi trạng thái chuyển động của bánh xe để ước lượng tốc độ di chuyển thực tế. Thông qua việc thu thập tín hiệu từ cảm biến gắn trực tiếp trên bánh xe, hệ thống có thể phản ánh chính xác mức độ vận hành của phương tiện trong từng khoảng thời gian ngắn. Dữ liệu tốc độ sau khi được xử lý sẽ được hiển thị trực tiếp trên màn hình LCD, cho phép người lái dễ dàng quan sát mà không gây xao nhãng.

Với vai trò như một công cụ hỗ trợ, hệ thống giám sát tốc độ bên trong không chỉ giúp người điều khiển hạn chế các vi phạm do vô ý mà còn góp phần nâng cao ý thức chấp hành



luật giao thông. Việc chủ động kiểm soát tốc độ trong quá trình lái xe giúp giảm nguy cơ tai nạn, đặc biệt khi di chuyển trong khu vực đông dân cư hoặc những đoạn đường có giới hạn tốc độ nghiêm ngặt.

### **3.2. Nguyên lý đo và tính toán tốc độ**

#### **3.2.1. Trường hợp bên ngoài đường**

Bài toán phát hiện phương tiện vượt quá tốc độ được xây dựng dựa trên nguyên lý đo thời gian di chuyển của phương tiện giữa hai điểm cố định. Hệ thống sử dụng hai cặp nguồn phát và cảm biến ánh sáng, trong đó nguồn phát là tia laser và cảm biến có nhiệm vụ thu nhận tín hiệu ánh sáng tương ứng.

Khi phương tiện di chuyển qua vị trí của cảm biến thứ nhất, tia laser bị che khuất, làm cho cảm biến không còn nhận được tín hiệu ánh sáng. Thời điểm này được hệ thống ghi nhận và quy ước là thời điểm ban đầu  $t_1$ . Sau đó, khi phương tiện tiếp tục di chuyển và đi qua vị trí của cảm biến thứ hai, hiện tượng che khuất tia laser tương tự xảy ra và thời điểm này được ghi nhận là  $t_2$ . Hiệu thời gian giữa hai thời điểm trên được xác định là  $\Delta t = t_2 - t_1$ .

Dựa trên quãng đường đã biết trước giữa hai cảm biến và khoảng thời gian  $\Delta t$ , vận tốc của phương tiện được tính theo công thức:

$$v = \frac{s}{\Delta t}$$

trong đó  $s$  là khoảng cách giữa hai cảm biến.

Vận tốc tính toán được sẽ được hệ thống xử lý để đánh giá xem phương tiện có vượt quá ngưỡng tốc độ cho phép hay không. Trong trường hợp phát hiện phương tiện vượt tốc độ, mô-đun ESP32-CAM sẽ được kích hoạt để thu thập dữ liệu hình ảnh, bao gồm hình ảnh phương tiện và biển số. Các dữ liệu này sau đó được lưu trữ hoặc gửi lên hệ thống quản lý để phục vụ công tác giám sát và xử lý.

Bên cạnh đó, giá trị vận tốc của phương tiện được hiển thị trực tiếp trên màn hình LCD, giúp người vận hành có thể theo dõi và xác nhận một cách trực quan và chính xác tình trạng vượt quá tốc độ của phương tiện.

#### **3.2.2. Trường hợp bên trong xe**

Khác với phương pháp xác định vận tốc dựa trên việc đo khoảng thời gian di chuyển giữa hai điểm cố định, thuật toán đo công tơ mét trong phương tiện xác định tốc độ dựa trên tần số xung thu được trong một cửa sổ thời gian xác định (Time Window). Thuật toán này được xây dựng theo phương pháp đếm xung (Pulse Counting), trong đó các xung được tạo ra từ cảm biến Hall gắn với bánh xe.



Khi bánh xe quay, cảm biến Hall sẽ phát sinh các xung tương ứng với mỗi vòng quay hoặc mỗi phần của vòng quay, tùy thuộc vào cấu hình cảm biến. Số lượng xung thu được trong một khoảng thời gian nhất định phản ánh trực tiếp tốc độ quay của bánh xe, từ đó cho phép suy ra vận tốc di chuyển của phương tiện.

Quy trình đo và tính toán tốc độ được thực hiện như sau:

1. Hệ thống khởi tạo một biến đếm **pulse\_count**, biến này được tăng lên một đơn vị mỗi khi vi điều khiển nhận được tín hiệu ngắt từ cảm biến Hall.
2. Một bộ định thời mềm (software timer) được sử dụng để tạo chu kỳ đo cố định  $T = 1\text{ s}$  (tương đương 1000 ms).
3. Tại mỗi chu kỳ  $T$ , hệ thống tính số vòng quay trên phút (RPM) của bánh xe theo công thức:

$$\text{RPM} = \frac{\text{pulse\_count}}{N} \times \frac{60}{T}$$

trong đó  $N$  là số xung tương ứng với một vòng quay hoàn chỉnh của bánh xe.

4. Từ giá trị RPM thu được, vận tốc dài của phương tiện được tính dựa trên chu vi bánh xe  $C$  theo công thức:

$$v = \frac{\text{RPM} \times C}{60}$$

trong đó  $v$  là vận tốc dài của phương tiện.

5. Sau khi hoàn tất quá trình tính toán, biến pulse\_count được đặt lại về 0 để bắt đầu chu kỳ đo tiếp theo.

Vận tốc tính toán được sẽ được hiển thị trên màn hình LCD theo thời gian thực, đồng thời có thể được so sánh với ngưỡng tốc độ cho phép nhằm hỗ trợ người lái xe trong việc kiểm soát và điều chỉnh tốc độ phù hợp khi tham gia giao thông.

### 3.3. Nguyên lý hoạt động của cảm biến và thiết bị sử dụng

Hệ thống được thiết kế dựa trên sự phối hợp giữa các cảm biến quang học và cảm biến từ trường để thu thập dữ liệu đầu vào, sau đó được xử lý bởi vi điều khiển trung tâm ESP32-CAM. Nguyên lý hoạt động cụ thể của từng khối như sau:

#### 3.3.1. Khối cảm biến quang (Laser Speed Trap)

Hệ thống sử dụng mô hình "ngắt tia" (Beam Break) để đo tốc độ tức thời của vật thể đi qua.

- **Cấu tạo:** Gồm 2 cặp thu phát Laser đặt song song, cách nhau một khoảng cố định (trong mô hình là 20cm).
- **Nguyên lý:** Ở trạng thái bình thường, tia Laser chiếu thẳng vào cảm biến quang trở (hoặc photodiode), mạch đầu ra duy trì mức logic HIGH (hoặc LOW tùy mạch). Khi xe đi qua, nó lần lượt chắn tia Laser 1 và tia Laser 2. Sự thay đổi cường độ ánh sáng đột ngột làm thay đổi điện áp đầu ra, tạo thành các sườn xung (Edge).
- **Giao tiếp:** Tín hiệu này được đưa vào chân GPIO của ESP32. Vì điều khiển được cấu hình để bắt sườn xuống (Falling Edge) – tức là thời điểm tia sáng bị cắt – để kích hoạt ngắt (Interrupt).

### 3.3.2. Khối cảm biến Hall (Speedometer)

Để đo tốc độ của chính phương tiện (xe chủ), hệ thống sử dụng cảm biến Hall (Module KY-003 hoặc A3144) kết hợp với nam châm vĩnh cửu gắn trên bánh xe.

- **Hiệu ứng Hall:** Khi nam châm quét qua bề mặt cảm biến, từ trường biến thiên sinh ra một hiệu điện thế Hall. Mạch so sánh bên trong module sẽ chuyển đổi tín hiệu này thành xung vuông Digital.
- **Nguyên lý:** Mỗi khi bánh xe quay hết một vòng, nam châm đi qua cảm biến một lần, tạo ra một xung tín hiệu logic thấp (Low Pulse). Tần số của chuỗi xung này tỷ lệ thuận với vận tốc quay của bánh xe.

## 3.4. Xử lý tín hiệu và thuật toán trong hệ thống nhúng

Để đảm bảo tính chính xác của phép đo và khả năng phản hồi thời gian thực (Real-time response), phần mềm được xây dựng trên nền tảng hệ điều hành thời gian thực FreeRTOS với các thuật toán xử lý tín hiệu sau:

### 3.4.1. Cơ chế ngắt (Interrupt Service Routine - ISR)

Do tốc độ xe mô hình di chuyển nhanh, việc sử dụng vòng lặp (polling) để đọc trạng thái cảm biến sẽ gây ra sai số lớn hoặc bỏ lỡ tín hiệu.

- **Hiệu ứng Hall:** Khi nam châm quét qua bề mặt cảm biến, từ trường biến thiên sinh ra một hiệu điện thế Hall. Mạch so sánh bên trong module sẽ chuyển đổi tín hiệu này thành xung vuông Digital.
- **Giải pháp:** Hệ thống sử dụng ngắt ngoài (External Interrupt) trên các chân GPIO.
- **Quy trình:** Khi có tín hiệu từ cảm biến Laser hoặc Hall, CPU sẽ tạm dừng tác vụ hiện tại để thực thi hàm ngắt (`sensor_isr_handler`). Trong hàm ngắt, hệ thống chỉ thực hiện việc ghi lại nhãn thời gian thực (Timestamp) bằng hàm `esp_timer_get_time()` (độ chính

xác micro giây) và đẩy sự kiện vào hàng đợi (Queue). Việc tính toán phức tạp được nhường lại cho Task chính để không làm nghẽn hệ thống.

### 3.4.2. Quản lý đa nhiệm và Máy trạng thái (Finite State Machine)

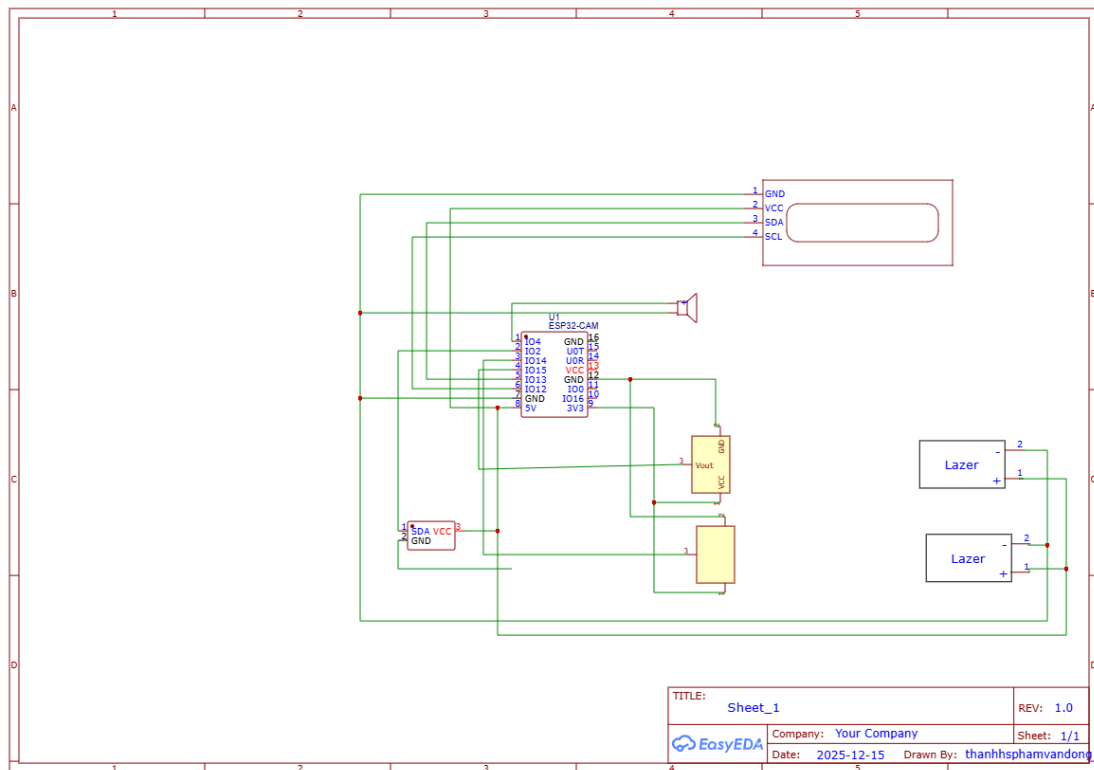
Do hệ thống hoạt động ở hai chế độ riêng biệt, một máy trạng thái đơn giản được thiết lập:

- **Hiệu ứng Hall:** Khi nam châm quét qua bề mặt cảm biến, từ trường biến thiên sinh ra một hiệu điện thế Hall. Mạch so sánh bên trong module sẽ chuyển đổi tín hiệu này thành xung vuông Digital.
- **Mode A (Trap):** Chỉ kích hoạt tính toán khi nhận đủ cặp tín hiệu từ Laser 1 và Laser 2.
- **Mode B (Meter):** Chạy vòng lặp định thời 1 giây để cập nhật tốc độ.
- **Chuyển mạch:** Nút nhấn (Button) sử dụng ngắt để thay đổi biến trạng thái `current_mode`. Để tránh hiện tượng "dội phím" (bouncing) cơ học, một thuật toán Debounce phần mềm được áp dụng (bỏ qua các tín hiệu nhiễu trong khoảng 200-500ms sau khi nhấn).

## CHƯƠNG 4. ỨNG DỤNG

### 4.1. Sơ đồ khối và nguyên tắc hoạt động của hệ thống

#### 4.1.1. Sơ đồ khối của hệ thống



Hình 13. Sơ đồ kết nối của hệ thống

#### 4.1.2. Nguyên tắc hoạt động của hệ thống

Nguyên tắc hoạt động của hệ thống được xây dựng dựa trên vi điều khiển ESP32-CAM đóng vai trò là khối xử lý trung tâm. Toàn bộ quá trình vận hành bao gồm các bước thu thập dữ liệu, xử lý tín hiệu, hiển thị thông tin và cảnh báo, được tổ chức rõ ràng theo từng chức năng cụ thể như sau:

##### + Khởi tạo và quản lý chế độ hoạt động

Khi hệ thống được cấp nguồn, ESP32 tiến hành khởi tạo các chân GPIO, cảm biến và thiết bị ngoại vi. Chân GPIO2 được cấu hình để đọc tín hiệu từ nút nhấn chuyển đổi chế độ trên mạch LCD.

- Khi nút được nhấn, ESP32 xác định trạng thái logic tại GPIO2.
- Hệ thống cho phép chuyển đổi giữa hai chế độ đo: Trap (đo bên ngoài) và Meter (đo bên trong).

- Chế độ hoạt động hiện tại được cập nhật và hiển thị trực tiếp trên màn hình LCD để người dùng theo dõi.

#### **+ Thu thập tín hiệu từ cảm biến đo tốc độ**

Hệ thống sử dụng mạch cảm biến ánh sáng để phát hiện sự di chuyển của phương tiện.

- Hai chân GPIO14 và GPIO15 được sử dụng để thu thập tín hiệu từ cảm biến.
- Khi phương tiện đi qua vùng đo, sự thay đổi tín hiệu ánh sáng được cảm biến ghi nhận và truyền về ESP32.
- Các tín hiệu này là cơ sở để xác định thời điểm và khoảng thời gian phương tiện đi qua vùng đo.

#### **+ Xử lý và tính toán tốc độ**

Dựa trên dữ liệu thu thập được từ cảm biến, ESP32 tiến hành:

- Xử lý tín hiệu đầu vào nhằm loại bỏ nhiễu và xác định chính xác các mốc thời gian.
- Tính toán tốc độ phương tiện dựa trên thời gian di chuyển và khoảng cách đã được thiết lập trước.
- So sánh tốc độ đo được với ngưỡng tốc độ cho phép tương ứng với từng chế độ hoạt động.

#### **+ Hiển thị thông tin lên màn hình LCD**

Thông tin vận hành của hệ thống được truyền đến màn hình LCD thông qua các chân GPIO12 và GPIO13.

- Hiển thị chế độ hoạt động hiện tại (Trap hoặc Meter).
- Hiển thị giá trị tốc độ đo được và các trạng thái liên quan.
- Đảm bảo dữ liệu được cập nhật theo thời gian thực, hỗ trợ người dùng giám sát hệ thống.

#### **+ Cảnh báo khi vượt quá tốc độ cho phép**

Khi phát hiện phương tiện vượt quá ngưỡng tốc độ cài đặt:

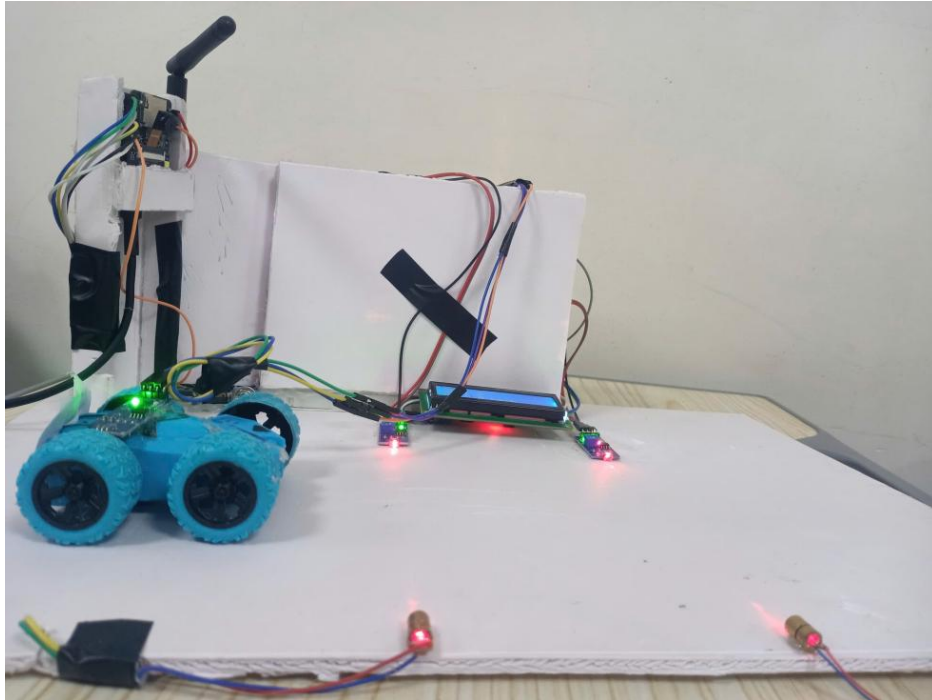
- ESP32 kích hoạt quy trình xử lý vi phạm, bao gồm chụp ảnh và lưu trữ hoặc truyền dữ liệu (nếu sử dụng chức năng camera).
- Sau khi quá trình xử lý hoàn tất, chân GPIO4 được sử dụng để xuất tín hiệu điều khiển còi báo.

- Còi phát âm thanh nhằm cảnh báo trực quan về hành vi vượt tốc độ.

Toàn bộ hệ thống hoạt động theo chu trình thu thập dữ liệu → xử lý → hiển thị → cảnh báo. Việc phân chia chức năng rõ ràng cho từng khối và từng chân GPIO giúp hệ thống hoạt động ổn định, dễ mở rộng và phù hợp cho các ứng dụng giám sát và phát hiện phương tiện vượt tốc độ trong thực tế.

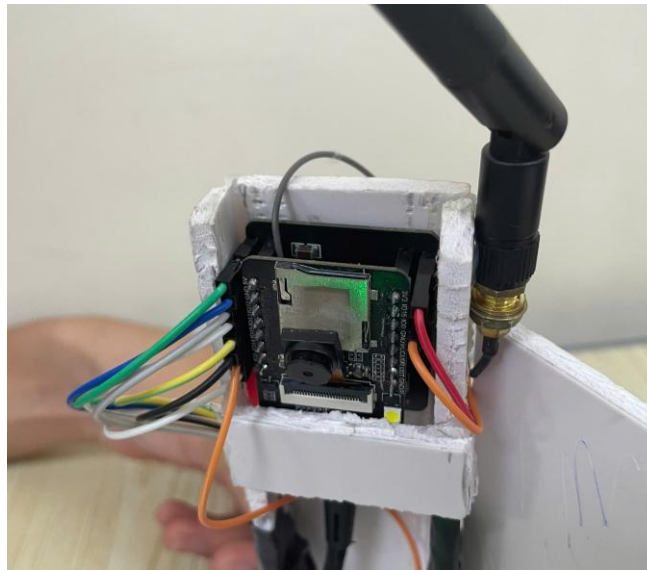
#### 4.2. Thiết kế và kết nối phần cứng ứng dụng

- + Kết nối phần cứng tổng thể

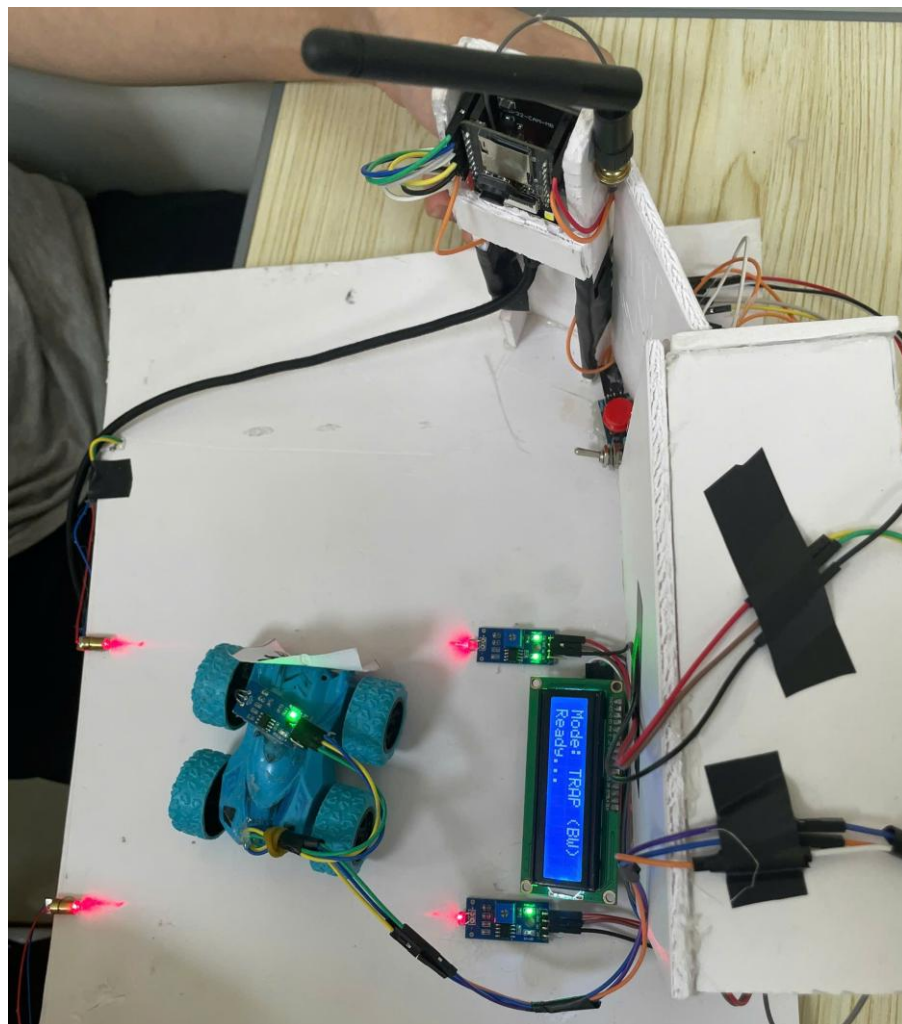


*Hình 14. Kết nối các phần cứng với nhau*

- + Kết nối ESP32 với các cảm biến



*Hình 15. Kết nối các cảm biến, mạch với ESP32*



*Hình 16. Các module được kết nối*

### 4.3. Xây dựng chương trình, nạp code

#### a) Chương trình xử lý trên ESP32-CAM

Chương trình trên ESP32-CAM được tổ chức theo mô hình đa tác vụ, trong đó mỗi chức năng chính được triển khai thành một task độc lập dưới sự quản lý của FreeRTOS, bao gồm:

- Task xử lý tín hiệu cảm biến
- Task xử lý logic chính
- Task giao tiếp mạng và upload dữ liệu

Các tác vụ này trao đổi dữ liệu với nhau thông qua hàng đợi (Queue) và cờ sự kiện (Event Group), giúp hệ thống vận hành ổn định và không phụ thuộc vào xử lý tuần tự.

Khi hệ thống được cấp nguồn, chương trình thực hiện đồng thời các bước khởi tạo sau:

- Đọc dữ liệu cấu hình từ bộ nhớ NVS để khôi phục chế độ hoạt động trước đó (Trap hoặc Meter).
- Khởi tạo module camera với cấu hình tốc độ chụp cao, phù hợp cho việc bắt các đối tượng chuyển động nhanh.
- Khởi tạo kết nối WiFi ở chế độ nền (non-blocking).
- Cấu hình các chân cảm biến và nút nhấn ở chế độ ngắt ngoài (Interrupt).
- Cập nhật trạng thái sẵn sàng (Ready) lên màn hình LCD.

Khi phương tiện đi qua cảm biến, ngắt phần cứng được kích hoạt. Tại đây, hệ thống chỉ thực hiện các thao tác tối thiểu:

- Ghi lại mốc thời gian chính xác đến micro giây.
- Đóng gói dữ liệu sự kiện và đưa vào hàng đợi.

Việc tính toán chi tiết được chuyển sang task logic chính nhằm tránh làm gián đoạn các tác vụ khác.

Task logic chính đóng vai trò trung tâm, liên tục đọc dữ liệu từ hàng đợi và xử lý theo chế độ hoạt động:

- **Chế độ Trap (đo tốc độ ngoài):**

Hệ thống tính toán vận tốc dựa trên thời gian xe đi qua hai cảm biến. Nếu vận tốc vượt quá ngưỡng cho phép, ESP32-CAM sẽ kích hoạt chuỗi hành động gồm: chụp ảnh, cảnh báo bằng còi buzzer, hiển thị cảnh báo trên LCD và gửi yêu cầu upload dữ liệu.



- **Chế độ Meter (đo vận tốc/RPM):**

Hệ thống chỉ sử dụng một cảm biến để đếm xung trong một khoảng thời gian cố định (1 giây), từ đó tính toán RPM và vận tốc, sau đó cập nhật liên tục lên màn hình LCD.

Nút nhấn vật lý được giám sát liên tục. Khi nút được giữ quá 3 giây, hệ thống sẽ chuyển đổi giữa hai chế độ Trap và Meter, đồng thời lưu trạng thái mới vào NVS và reset các biến liên quan để đảm bảo kết quả đo chính xác.

Tác vụ truyền dữ liệu mạng được tách riêng hoàn toàn. Khi có yêu cầu upload, hệ thống sẽ gửi ảnh và thông tin tốc độ tới server thông qua giao thức HTTP POST. Sau khi hoàn thành, task này tự kết thúc để giải phóng tài nguyên bộ nhớ.

## **b) Chương trình xử lý trên Server trung gian**

Server được xây dựng theo mô hình Hybrid Local–Cloud, sử dụng Python và framework Flask. Server cục bộ đóng vai trò là cầu nối trung gian giữa ESP32 và Internet thông qua Ngrok.

Server lắng nghe yêu cầu HTTP POST từ ESP32 trên mạng LAN, tiếp nhận ảnh và dữ liệu tốc độ, sau đó:

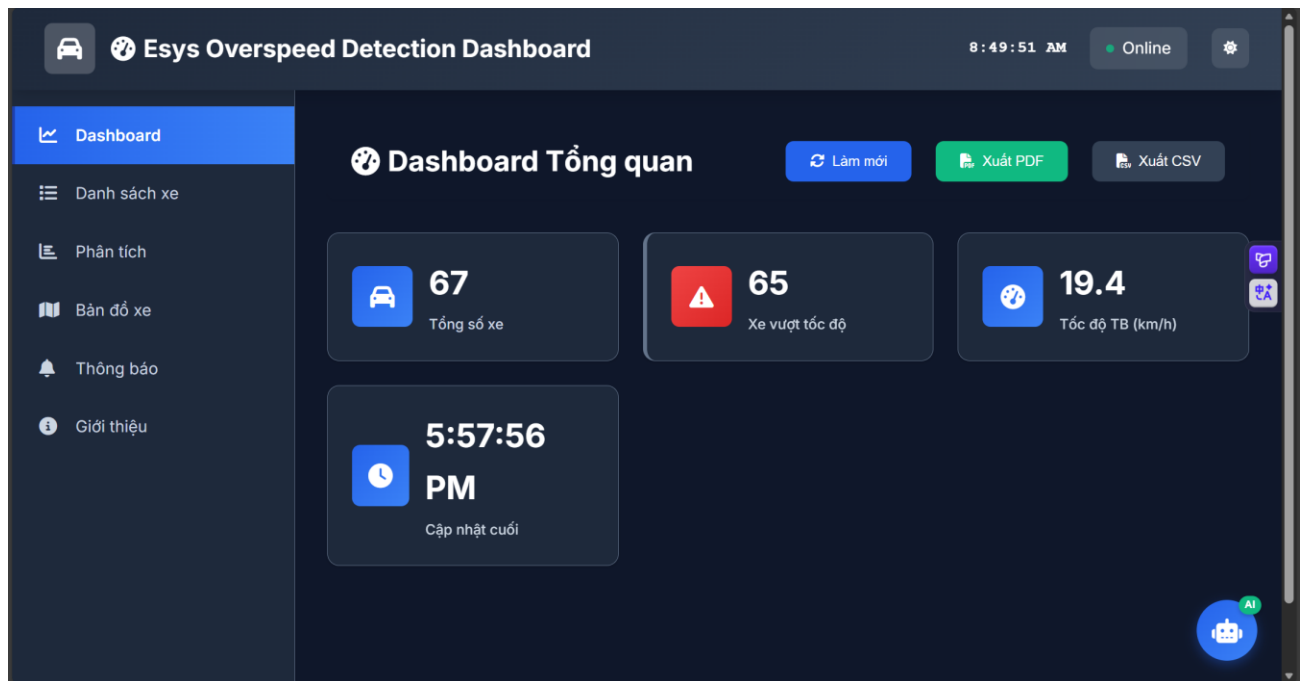
- Lưu ảnh trực tiếp vào ổ cứng máy tính với tên file chứa timestamp để phục vụ lưu trữ bằng chứng.
- Thực hiện chuyển tiếp dữ liệu lên server Cloud thông qua đường hầm Ngrok.

Cơ chế xử lý lỗi được triển khai bằng khối try–except nhằm đảm bảo server không bị dừng đột ngột khi xảy ra sự cố mạng.

Sau khi hoàn thiện mã nguồn, chương trình được biên dịch và nạp vào ESP32-CAM thông qua Arduino IDE. Hệ thống được kiểm thử ở cả hai chế độ Trap và Meter để đánh giá độ chính xác đo tốc độ, khả năng chụp ảnh, cảnh báo và truyền dữ liệu. Kết quả cho thấy hệ thống hoạt động ổn định, đáp ứng yêu cầu đo tốc độ và giám sát phương tiện theo thời gian thực.

## **4.4. Triển khai giao diện kết quả**

Ứng dụng được xây dựng dưới dạng dashboard quản lý dữ liệu tập trung, có nhiệm vụ thu nhận, lưu trữ, hiển thị và phân tích các dữ liệu vi phạm tốc độ được gửi lên từ phần cứng đo (ESP32 và các module liên quan).



Hình 17. Giao diện tổng quan của Dashboard

Bên trong trang web có tích hợp các tính năng hỗ trợ kiểm soát dữ liệu cho người dùng.

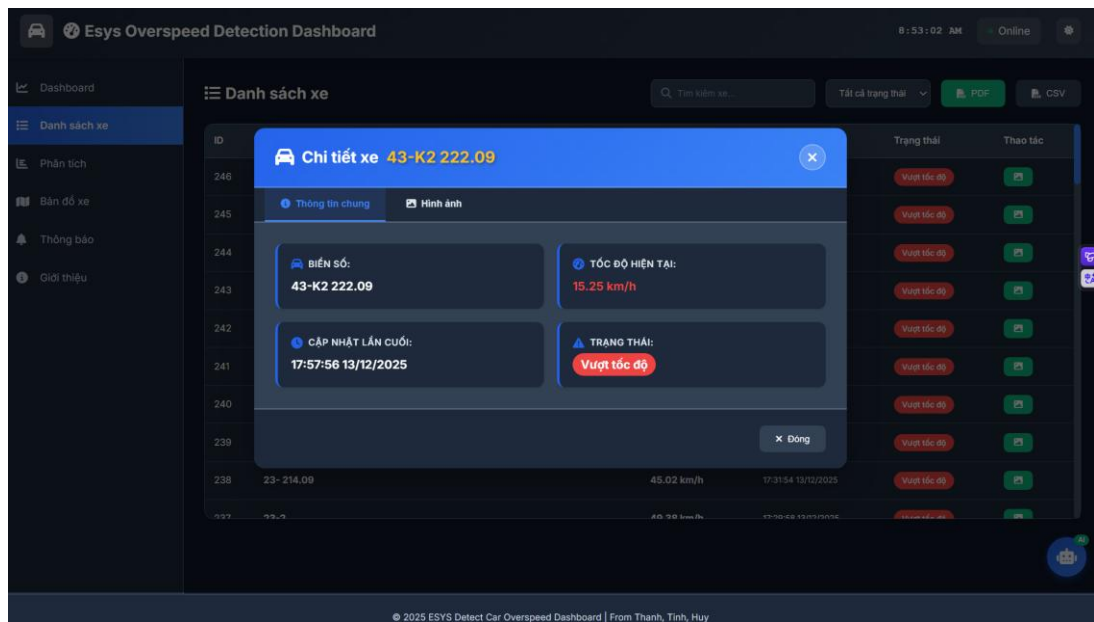
1. Quản lý danh sách xe vi phạm tốc độ theo thời gian thực.

The screenshot shows the 'Danh sách xe' (Vehicle List) section of the dashboard. It includes a search bar, a dropdown for 'Tất cả trạng thái', and buttons for 'PDF' and 'CSV'. The table below lists vehicles with their IDs, license plates, speeds, timestamps, status, and actions.

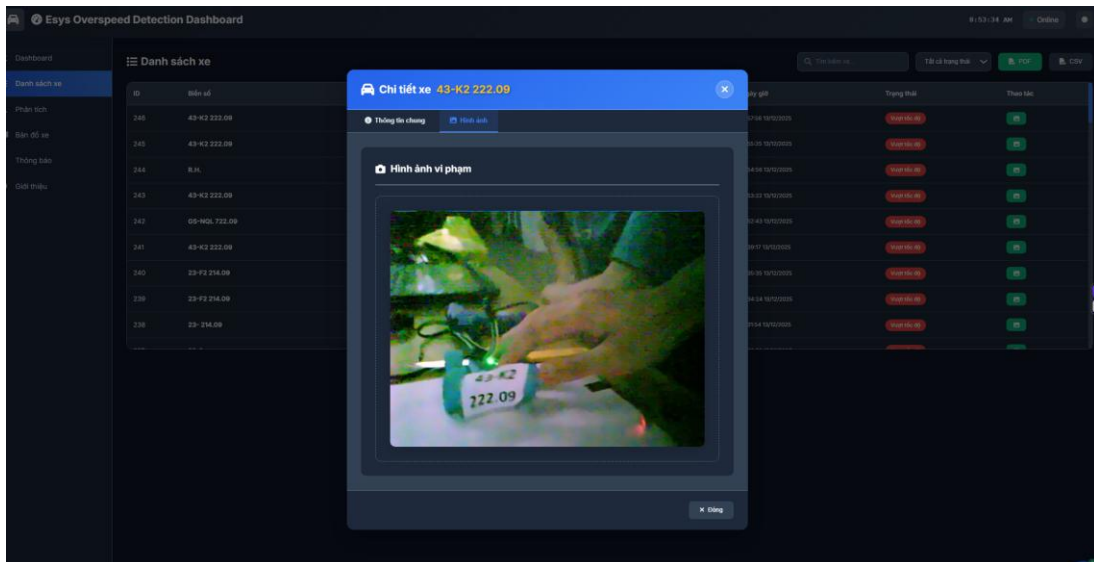
ID	Biển số	Tốc độ (km/h)	Ngày giờ	Trạng thái	Thao tác
246	43-K2 222.09	15.25 km/h	17:57:56 13/12/2025	Vượt tốc độ	
245	43-K2 222.09	16.47 km/h	17:55:25 13/12/2025	Vượt tốc độ	
244	R.H.	21.72 km/h	17:54:56 13/12/2025	Vượt tốc độ	
243	43-K2 222.09	7.5 km/h	17:53:22 13/12/2025	Vượt tốc độ	
242	GS-NQL 722.09	14.19 km/h	17:52:43 13/12/2025	Vượt tốc độ	
241	43-K2 222.09	29.57 km/h	17:39:17 13/12/2025	Vượt tốc độ	
240	23-F2 214.09	14.52 km/h	17:35:35 13/12/2025	Vượt tốc độ	
239	23-F2 214.09	65.93 km/h	17:34:24 13/12/2025	Vượt tốc độ	
238	23- 214.09	45.02 km/h	17:31:54 13/12/2025	Vượt tốc độ	
237	23- 214.09	49.98 km/h	17:30:48 13/12/2025	Vượt tốc độ	

Hình 18. Giao diện danh sách xe

2. Hiện thị hình ảnh phương tiện vi phạm được trích xuất từ camera.



Hình 19. Giao diện thông tin xe vi phạm



Hình 20. Hình ảnh xe vi phạm

3. Nhận diện biển số xe và lưu trữ thông tin vi phạm gồm: thời gian, tốc độ đo được, giới hạn tốc độ, mức vượt



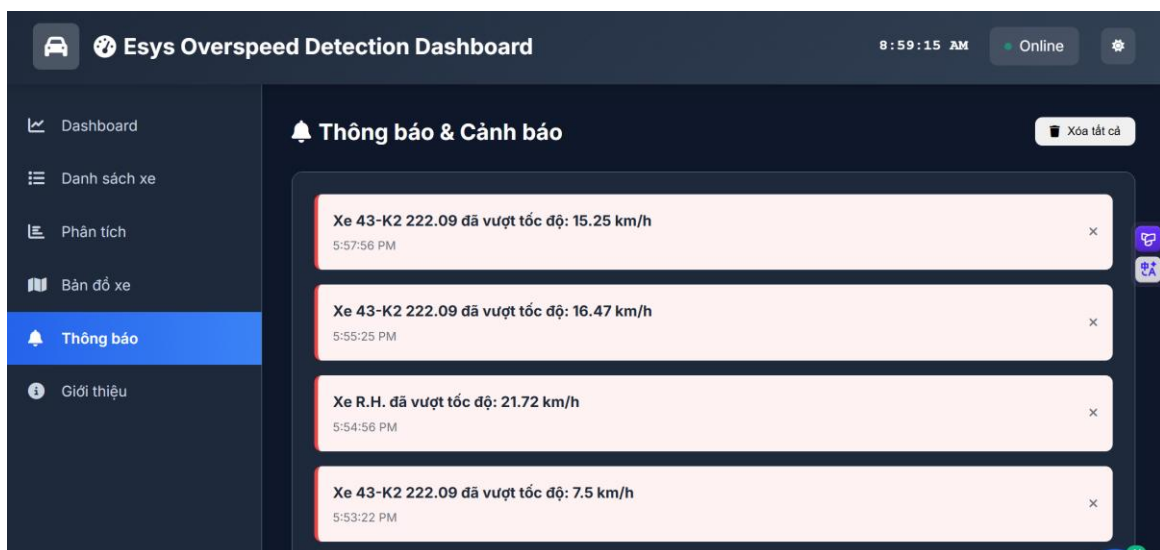
Hình 21. Biển số xe được nhận diện

4. Hiển thị biểu đồ thống kê và phân tích dữ liệu vi phạm theo khung giờ.



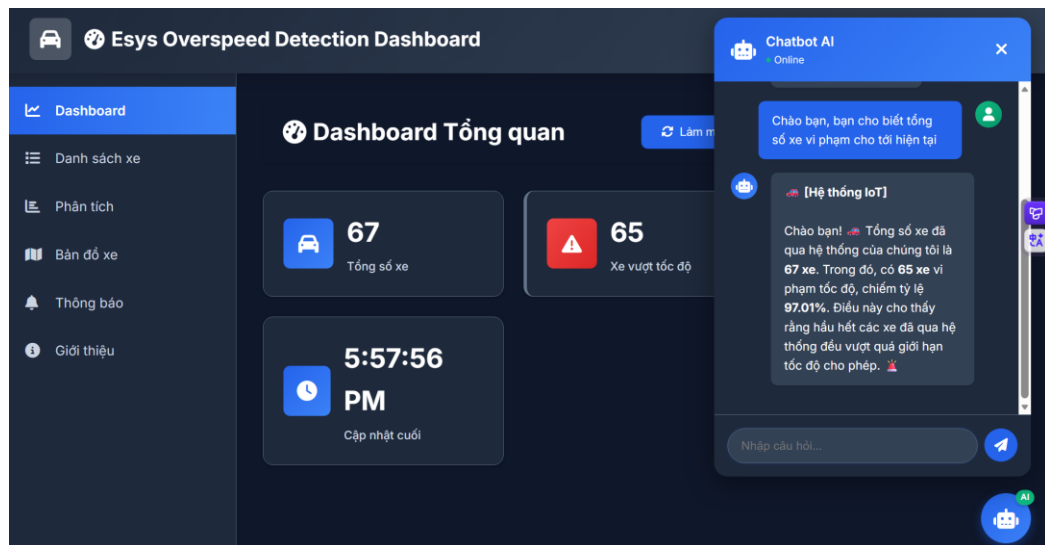
Hình 22. Biểu đồ thống kê vi phạm

5. Cảnh báo popup vi phạm khi có dữ liệu mới được ghi nhận.



Hình 23. Popup cảnh báo vi phạm

6. Tích hợp Chatbot AI hỗ trợ truy vấn và giải thích dữ liệu vi phạm.



Hình 24. ChatBot AI hỗ trợ giải đáp

7. Xuất dữ liệu bằng các định dạng pdf, csv,... để lưu trữ.

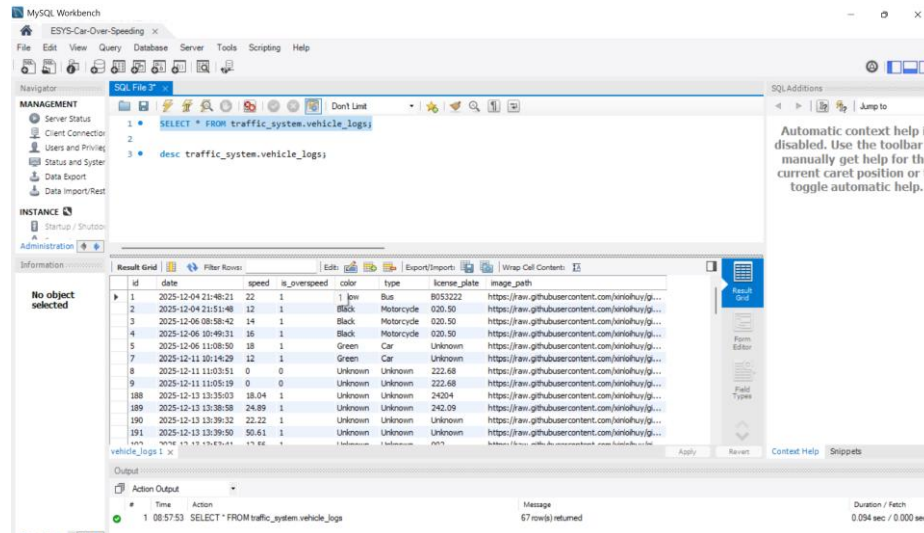
**DANH SÁCH XE**

Thời gian xuất: 09:01:44 15/12/2025  
Tổng số: 67 xe

#	Biển số	Tốc độ	Trạng thái	Thời gian
1	43-K2 222.09	15.3	Vượt	17:57 13/12/2025
2	43-K2 222.09	16.5	Vượt	17:55 13/12/2025
3	R.H.	21.7	Vượt	17:54 13/12/2025
4	43-K2 222.09	7.5	Vượt	17:53 13/12/2025
5	GS-NQL 722.09	14.2	Vượt	17:52 13/12/2025
6	43-K2 222.09	29.6	Vượt	17:39 13/12/2025
7	23-F2 214.09	14.5	Vượt	17:35 13/12/2025
8	23-F2 214.09	65.9	Vượt	17:34 13/12/2025
9	23- 214.09	45.0	Vượt	17:31 13/12/2025
10	23-2	49.4	Vượt	17:29 13/12/2025
11	12322214.0909_2322323_21 4.09214.0921409.23214.09	16.0	Vượt	17:26 13/12/2025
12	1118	17.0	Vượt	17:24 13/12/2025

Hình 25. Xuất file pdf cho việc lưu trữ

8. Kết nối cơ sở dữ liệu (Database) để lưu trữ lâu dài và truy xuất dữ liệu.

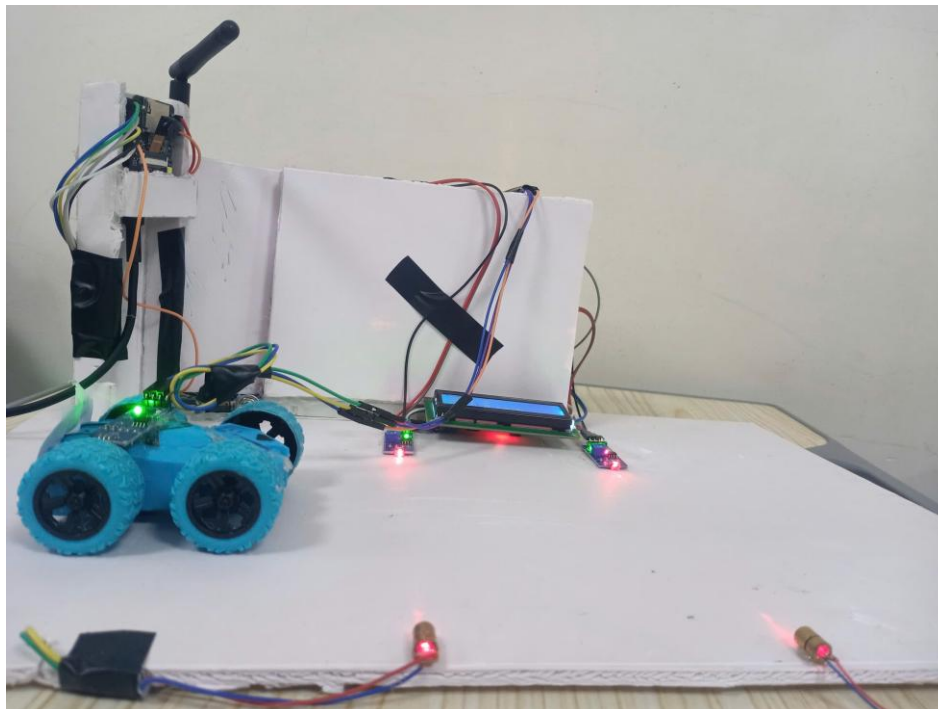


Hình 26. Cơ sở dữ liệu được tổ chức

## 4.5. Kết quả thực hiện, khó khăn và hướng khắc phục

### 4.5.1. Kết quả thực hiện

Hệ thống giám sát và phát hiện phương tiện vượt quá tốc độ đã được xây dựng và vận hành theo đúng mục tiêu đề ra. Kết quả thử nghiệm cho thấy hệ thống có khả năng đo và tính toán tốc độ phương tiện một cách ổn định trong cả hai trường hợp: giám sát từ bên ngoài và giám sát bên trong phương tiện.



Hình 27. Mô hình hoàn thiện

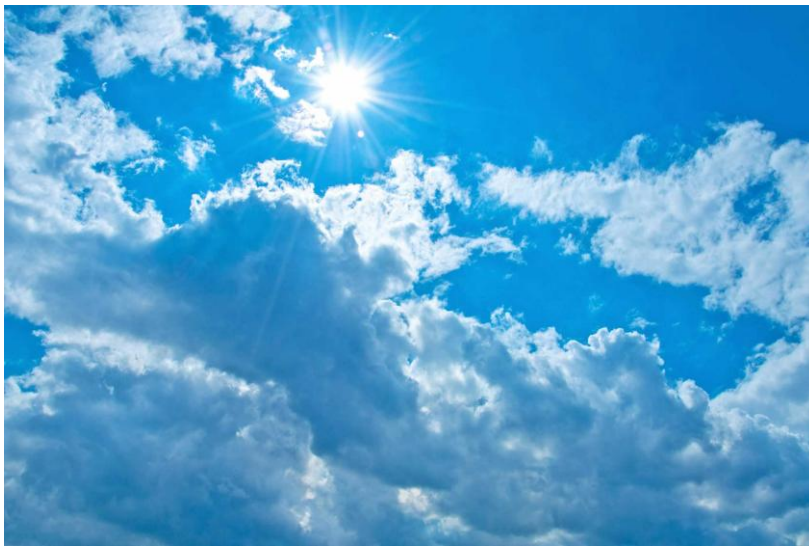
Đối với hệ thống bên ngoài, tốc độ phương tiện được xác định chính xác tại khu vực đo và các trường hợp vượt quá ngưỡng cho phép được phát hiện kịp thời. Khi xảy ra vi phạm, mô-đun ESP32-CAM hoạt động đúng chức năng, ghi nhận hình ảnh phương tiện kèm biển số, đồng thời hiển thị tốc độ đo được trên màn hình LCD.

Đối với hệ thống bên trong phương tiện, thuật toán đo công tơ mét dựa trên cảm biến Hall cho phép theo dõi tốc độ theo thời gian thực. Thông tin tốc độ được hiển thị liên tục, giúp người lái xe dễ dàng nhận biết và chủ động điều chỉnh vận tốc khi cần thiết. Nhìn chung, hệ thống đáp ứng được yêu cầu về tính trực quan, tính tự động và khả năng hỗ trợ giám sát tốc độ trong thực tế.

#### **4.5.2. Các khó khăn trong quá trình thực hiện**

Trong quá trình triển khai, nhóm thực hiện gặp một số khó khăn đáng chú ý.

Thứ nhất, độ ổn định của tín hiệu từ cảm biến (cảm biến ánh sáng và cảm biến Hall) chịu ảnh hưởng bởi môi trường bên ngoài như ánh sáng mạnh, rung động cơ học và sai lệch vị trí lắp đặt.



*Hình 28. Ảnh hưởng bởi điều kiện ánh sáng*

Thứ hai, việc đồng bộ thời gian và xử lý ngắt trong vi điều khiển đòi hỏi cấu hình chính xác, nếu không có thể gây sai số trong quá trình đo tốc độ. Ngoài ra, chất lượng hình ảnh thu được từ ESP32-CAM trong điều kiện ánh sáng yếu hoặc khi phương tiện di chuyển nhanh còn hạn chế, ảnh hưởng đến khả năng nhận diện biển số.





*Hình 29. Chất lượng hình ảnh thu được chưa cao*

Bên cạnh đó, quá trình phát triển phần mềm cũng gặp nhiều trở ngại. Việc làm việc với mã nguồn trên nhiều máy tính khác nhau, cũng như chuyển đổi môi trường phát triển, đôi khi phát sinh các sự cố như khác biệt cấu hình, lỗi thư viện, phiên bản công cụ không đồng nhất hoặc mất đồng bộ mã nguồn. Những vấn đề này gây gián đoạn trong quá trình lập trình, kiểm thử và tích hợp hệ thống, làm kéo dài thời gian hoàn thiện sản phẩm.

#### **4.5.3. Hướng khắc phục và phát triển tương lai**

Để khắc phục các khó khăn nêu trên, hệ thống cần được hiệu chỉnh và cải tiến theo nhiều hướng.

Trước hết, việc che chắn và cố định chắc chắn các cảm biến, đồng thời hiệu chỉnh vị trí lắp đặt, sẽ giúp giảm nhiễu và nâng cao độ chính xác của tín hiệu.

Thứ hai, có thể áp dụng các kỹ thuật lọc nhiễu và xử lý tín hiệu trong phần mềm nhằm hạn chế sai số do rung động hoặc xung giả. Đối với mô-đun ESP32-CAM, việc tối ưu thông số camera, bổ sung nguồn sáng hỗ trợ hoặc nâng cấp mô-đun camera có độ phân giải cao hơn sẽ cải thiện chất lượng hình ảnh thu được.

Trong tương lai, hệ thống có thể được mở rộng bằng cách tích hợp truyền dữ liệu không dây, lưu trữ tập trung và ứng dụng nhận dạng biến số tự động nhằm nâng cao hiệu quả giám sát và khả năng ứng dụng trong thực tế.



## PHẦN KẾT LUẬN

Sau quá trình nghiên cứu và thực hiện, đề tài “Car Overspeeding Detection Project” đã đạt được các mục tiêu đề ra. Nhóm đã thiết kế và xây dựng thành công mô hình hệ thống phát hiện xe vượt quá tốc độ cho phép dựa trên vi điều khiển ESP32. Hệ thống có khả năng đo tốc độ phương tiện thông qua các cảm biến, xử lý dữ liệu và đưa ra cảnh báo khi phát hiện xe vượt quá ngưỡng tốc độ cài đặt. Kết quả thực nghiệm cho thấy hệ thống hoạt động ổn định, đáp ứng được yêu cầu cơ bản của đề tài trong cả hai trường hợp sử dụng bên trong xe và ngoài đường.

Bên cạnh các kết quả đạt được, hệ thống vẫn còn một số hạn chế nhất định. Độ chính xác của phép đo tốc độ còn phụ thuộc vào điều kiện môi trường như ánh sáng, vị trí lắp đặt cảm biến và khoảng cách giữa các module laser. Ngoài ra, mô hình mới chỉ dừng lại ở mức thử nghiệm, chưa được kiểm chứng trong điều kiện giao thông thực tế với nhiều yếu tố phức tạp.

Trong thời gian tới, đề tài có thể tiếp tục được phát triển và hoàn thiện hơn bằng cách cải tiến thuật toán xử lý nhằm giảm sai số đo, bổ sung các cảm biến có độ chính xác cao hơn, cũng như tích hợp các công nghệ truyền thông không dây để lưu trữ và giám sát dữ liệu từ xa. Ngoài ra, việc kết hợp camera và xử lý hình ảnh để nhận dạng phương tiện vi phạm cũng là một hướng phát triển tiềm năng, góp phần nâng cao tính ứng dụng và hiệu quả của hệ thống trong thực tế.

## TÀI LIỆU THAM KHẢO

- [1]. Kanavi, P., Chaithra, K. B., Chaitra, K. T., Bhoomika, M. G., & Lakshmi, C. T. (2022). *Automatic Vehicle Over Speed Detection Alert and Controlling System on Highway*. Published online 30-08-2022. Available at: <https://www.ijert.org/automatic-vehicle-over-speed-detection-alert-and-controlling-system-on-highway> (Accessed: 03 December 2025).
- [2]. Sreekanth, N., Spandana, I., Shravya, J., & Sravani, K. (2024). *Vehicle Over Speed Indication Based IoT*. *International Journal of Advanced Research in Science and Technology (IJARST)*, Vol. 14, Issue 10, pp. 170–174. Available at: <https://www.ijarst.in/public/uploads/paper/582441729063704.pdf> (Accessed: 03 December 2025).
- [3]. Chandorkar, M., Pednekar, S. & Bojewar, S. (2021) ‘*Vehicle detection and speed tracking*’, *International Journal of Engineering Research & Technology (IJERT)*, 10(05). Available at: <https://www.ijert.org/research/vehicle-detection-and-speed-tracking-IJERTV10IS050428.pdf>(Accessed: 03 December 2025).
- [4]. GeeksforGeeks. (2025). *Introduction of Microprocessor*. *GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/electronics-engineering/introduction-of-microprocessor/> (Accessed: 08 December 2025).
- [5]. Random Nerd Tutorials. (2025). *ESP32-CAM Video Streaming and Face Recognition with Arduino IDE*. Available at: <https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/> (Accessed: 08 December 2025).
- [6]. Handson Technology. (n.d.). *I2C Serial Interface 1602 LCD Module User Guide*. Available at: [https://www.handsontec.com/dataspecs/module/I2C\\_1602\\_LCD.pdf](https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf) (Accessed: 08 December 2025).