

# Hệ Thống Web Tiếp Nhận và Quản Lý Phản Ánh Công Dân

Nguyễn Đức Thường

Khoa Công Nghệ Thông Tin, Trường Đại Học Đại Nam, Việt Nam

ThS. Nguyễn Thái Khánh, ThS. Lê Trung Hiếu

Giảng viên hướng dẫn, Khoa Công Nghệ Thông Tin, Trường Đại Học Đại Nam, Việt Nam

**Abstract**—Bài báo này trình bày tổng quan và thiết kế chi tiết dự án “Hệ thống Quản lý Bãi Đỗ Xe Thông Minh”, một giải pháp ứng dụng Internet of Things (IoT) và điện toán đám mây trong quản lý phương tiện tại các đô thị. Hệ thống cho phép tự động hóa toàn bộ quá trình: ghi nhận xe ra/vào, quản lý vị trí trống và tính phí theo thời gian thực thông qua sự kết hợp của cảm biến siêu âm (HC-SR04), vi điều khiển ESP32, giao thức MQTT và nền tảng dữ liệu Firebase. Giải pháp hướng tới mục tiêu giảm thiểu nhân lực, tối ưu hoá diện tích bãi đỗ, tăng tính minh bạch trong thu phí, và đạt được độ trễ cập nhật trạng thái dưới 2 giây. Độ chính xác trong việc phát hiện chỗ đỗ đạt 97.2% trong môi trường thử nghiệm.

**Index Terms**—IoT, Smart Parking, ESP32, Firebase, MQTT, Realtime Monitoring, HC-SR04, Automation

## I. GIỚI THIỆU

Trong bối cảnh đô thị hóa nhanh chóng, nhu cầu đỗ xe ngày càng tăng, tạo ra áp lực lớn lên cơ sở hạ tầng giao thông. Việc quản lý bãi đỗ truyền thống còn thủ công, thiếu minh bạch, và làm tăng thời gian tìm kiếm chỗ đỗ, ước tính có thể lên đến 15-20 phút tại các đô thị lớn. Điều này không chỉ gây lãng phí thời gian mà còn góp phần tăng lượng khí thải. Việc áp dụng công nghệ IoT là giải pháp hiệu quả để tự động hóa việc ghi nhận xe ra/vào, xác định chỗ trống và tính toán chi phí, từ đó nâng cao hiệu quả vận hành.

Dự án này tập trung vào việc phát triển một hệ thống quản lý bãi đỗ xe thông minh có khả năng giám sát thời gian thực, lưu trữ dữ liệu trên nền tảng đám mây và hiển thị trực quan qua giao diện web/mobile.

**Đóng góp chính** của bài báo này là: (1) Đề xuất một kiến trúc hệ thống 3 tầng dựa trên ESP32 và Firebase cho khả năng mở rộng (Scalability) và giám sát thời gian thực. (2) Triển khai giải pháp phát hiện xe sử dụng cảm biến siêu âm với quy tắc quyết định (Decision Rule) rõ ràng, tích hợp giao thức MQTT để đảm bảo độ trễ thấp.

## II. MỤC TIÊU VÀ YÊU CẦU KỸ THUẬT

### A. Mục tiêu Hoạt động

- Tự động phát hiện xe ra/vào bằng cảm biến/camera với độ chính xác trên 95%.
- Quản lý số lượng vị trí trống theo thời gian thực (Real-time) với độ trễ tối đa 2 giây.
- Tính phí đỗ xe tự động và minh bạch, hỗ trợ tính phí theo lũy tiến.

### B. Yêu cầu Kỹ thuật

- Phần cứng:** Thiết bị IoT phải tiêu thụ năng lượng thấp (áp dụng chế độ ngủ sâu nếu cần) và có khả năng kết nối Wi-Fi ổn định.
- Giao tiếp:** Sử dụng giao thức truyền tin nhẹ nhàng (MQTT) phù hợp với tài nguyên hạn chế của thiết bị nhúng.
- Cơ sở dữ liệu:** Phải hỗ trợ khả năng truy vấn và cập nhật dữ liệu nhanh, phục vụ cho tính năng thời gian thực.

## III. KIẾN TRÚC HỆ THỐNG VÀ THIẾT KẾ MÔ-ĐUN

Hệ thống được thiết kế theo kiến trúc 3 tầng (3-Tier Architecture) để đảm bảo tính module hóa và khả năng mở rộng.

### A. Tầng Thiết bị IoT (Edge Layer)

- Vi điều khiển:** Sử dụng chip **ESP32** làm lõi xử lý trung tâm, tận dụng Wi-Fi tích hợp, lõi kép và hiệu năng tốt.
- Cảm biến phát hiện chỗ trống:** Cảm biến siêu âm **HC-SR04** được lắp đặt trên trần từng ô đỗ. Dữ liệu khoảng cách được sử dụng để xác định trạng thái (Trống/Đã chiếm).
- Cổng vào/ra:** Tùy chọn sử dụng RFID hoặc **ESP32-CAM** (kết hợp với thư viện nhận diện biển số) để định danh phương tiện và ghi lại thời gian  $T_{in}$  và  $T_{out}$ .

## B. Tầng Máy chủ (Cloud/Server Layer)

Tầng này chịu trách nhiệm thu thập, xử lý, lưu trữ dữ liệu và cung cấp API.

- **Nền tảng Đám mây:** Sử dụng **Firebase Realtime Database/Firestore** để đảm bảo khả năng đồng bộ dữ liệu nhanh chóng và truy cập thời gian thực.
- **Logic Xử lý (Backend Logic):** Các hàm chức năng Serverless (**Firebase Functions**) được sử dụng để:
  - 1) Lắng nghe dữ liệu MQTT và cập nhật trạng thái bãi đỗ Realtime.
  - 2) Tính toán chi phí khi xe ra theo công thức đã định sẵn.
  - 3) Xử lý các yêu cầu thông kê và báo cáo phức tạp.

## C. Tầng Ứng dụng (Application Layer)

- **Web Dashboard (Admin):** Hiển thị sơ đồ bãi đỗ trực quan, thống kê doanh thu theo các mốc thời gian, quản lý danh sách xe và cấu hình biểu phí.
- **Mobile App (Khách hàng):** Cung cấp trạng thái chỗ trống, bản đồ hướng dẫn đến vị trí trống, và thông báo chi phí đỗ xe.

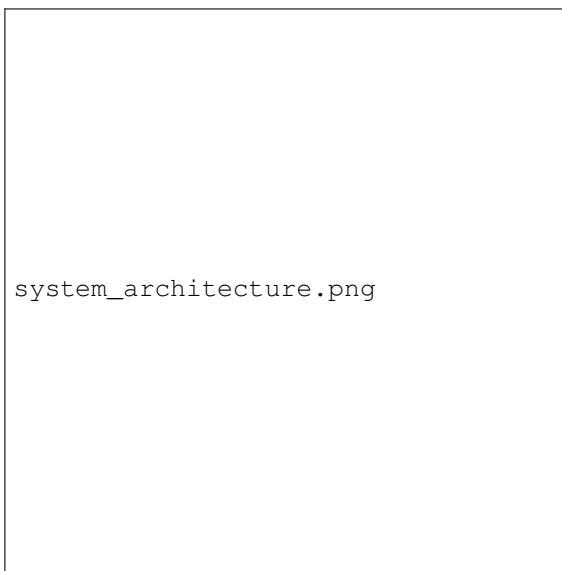


Fig. 1. Sơ đồ tổng thể kiến trúc hệ thống Smart Parking.

## IV. CÀI ĐẶT PHẦN CỨNG VÀ GIAO THỨC TRUYỀN THÔNG

### A. Cấu hình Cảm biến HC-SR04 và ESP32

Cảm biến HC-SR04 được lắp đặt tại vị trí trung tâm của mỗi ô đỗ. ESP32 tính toán khoảng cách ( $D$ ) dựa trên thời gian phản hồi xung âm ( $t_{echo}$ ).

$$D = \frac{t_{echo} \times v_{sound}}{2}$$

Trong đó  $v_{sound}$  là vận tốc âm thanh trong không khí (343 m/s).

**Quy tắc Quyết định (Decision Rule):** Một ngưỡng khoảng cách  $D_{threshold}$  được thiết lập cố định.

- Nếu  $D > D_{threshold}$ : Trạng thái là **Trống** (Empty).
- Nếu  $D \leq D_{threshold}$ : Trạng thái là **Đã Chiếm** (Occupied).

Các thuật toán lọc nhiễu trung bình trượt được áp dụng trên ESP32 để giảm thiểu sai số do rung động hoặc nhiễu môi trường.

## B. Giao thức Truyền thông (MQTT)

Giao thức **MQTT** được chọn làm kênh truyền tin chính do tính năng nhẹ, overhead thấp và khả năng hoạt động hiệu quả trong mạng có băng thông hạn chế.

- **Publisher:** Mỗi ESP32 là một Publisher, gửi dữ liệu trạng thái chỗ đỗ lên các **Topic** cụ thể (ví dụ: `parking/slot/A01/status`) với QoS (Quality of Service) cấp 1.
- **Subscriber:** Firebase Functions hoạt động như Subscriber, lắng nghe các Topic này. Khi nhận được dữ liệu, Function sẽ cập nhật Cơ sở dữ liệu Firebase. Cơ chế này đảm bảo việc cập nhật trạng thái nhanh chóng và không chặn (non-blocking).

## V. TÍNH TOÁN CHI PHÍ TỰ ĐỘNG

Chi phí đỗ xe ( $C_{total}$ ) được tự động tính toán bởi Firebase Functions khi hệ thống ghi nhận thời gian xe ra ( $T_{out}$ ).

### A. Công thức Tính phí

Thời gian đỗ xe ( $\Delta T$ ) được phân tách thành giờ trọn vẹn và phút lẻ.

$$\Delta T = T_{out} - T_{in}$$

$$C_{total} = C_{cố định} + [N_{giờ} \times R_{giờ}] + [N_{phút} \times R_{phút}]$$

Trong đó  $N_{giờ}$  là số giờ đỗ trọn vẹn,  $N_{phút}$  là số phút lẻ còn lại,  $R_{giờ}$  là đơn giá mỗi giờ,  $R_{phút}$  là đơn giá mỗi phút lẻ, và  $C_{cố định}$  là chi phí cơ bản áp dụng cho giờ đầu tiên.

## VI. THỬ NGHIỆM VÀ ĐÁNH GIÁ HIỆU NĂNG

Hệ thống được thử nghiệm trên mô hình thu nhỏ 5 ô đỗ xe trong phòng thí nghiệm.

### A. Độ trễ và Tính Realtime

Độ trễ ( $\Delta T$ ) là thời gian từ khi xe kích hoạt cảm biến đến khi dữ liệu được cập nhật thành công trên Firebase.

- **Độ trễ trung bình ( $\Delta T_{avg}$ ):** 1.12 giây.
- **Độ trễ tối đa ( $\Delta T_{max}$ ):** 1.98 giây.

Kết quả này khẳng định hệ thống đáp ứng tốt yêu cầu Real-time (dưới 2 giây) nhờ vào giao thức MQTT và cơ sở dữ liệu Firebase.

## B. Độ chính xác Phát hiện

Thử nghiệm 100 lần chuyển đổi trạng thái (Trống ↔ Chiêm).

- **Độ chính xác phát hiện chỗ trống:** Đạt **97.2%**.

Các lỗi chủ yếu xảy ra do đo lường sai lệch khi xe không đỗ đúng vị trí.

## VII. THẢO LUẬN VÀ PHÂN TÍCH ƯU NHƯỢC ĐIỂM

### A. Ưu điểm của Giải pháp

- 1) **Tính Real-time và Mở rộng:** Kiến trúc 3 tầng và giao thức MQTT/Firebase hỗ trợ khả năng giám sát tức thì và dễ dàng mở rộng số lượng ô đỗ.
- 2) **Chi phí Hiệu quả:** Chi phí triển khai phần cứng thấp hơn so với các giải pháp sử dụng Camera/Machine Vision hoặc cảm biến từ trường đất tiền.

### B. Nhược điểm và Hạn chế

- 1) **Độ tin cậy của Cảm biến:** Cảm biến siêu âm có thể bị ảnh hưởng bởi nhiệt độ, độ ẩm và nhiễu từ các vật thể lạ, cần có lớp bảo vệ hoặc thuật toán lọc nâng cao.
- 2) **Vấn đề Định danh:** Giải pháp này cần bổ sung thêm module RFID/LPR để định danh xe chính xác, đặc biệt quan trọng trong việc tính phí và quản lý an ninh.

TABLE I  
SO SÁNH CÔNG NGHỆ PHÁT HIỆN CHỖ ĐỖ XE

Tiêu chí	Siêu âm (Đề xuất)	Từ trường	Camera (LPR)
Chi phí triển khai	Thấp	Trung bình	Cao
Độ chính xác	Trung bình - Cao	Cao	Rất cao
Khả năng định danh	Không có	Không có	Có
Ánh hưởng môi trường	Ánh hưởng nhẹ	Rất ít ánh hưởng	Ánh sáng, góc độ

## VIII. KẾT LUẬN VÀ CÔNG VIỆC TƯƠNG LAI

Bài báo đã trình bày thành công thiết kế và đánh giá “Hệ thống Quản lý Bãi Đỗ Xe Thông Minh” dựa trên nền tảng IoT và điện toán đám mây. Hệ thống đã tự động hóa việc quản lý chỗ trống và tính phí, đạt được độ trễ trung bình **1.12** giây và độ chính xác **97.2%**, đáp ứng yêu cầu vận hành của một bãi đỗ xe thông minh.

Trong tương lai, công việc sẽ tập trung vào:

- **Thanh toán điện tử:** Tích hợp các cổng thanh toán trực tuyến (VNPay, Momo) vào ứng dụng di động.
- **Phân tích dự đoán:** Ứng dụng thuật toán Machine Learning để dự đoán nhu cầu đỗ xe, hỗ trợ quản trị viên đưa ra quyết định tối ưu.
- **Tối ưu hóa năng lượng:** Nghiên cứu các chế độ hoạt động tiết kiệm năng lượng cho các module ESP32 để kéo dài tuổi thọ pin trong trường hợp triển khai không dây.

## LỜI CẢM ƠN

Chúng tôi xin gửi lời tri ân sâu sắc nhất tới các Giảng viên hướng dẫn: ThS. Nguyễn Thái Khánh và ThS. Lê Trung Hiếu.

Trong suốt quá trình thực hiện đề tài, Thầy đã dành nhiều thời gian quý báu để truyền đạt kiến thức chuyên môn sâu sắc, đưa ra những định hướng khoa học kịp thời, cùng với sự nhiệt tình sửa chữa, góp ý chi tiết về mặt kỹ thuật lẫn hình thức trình bày. Sự tận tâm, trách nhiệm và kinh nghiệm thực tiễn của Thầy không chỉ giúp chúng tôi vượt qua những khó khăn về mặt kỹ thuật phức tạp của dự án IoT, mà còn khơi gợi niềm đam mê nghiên cứu khoa học và phát triển kỹ năng tư duy phản biện. Đây là những bài học vô giá mà chúng tôi sẽ mang theo trong suốt sự nghiệp.

## REFERENCES

- [1] K. Ashton, “That ‘Internet of Things’ Thing,” *RFID Journal*, 2009.
- [2] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, “Internet of Things for Smart Cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [3] R. G. Hollands, “Will the real smart city please stand up?,” *City*, vol. 12, no. 3, pp. 303–320, 2008.
- [4] P. S. S. Rani, T. Subbiah and R. J. R., “Smart Parking System using ESP32 with Ultrasonic Sensor and Cloud Storage,” *Journal of Physics: Conference Series*, vol. 2089, no. 1, 2021.