

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH

KHOA ĐIỆN - ĐIỆN TỬ

BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

TRƯƠNG DUY KHANG

NGUYỄN NHỰT THÀNH

LUẬN VĂN TỐT NGHIỆP

**THIẾT KẾ HỆ THỐNG HOẠCH ĐỊNH QUÝ ĐẠO
VÀ GIÁM SÁT HOẠT ĐỘNG CHO XE ĐIỆN TỰ HÀNH
QUA MẠNG DI ĐỘNG 4G**

KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

TP. HỒ CHÍ MINH, 2022

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH

KHOA ĐIỆN - ĐIỆN TỬ

BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

TRƯƠNG DUY KHANG - 1810985

NGUYỄN NHỰT THÀNH - 1811223

LUẬN VĂN TỐT NGHIỆP

**THIẾT KẾ HỆ THỐNG HOẠCH ĐỊNH QUÝ ĐẠO
VÀ GIÁM SÁT HOẠT ĐỘNG CHO XE ĐIỆN TỰ HÀNH
QUA MẠNG DI ĐỘNG 4G**

**(DESIGN A PATH PLANNING AND OPERATION
MONITORING SYSTEM FOR AUTONOMOUS ELECTRIC
VEHICLE VIA 4G NETWORK)**

KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

GIẢNG VIÊN HƯỚNG DẪN

TS. NGUYỄN HOÀNG GIÁP

TP. HỒ CHÍ MINH, 2022

TP. HCM, ngày....tháng.....năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP
CỦA CÁN BỘ HƯỚNG DẪN**

Tên luận văn:

**THIẾT KẾ HỆ THỐNG HOẠCH ĐỊNH QUÝ ĐẠO VÀ GIÁM SÁT HOẠT ĐỘNG CHO XE
ĐIỆN TỰ HÀNH QUA MẠNG DI ĐỘNG 4G/ DESIGN A PATH PLANNING AND
OPERATION MONITORING SYSTEM FOR AUTONOMOUS ELECTRIC VEHICLE VIA
4G NETWORK**

Nhóm Sinh viên thực hiện:

Trương Duy Khang

1810985 TS. Nguyễn Hoàng Giáp

Nguyễn Nhựt Thành

1811223 TS. Nguyễn Hoàng Giáp

Cán bộ hướng dẫn:

Đánh giá Luận văn

1. Về cuốn báo cáo:

Số trang _____ Số chương _____
Số bảng số liệu _____ Số hình vẽ _____
Số tài liệu tham khảo _____ Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

.....
.....
.....
.....
.....

2. Về nội dung luận văn:

.....
.....
.....
.....
.....

3. Về tính ứng dụng:

.....

.....

.....

.....

.....

.....

4. Về thái độ làm việc của sinh viên:

.....

.....

.....

.....

.....

.....

Đánh giá chung: Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình

Điểm từng sinh viên:

Trương Duy Khang: /10

Nguyễn Nhựt Thành: /10

Cán bộ hướng dẫn

(Ký tên và ghi rõ họ tên)

TP. HCM, ngày....tháng.....năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP
CỦA CÁN BỘ PHẢN BIỆN**

Tên luận văn:

**THIẾT KẾ HỆ THỐNG HOẠCH ĐỊNH QUÝ ĐẠO VÀ GIÁM SÁT HOẠT ĐỘNG CHO XE
ĐIỆN TỰ HÀNH QUA MẠNG DI ĐỘNG 4G/ DESIGN A PATH PLANNING AND
OPERATION MONITORING SYSTEM FOR AUTONOMOUS ELECTRIC VEHICLE VIA
4G NETWORK**

Nhóm Sinh viên thực hiện:

Trương Duy Khang 1810985

Nguyễn Nhựt Thành 1811223

Cán bộ phản biện:

Đánh giá Luận văn

5. Về cuốn báo cáo:

Số trang _____ Số chương _____

Số bảng số liệu _____ Số hình vẽ _____

Số tài liệu tham khảo _____ Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

.....
.....
.....
.....
.....
.....

6. Về nội dung luận văn:

.....
.....
.....
.....
.....
.....

7. Về tính ứng dụng:

.....

.....

.....

.....

.....

.....

8. Về thái độ làm việc của sinh viên:

.....

.....

.....

.....

.....

.....

Đánh giá chung: Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình

Điểm từng sinh viên:

Trương Duy Khang: /10

Nguyễn Nhựt Thành: /10

Người nhận xét

(Ký tên và ghi rõ họ tên)

TP. HCM, ngày....tháng.....năm.....

ĐỀ CƯƠNG CHI TIẾT

TÊN LUẬN VĂN: THIẾT KẾ HỆ THỐNG HOẠCH ĐỊNH QUỸ ĐẠO VÀ GIÁM SÁT HOẠT ĐỘNG CHO XE ĐIỆN TỰ HÀNH QUA MẠNG DI ĐỘNG 4G

Cán bộ hướng dẫn: TS. Nguyễn Hoàng Giáp

Thời gian thực hiện: Từ ngày 03/01/2022 đến ngày 20/05/2022

Sinh viên thực hiện:

Trương Duy Khang - 1810985

Nguyễn Nhựt Thành - 1811223

Nội dung đề tài:

- *Mục tiêu:* xây dựng hệ thống hoạch định quỹ đạo di chuyển tối ưu cho xe điện tự hành, hệ thống giám sát hoạt động của xe (tốc độ, vị trí...) và vận hành từ xa hoạt động của xe (điều xe, giao nhiệm vụ cho các xe...) qua mạng di động 4G.
- *Phạm vi:* ứng dụng trong phạm vi cục bộ (khu du lịch, sân golf...)
- *Đối tượng nghiên cứu:* phương tiện xe điện, hệ thống phần cứng và điều khiển xe, cảm biến GPS, mô-đun 4G.
- *Phương pháp thực hiện:* tìm hiểu và phân tích lý thuyết dựa vào các bài nghiên cứu được công bố, tiến hành thực nghiệm đánh giá và so sánh kết quả.
- *Kết quả mong đợi:* xây dựng được hệ thống giám sát và hoạch định đường đi tối ưu cho xe tự hành; đảm bảo quá trình truyền nhận dữ liệu từ xa qua mạng 4G giữa các xe và trung tâm vận hành được ổn định; phát triển các giao diện người dùng thực hiện giao tiếp với xe, giám sát và vận hành hoạt động của xe từ xa, các giao diện, ứng dụng cho khách hàng là người đặt xe.

Kế hoạch thực hiện:

Thời gian	Công việc	Người thực hiện
Tháng 01/2022	Tìm hiểu tổng quan về đề tài, yêu cầu và nhiệm vụ đặt ra.	Trương Duy Khang Nguyễn Nhựt Thành
Tháng 01/2022	Tìm hiểu giải thuật Path Planning và các thuật toán tìm đường đi ngắn nhất (Astar, Dijkstra...) thông qua các bài báo khoa học. Nghiên cứu và xây dựng thuật toán phù hợp và tối ưu áp dụng riêng trên xe tự lái.	Trương Duy Khang
Tháng 01/2022	Tìm hiểu các mô hình IoT, Webserver thông qua các bài báo khoa học. Lựa chọn mô hình phù hợp.	Nguyễn Nhựt Thành
Tháng 02/2022	Lựa chọn mô-đun GPS phục vụ cho giải thuật Path Planning. Thiết kế và xây dựng tủ điện điều khiển (nguồn, vi xử lý).	Trương Duy Khang
Tháng 02/2022	Xử lý dữ liệu GPS để xác định vị trí phương tiện.	Trương Duy Khang
Tháng 02/2022	Xây dựng server cơ bản, cơ sở dữ liệu để bắt đầu thực hiện việc truyền nhận dữ liệu giữa server và website, giữa server và phương tiện.	Nguyễn Nhựt Thành
Tháng 03/2022	Triển khai thực hiện thuật toán tìm đường đi ngắn nhất.	Trương Duy Khang
Tháng 03/2022	Thực hiện giải thuật Path Planning.	Trương Duy Khang
Tháng 03/2022	Thiết kế GUI giúp người dùng tương tác với phương tiện.	Trương Duy Khang

Tháng 03/2022	Lựa chọn mô-đun 4G phù hợp, khảo sát và thực hiện truyền nhận dữ liệu từ vi điều khiển lên server thông qua mô-đun.	Nguyễn Nhựt Thành
Tháng 03/2022	Thiết kế giao diện web cơ bản cho người đặt xe, và cho trung tâm vận hành đội xe.	Nguyễn Nhựt Thành
Tháng 04/2022	Thực hiện truyền nhận dữ liệu từ xa thông qua mạng 4G	Trương Duy Khang Nguyễn Nhựt Thành
Tháng 04/2022	Kết nối các phần mềm trong hệ thống lại với nhau, kết nối dữ liệu trên xe với dữ liệu trên server	Trương Duy Khang Nguyễn Nhựt Thành
Tháng 04/2022	Thiết kế hộp điện kết nối các mô-đun GPS, mô-đun 4G, và vi điều khiển	Nguyễn Nhựt Thành
Tháng 04/2022	Thực nghiệm lý thuyết trên phương tiện thực tế.	Trương Duy Khang Nguyễn Nhựt Thành
Tháng 05/2022	Đánh giá, hiệu chỉnh đạt kết quả mong muốn và viết báo cáo.	Trương Duy Khang Nguyễn Nhựt Thành

Xác nhận của Cán bộ hướng dẫn (Ký tên và ghi rõ họ tên)	TP. HCM, ngàythángnăm.....
	Sinh viên (Ký tên và ghi rõ họ tên)

Trương Duy Khang Nguyễn Nhựt Thành

DANH SÁCH HỘI ĐỒNG BẢO VỆ LUẬN VĂN

Hội đồng chấm luận văn tốt nghiệp, thành lập theo Quyết định số
ngày của Hiệu trưởng Trường Đại học Bách khoa TP.HCM.

1. – Chủ tịch.
2. – Thư ký.
3. – Ủy viên.
4. – Ủy viên.
5. – Ủy viên.

LỜI CẢM ƠN

Đầu tiên, chúng em xin chân thành gửi lời cảm ơn đến các thầy cô ở trường đại học Bách khoa nói chung và các thầy cô thuộc khoa Điện – Điện tử nói riêng – là những người đã giảng dạy, cho chúng em kiến thức trong những năm đại học đã qua. Những kiến thức đó là yếu tố hết sức quan trọng để chúng em có thể hoàn thành được luận văn này.

Chúng em cũng đặc biệt cảm ơn thầy Nguyễn Hoàng Giáp - là người trực tiếp hướng dẫn chúng em trong suốt quá trình làm luận văn. Nhờ sự định hướng, kiểm tra, nhắc nhở, và những lời khuyên của thầy mà chúng em có thể thực hiện được luận văn này. Chúng em vô cùng biết ơn những chỉ dẫn của thầy trong suốt thời gian làm luận văn.

Chúng em cảm ơn công ty VAS đã hỗ trợ cho chúng em rất nhiều về trang thiết bị phục vụ cho luận văn của mình. Những hỗ trợ đó giúp cho chúng em rất nhiều trong việc nâng cao chất lượng luận văn.

Cảm ơn các thành viên nhóm AEV (Autonomous Electric Vehicle) là những người đã cùng đồng hành với chúng em, trực tiếp thực hiện các quá trình tự hành cho xe, cũng như hỗ trợ chúng em trong việc thực hiện đề tài.

TP. Hồ Chí Minh, ngày 20 tháng 5 năm 2022

Trương Duy Khang

Nguyễn Nhựt Thành

MỤC LỤC

Chương 1. GIỚI THIỆU ĐỀ TÀI.....	2
1.1. Lý do chọn đề tài.....	2
1.1.1. Tổng quan về xe tự hành	2
1.1.2. Sự cần thiết của đề tài	3
1.2. Nhiệm vụ của đề tài	4
1.2.1. Ý nghĩa khoa học.....	4
1.2.2. Phạm vi ứng dụng	4
1.2.3. Các bước thực hiện luận văn.....	5
1.3. Các nghiên cứu liên quan trong và ngoài nước.....	6
1.3.1. Tình hình ngoài nước.....	7
1.3.2. Tình hình trong nước	8
Chương 2. CƠ SỞ LÝ THUYẾT	11
2.1. Thuật toán hoạch định quỹ đạo cho xe tự hành.....	11
2.1.1. Bài toán quỹ đạo tối ưu	11
2.1.2. Thuật toán A *	12
2.2. Phương pháp số hóa bản đồ cục bộ.....	13
2.3. Phương pháp hồi quy tuyến tính ứng dụng cho xác định hướng di chuyển của xe.....	14
2.4. Phương pháp phát hiện làn đường.....	15
2.5. Bộ điều khiển.....	16
2.5.1. Bộ điều khiển Proportional Integral Derivative (PID)	16
2.5.2. Bộ điều khiển Model Predictive Control (MPC)	17

2.6. Mô hình webserver giám sát hoạt động của xe điện tự hành theo giao thức HTTP	19
2.6.1. Giao thức HTTP (HyperText Transfer Protocol) trong IoTs.....	19
2.6.2. Mô hình webserver giám sát hoạt động của xe điện tự hành theo giao thức HTTP	22
Chương 3. THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN CHO XE ĐIỆN TỰ HÀNH	24
3.1. Tổng quan hệ thống điều khiển trên xe điện	24
3.2. Tổng quan hệ thống điều khiển trên xe điện tự hành	25
3.3. Tổng quan hệ thống giám sát hoạt động của xe tự hành qua mạng 4G..	27
Chương 4. THIẾT KẾ HỆ THỐNG PHẦN CỨNG.....	29
4.1. Sơ đồ kết nối phần cứng hệ thống truyền nhận dữ liệu qua mạng 4G ...	29
4.2. Tổng quan các thiết bị	29
4.2.1. Vi xử lý STM32F407.....	29
4.2.2. Thiết bị GPS.....	31
4.2.3. Module 4G	33
Chương 5. THIẾT KẾ HỆ THỐNG PHẦN MỀM HOẠCH ĐỊNH QUÝ ĐẠO VÀ GIAO DIỆN ĐIỀU KHIỂN CHO XE TỰ HÀNH	35
5.1. Tổng quan kết nối hệ thống phần mềm.....	35
5.2. Xử lý dữ liệu từ cảm biến GPS.....	36
5.3. Xây dựng và hiển thị bản đồ số	38
5.3.1. Xây dựng bản đồ số khu vực cục bộ.....	38
5.3.2. Hiển thị bản đồ	40
5.4. Thực hiện thuật toán A *	40
5.5. Phương pháp hồi quy tuyến tính.....	44

5.6. Thiết kế giao diện điều khiển	47
5.6.1. Tổng quan giao diện điều khiển	47
5.6.2. Thiết kế các tính năng điều khiển xe	48
5.7. Ứng dụng hệ thống lên xe điện tự hành	49

Chương 6. THIẾT KẾ HỆ THỐNG WEBSERVER GIÁM SÁT VÀ VẬN HÀNH HOẠT ĐỘNG XE ĐIỆN TỰ HÀNH.....51

6.1. Tổng quan về hệ thống giám sát và vận hành hoạt động xe tự hành	51
6.2. Xây dựng Server	53
6.3. Thiết kế cơ sở dữ liệu	55
6.3.1. Tổng quan cơ sở dữ liệu.....	55
6.3.2. Cấu trúc cơ sở dữ liệu	56
6.4. Thiết kế giao diện web	58
6.4.1. Thiết kế giao diện giám sát cho người vận hành	59
6.4.2. Thiết kế giao diện đặt xe cho khách hàng	60
6.5. Lưu đồ giải thuật của hệ thống các luồng dữ liệu	60
6.6. Kết nối dữ liệu giữa xe và server qua mô-đun 4G.....64	64
6.6.1. Khảo sát mô-đun 4G SIMA7670C.....	64
6.6.2. Lưu đồ giải thuật thực hiện POST, GET request từ vi điều khiển đến server qua mô-đun 4G.....66	66
6.7. Ứng dụng thực tế của hệ thống vào xe tự hành	67

Chương 7. KẾT QUẢ THỰC HIỆN.....68

7.1. Kết quả thực hiện phần cứng	68
7.2. Kết quả thiết kế giao diện điều khiển cho xe điện tự hành.....71	71
7.3. Kết quả thực hệ thống phần mềm hoạch định quỹ đạo.....72	72

7.3.1. Kết quả xử lý dữ liệu cảm biến GPS	72
7.3.2. Kết quả thực hiện số hóa bản đồ	76
7.3.2.1. Công cụ thực hiện số hóa	76
7.3.2.2. Đồng bộ tọa độ thực tế và bản đồ số hóa.....	81
7.3.2.3. Khảo sát độ chính xác của bản đồ xây dựng.....	81
7.3.3. Kết quả thực hiện thuật toán A *	83
7.3.3.1. Kiểm tra sơ bộ kết quả.....	83
7.3.3.2. Kết quả tìm đường trên bản đồ số.....	86
7.3.4. Kết quả điều hướng kết hợp phương pháp hồi quy tuyến tính và quỹ đạo A *	89
7.4. Kết quả thực hiện server và cơ sở dữ liệu.....	91
7.4.1. Kết quả xây dựng server	91
7.4.2. Kết quả xây dựng cơ sở dữ liệu	93
7.5. Kết quả thực hiện server kết nối dữ liệu.....	94
7.5.1. Thực nghiệm kiểm tra tốc độ truyền nhận dữ liệu	94
7.5.2. Thực hiện gửi dữ liệu hoạt động của xe lên cơ sở dữ liệu qua mô-đun 4G	95
7.5.3. Thực hiện giám sát lộ trình di chuyển của xe từ GUI giám sát trung tâm	99
7.6. Kết quả thực hiện giao diện web cho người sử dụng.....	102
7.6.1. Giao diện giám sát cho người vận hành	102
7.6.2. Giao diện web cho khách hàng đặt xe	103
7.7. Kết quả ứng dụng lên xe điện tự hành	103
Chương 8. ĐÁNH GIÁ, KẾT LUẬN, HƯỚNG PHÁT TRIỂN ĐỀ TÀI.....	111

8.1. Đánh giá kết quả	111
8.2. Kết luận đề tài	112
8.3. Hướng phát triển đề tài	113

DANH MỤC HÌNH VẼ

Hình 1-1 Xe điện giao hàng Nuro R2	7
Hình 1-2 Xe tự hành cấp độ 4 của Đại học Phenikaa (Hà Nội)	8
Hình 1-3 Giao diện phần mềm TRACKING.VN	9
Hình 2-1 Ánh xạ bản đồ.....	14
Hình 2-2 Lưu đồ giải thuật sử dụng xử lý ảnh để phát hiện làn đường.....	15
Hình 2-3 Sơ đồ bộ điều khiển PID	17
Hình 2-4 Ý tưởng điều khiển dùng Model Predictive Control	19
Hình 2-5 Kiến trúc tổng quan của Model Predictive Control	19
Hình 2-6 Dạng yêu cầu/phản hồi của HTTP	20
Hình 2-7 Ví dụ URL cho 1 GET request trong đề tài	21
Hình 2-8 Mô hình webserver theo giao thức HTTP.....	22
Hình 3-1 Hình ảnh thực tế xe điện 4 chỗ Sanyo	24
Hình 3-2 Tổng quan phần điều khiển trên xe	24
Hình 3-3 Tổng quan hệ thống điều khiển trên xe điện tự hành.....	25
Hình 3-4 Hệ thống các cảm biến được tích hợp vào xe điện hỗ trợ quá trình tự hành	26
Hình 3-5 Hình ảnh thực tế của các cảm biến, và thiết bị tích hợp vào xe điện tự hành	27
Hình 3-6 Mô hình hệ thống IoT cho xe tự hành qua giao thức HTTP	28
Hình 4-1 Sơ đồ tổng quan hệ thống phần cứng truyền nhận dữ liệu qua mạng 4G	29
Hình 4-2 Board STM32F407 Discovery	29
Hình 4-3 Mô – đun GPS U-Blox NEO-7N	32
Hình 4-4 Module 4G/LTE SIM A7670C	33
Hình 5-1 Tổng quan hệ thống hoạch định quỹ đạo và gửi dữ liệu lên server....	35
Hình 5-2 Robot Operating System	36
Hình 5-3 Cấu trúc một gói tin giao thức NMEA [10]	37

Hình 5-4 Cấu trúc gói tin GPGLL của giao thức NMEA.....	37
Hình 5-5 Quy trình xử lý dữ liệu nhận được từ GPS	38
Hình 5-6 Quy trình thực hiện số hóa khu vực cục bộ	39
Hình 5-7 Quá trình tìm kiếm của thuật toán A * [11]	41
Hình 5-8 Sơ đồ giải thuật điều hướng theo quỹ đạo đã hoạch định.....	44
Hình 5-9 Góc tạo bởi vector chuyển động và vector điều hướng $\alpha < 0$	46
Hình 5-10 Góc tạo bởi vector chuyển động và vector điều hướng $\alpha > 0$	46
Hình 5-11 Tổng quan giao diện người dùng.....	48
Hình 5-12 Ứng dụng hoạch định quỹ đạo và phương pháp điều hướng lên xe tự hành	49
Hình 6-1 Mô hình hệ thống giám sát và vận hành hoạt động của xe (đội xe) tự hành	51
Hình 6-2 Dữ liệu từ xe gửi lên server và trung tâm giám sát lấy dữ liệu về.....	52
Hình 6-3 Dữ liệu điều khiển gửi đến xe	52
Hình 6-4 Dữ liệu đặt xe với điểm đến cuối cùng là yêu cầu xe thực hiện lộ trình được đặt	52
Hình 6-5 Mô hình kết nối hệ thống dữ liệu qua mạng.....	54
Hình 6-6 Cách thức trao đổi dữ liệu	54
Hình 6-7 Cách thức truy cập MySQL	55
Hình 6-8 Server đóng vai trò là client khi truy cập vào database	56
Hình 6-9 Cách thức hoạt động của PHP Framework	59
Hình 6-10 Giao diện giám sát từ xa thông qua giao thức HTTP	59
Hình 6-11 Giao diện đặt xe cho khách hàng.....	60
Hình 6-12 Sơ đồ giải thuật khi Client là người dùng đặt xe	61
Hình 6-13 Sơ đồ giải thuật khi Client là người vận hành yêu cầu dữ liệu của đội xe	61
Hình 6-14 Giải thuật chọn xe tối ưu thực hiện nhiệm vụ.....	62
Hình 6-15 Sơ đồ giải thuật khi Client là xe (vi điều khiển) yêu cầu thông tin đặt xe từ server qua mô-đun 4G	63

Hình 6-16 Sơ đồ giải thuật khi Client là xe (vi điều khiển) gửi dữ liệu về server qua mô-đun 4G.....	63
Hình 6-17 Lưu đồ giải thuật thực hiện POST, GET request từ vi điều khiển qua mô-đun 4G.....	66
Hình 6-18 Lưu đồ giải thuật ứng dụng thực tế của hệ thống vào xe tự hành....	67
Hình 7-1 Mạch lớp trên của mô-đun GPS-4G	68
Hình 7-2 Mạch lớp dưới của mô-đun GPS-4G	69
Hình 7-3 Hộp điện GPS-4G hoàn chỉnh	70
Hình 7-4 Giao diện chính để điều khiển xe.....	71
Hình 7-5 Thực hiện kết nối TCP với trung tâm điều khiển.....	72
Hình 7-6 Địa điểm khảo sát GPS.....	73
Hình 7-7 Dữ liệu GPS.....	74
Hình 7-8 Công cụ hỗ trợ xây dựng bản đồ số	77
Hình 7-9 Bản đồ thu thập được từ Google Maps	77
Hình 7-10 Bản đồ sau khi qua công cụ số hóa.....	78
Hình 7-11 Lớp bản đồ thu được sau khi thực hiện số hóa qua các bước trên ...	80
Hình 7-12 Chọn 5 điểm khảo sát bản đồ số.....	82
Hình 7-13 Tọa độ 5 điểm của bản đồ số được vẽ lên Google Maps.....	83
Hình 7-14 Đường đi A* thẳng xuống.....	84
Hình 7-15 Đường đi A * thẳng ngang.....	84
Hình 7-16 Đường đi A * đường chéo.....	85
Hình 7-17 Đường đi A * đường chéo.....	85
Hình 7-18 Đường đi A * có tránh vật cản	86
Hình 7-19 Quỹ đạo A * với đường đi thẳng.....	87
Hình 7-20 Quỹ đạo A * với đường cong ngắn.....	87
Hình 7-21 Quỹ đạo A * với đoạn đường dài	88
Hình 7-22 Quỹ đạo A * với đoạn đường phức tạp hơn.....	88
Hình 7-23 Chọn địa điểm di chuyển trên giao diện điều khiển.....	90
Hình 7-24 Tín hiệu điều hướng	90

Hình 7-25 Trình duyệt gửi yêu cầu đến server qua URL	91
Hình 7-26 Kết quả trả về từ server là trang web có tên Fleet Vehicles	92
Hình 7-27 Thực hiện GET dữ liệu từ server và nhận được phản hồi	92
Hình 7-28 Đoạn đường di chuyển khi khảo sát.....	95
Hình 7-29 Kết quả dựng lại lộ trình di chuyển bằng công cụ Google MyMaps với dữ liệu từ database – bảng gps_data_list.....	96
Hình 7-30 Lộ trình di chuyển trên Google Maps.....	97
Hình 7-31 Kết quả dựng lại đoạn đường di chuyển với Google MyMaps	98
Hình 7-32 Cung đường thực hiện quá trình Post Get song song	99
Hình 7-33 Kết quả dựng lại đoạn đường di chuyển với Google MyMaps	100
Hình 7-34 Giao diện giám sát trung tâm	101
Hình 7-35 Giao diện giám sát ở trung tâm vận hành	102
Hình 7-36 Giao diện đặt xe cho khách hàng.....	103
Hình 7-37 Kết nối giao diện với hệ thống điều khiển	104
Hình 7-38 Hiển thị vị trí hiện tại của xe	104
Hình 7-39 Điều hướng cho xe chạy theo quỹ đạo A *	106
Hình 7-40 Tín hiệu điều hướng của quá trình di chuyển.....	106
Hình 7-41 Đồ thị thay đổi vận tốc trong quá trình di chuyển	107
Hình 7-42 Chạy theo quỹ đạo được hoạch định từ thuật toán A* dựa trên dữ liệu từ server.....	108
Hình 7-43 Hệ thống Get dữ liệu đặt xe và di chuyển hết quỹ đạo	109
Hình 7-44 Tín hiệu điều hướng của quá trình di chuyển.....	109
Hình 7-45 Đồ thị vận tốc trong quá trình di chuyển	110

DANH MỤC BẢNG

Bảng 4-1 Thông số kỹ thuật vi xử lý STM32F407.....	31
Bảng 4-2 Một số thông số quan trọng của thiết bị GPS NEO 7N	32
Bảng 4-3 Một số thông số quan trọng của Module 4G/LTE SIM A7670C.....	34
Bảng 6-1 Cấu trúc cơ sở dữ liệu tổng.....	56
Bảng 6-2 Bảng fleet_info lưu các thông số hoạt động của xe.....	56
Bảng 6-3 Bảng gps_data_info lưu dữ liệu vị trí của xe.....	57
Bảng 6-4 Bảng booked_info lưu thông tin đặt xe.....	57
Bảng 6-5 Bảng task_control lưu thông tin điều xe	57
Bảng 7-1 Tính toán kích thước sân dựa trên dữ liệu trả về từ GPS.....	74
Bảng 7-2 Sai số tuyệt đối của các lần đo	75
Bảng 7-3 Bảng khảo sát độ chính xác bản đồ số	82
Bảng 7-4 Bảng gps_data_list lưu dữ liệu GPS	93
Bảng 7-5 Bảng fleet_info lưu thông tin đầy đủ hoạt động của đội xe	93
Bảng 7-6 Bảng booked_info lưu thông tin của khách hàng.....	94
Bảng 7-7 Bảng gps_data_list lưu dữ liệu POST gói tin mặc định liên tục	94
Bảng 7-8 Bảng dữ liệu lưu vị trí của xe trong khảo sát 1	96
Bảng 7-9 Bảng dữ liệu lưu vị trí của xe trong khảo sát 2	98
Bảng 7-10 Bảng dữ liệu lưu vị trí của xe trong khảo sát	100

DANH MỤC TỪ VIẾT TẮT

Từ khóa	Ý nghĩa
AEV	Autonomous Electric Vehicle
CoAP	Constrained Application Protocol
GPS	Global Positioning System
IOT	Internet of Things
HTTP	HyperText Transfer Protocol
LTE	Long Term Evolution
MPC	Model Predictive Controller
MQTT	Message Queuing Telemetry Transport
ROS	Robot Operating System
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver Transmitter
JSON	JavaScript Object Notation
M2M	Machine to machine
TTGSVH	Trung tâm giám sát và vận hành
CSDL	Cơ sở dữ liệu (database)
PID	Proportional Integral Derivative
MPC	Model Predictive Control

TÓM TẮT LUẬN VĂN

Đề tài thực hiện việc giám sát hoạt động và thực hiện các yêu cầu điều khiển hoạt động của một xe (đội xe) tự hành trên nền tảng web sử dụng giao thức HTTP (Hyper Text Transfer Protocol). Mục tiêu của đề tài là cho phép người quản lý có thể theo dõi từ xa các thông tin của xe (đội xe) tự hành như vị trí hay trạng thái hoạt động... từ đó có thể đưa ra các yêu cầu cụ thể về lộ trình, nhiệm vụ... cho từng xe thực hiện (điều khiển hoạt động). Các kết nối của xe đến internet được thực hiện qua mạng 4G/LTE.

Hướng đến mục tiêu hoạt động ở khu vực cục bộ (khu dân cư, khu du lịch, sân golf...) việc hoạch định chuyển động theo quỹ đạo đặt của người dùng không thể sử dụng API của Google Maps để điều hướng được. Vì vậy, phương pháp thực hiện để số hóa bản đồ của khu vực cục bộ sẽ được đề xuất trong đề tài này. Thuật toán tìm quỹ đạo chuyển động tối ưu hay giải thuật hỗ trợ điều hướng chuyển động cũng sẽ được triển khai. Đồng thời, thực hiện thiết kế một giao diện để người dùng có thể trực tiếp tương tác với phương tiện trong quá trình di chuyển khi cần thiết.

Mỗi xe tự hành bên cạnh các phần cứng có sẵn để có thể tự hoạt động sẽ được trang bị thêm các mô-đun GPS để xác định vị trí và hỗ trợ quá trình điều hướng, mô-đun 4G/LTE để kết nối dữ liệu của xe với server, và một vi điều khiển đảm nhận chức năng trung gian xử lý và giao tiếp dữ liệu giữa các mô-đun đó (gọi chung là mô-đun GPS-4G).

Các nhiệm vụ trọng tâm của đề tài là:

- Thiết kế phần cứng cho việc giao tiếp dữ liệu giữa xe và mô-đun GPS-4G.
- Xây dựng giao diện điều khiển xe; ứng dụng các thuật toán để hoạch định đường đi tối ưu cho mỗi xe thực hiện các nhiệm vụ được hiệu quả nhất.
- Xây dựng trang web giám sát và điều khiển hoạt động của xe (đội xe) cho người vận hành, trang web thực hiện các thao tác đặt xe cho khách hàng.

Chương 1. GIỚI THIỆU ĐỀ TÀI

1.1. Lý do chọn đề tài

1.1.1. Tổng quan về xe tự hành

Xe tự lái hay còn gọi là xe tự hành (self-driving car) là phương tiện có khả năng tự di chuyển an toàn với ít hoặc không cần sự hỗ trợ từ con người. Với sự hỗ trợ của hệ thống các cảm biến, camera, radar, GPS... cùng các thuật toán phát hiện mà xe có thể cảm nhận được môi trường xung quanh từ đó áp dụng các giải thuật điều khiển để đưa ra thao tác phù hợp giữ cho xe đi đúng và bảo đảm an toàn. Theo sự phát triển của các công nghệ cho xe tự hành thì xe tự hành được chia theo 6 cấp độ tự hành từ cấp 0 (con người điều khiển hoàn toàn – không tự hành) cho đến cấp 5 (xe tự hành hoàn toàn).

Một xe tự hành khi đưa vào hoạt động phải đảm bảo một số yêu cầu sau:

- Hoạt động an toàn, không vi phạm luật giao thông, không gây nguy hiểm cho người ngồi trên xe và các phương tiện giao thông khác là yêu cầu bắt buộc.
- Có khả năng kiểm soát các trạng thái hoạt động của xe như tốc độ, đánh lái, phanh... và các tín hiệu cảnh báo cho các phương tiện xung quanh: xi nhan, còi.
- Có khả năng tự di chuyển đúng theo lộ trình đã vạch ra, giữ đúng làn đường, và tuân thủ quy định an toàn giao thông như một phương tiện giao thông bình thường.
- Có khả năng nhận diện, phát hiện các vật cản trên đường đi (có gây cản trở đối với xe) như người, xe máy... các tín hiệu, biển báo giao thông từ đó có giải pháp để giữ an toàn như đánh lái né vật cản, dừng lại hay cảnh báo yêu cầu sự điều khiển từ người lái xe...
- Có khả năng kiểm soát và phát hiện các lỗi hệ thống để cảnh báo và khắc phục kịp thời.

Công cuộc xe tự hành đã được phát triển rất mạnh mẽ, và cho đến ngày nay thì những kỹ thuật, công nghệ cho xe tự hành đã có thể đảm bảo rất tốt những yếu tố trên. Tuy nhiên, để đạt đến cấp độ cao nhất của xe tự hành là tự hành hoàn toàn trên đường phố mà không cần bất cứ sự can thiệp của con người thì vẫn còn cần thêm thời gian.

Cho nên, ở mức độ đại học, đề tài chúng em chọn là **THIẾT KẾ HỆ THỐNG HOẠCH ĐỊNH QUÝ ĐẠO VÀ GIÁM SÁT HOẠT ĐỘNG CHO XE ĐIỆN TỰ HÀNH QUA MẠNG DI ĐỘNG 4G**. Mức độ tự hành của các xe trong đề tài của chúng em là cấp độ 2 – xe có khả năng tự hành trong một khu vực nhất định như khu resort, sân golf và thực hiện một số nhiệm vụ chuyên biệt như đưa đón khách, vận chuyển hàng hóa, tự động sạc pin và có yêu cầu tài xế phải thường xuyên giám sát và thực hiện thao tác khi cần.

1.1.2. Sự cần thiết của đề tài

Hướng đến mục tiêu xe điện có thể hoạt động tại những khu vực cục bộ (các khu dân cư, khu du lịch, sân golf...), nơi mà bản đồ của Google không thể hiện những đặc trưng chi tiết của khu vực cho quá trình di chuyển thì ứng dụng của thuật toán tìm đường đi là vô cùng cần thiết. Tại những khu vực cục bộ này khi người dùng muốn cho xe di chuyển từ điểm bắt đầu đến điểm kết thúc, sẽ không thể sử dụng API của Google Maps để điều hướng vì Google Maps không có dữ liệu của những con đường trong khu vực này; khi đó thông qua một bản đồ số hóa do nhóm tự phát triển và thuật toán tìm đường đi sẽ hoạch định được một quỹ đạo chuyển động tối ưu giữa hai điểm đó giúp người đến được điểm đích bằng con đường tối ưu nhất có thể.

Hơn nữa, nhận thấy rằng trong thời buổi xu hướng IoTs đang thống trị giới công nghệ, nếu một phương tiện tự hành chỉ hoạt động đơn lẻ, không kết nối vào thế giới internet thì dự án đó sẽ không thể tiến xa được. Cho nên để phù hợp với những đòi hỏi thực tế của xã hội, việc kết nối dữ liệu, giám sát và điều khiển xe từ xa là vô cùng cần thiết. Từ đó tạo cơ sở để phát triển đội xe tự hành

hoạt động trong một khu vực nhất định, với các thao tác giao nhiệm vụ được thực hiện từ xa và được giám sát từ trung tâm vận hành để việc thực hiện các yêu cầu khách hàng được tối ưu nhất.

1.2. Nhiệm vụ của đề tài

1.2.1. Ý nghĩa khoa học

Sau khi cùng các thành viên trong dự án tự hành cho xe điện ở cấp độ đã có bước đầu đạt được cơ bản các mục tiêu, ý tưởng tiếp theo được ra là giám sát hoạt động của xe từ xa và về sau sẽ là phát triển đối với đội xe tự hành.

Ứng dụng định vị GPS đóng vai trò là dữ liệu nền tảng trong đề tài này, nó cho biết thông tin quan trọng nhất trong đề tài chính là vị trí hiện tại của các xe trong đội xe đang vận hành trong dự án. Sau đó, do xe tự hành ở cấp độ 2 tức là chỉ hoạt động trong một khu vực cố định nên việc số hóa bản đồ cũng cho thấy hiệu quả cao hơn do nó có thể hoạt động nội bộ mà không cần thêm kết nối internet nếu như khu vực đó đã được số hóa. Thuật toán lựa chọn đường đi ngắn nhất A* quyết định lộ trình di chuyển cho xe và cho thêm các chỉ dẫn hỗ trợ cho việc tự động lái của xe.

Ứng dụng IoT thông qua việc giám sát dữ liệu từ xa là việc áp dụng xu thế của thời đại, khi vật vạn đều được kết nối Internet. Tìm hiểu và ứng dụng các kiến thức về lập trình webserver để thiết lập hệ thống giao tiếp giữa đội xe tự hành, người quản lý đội xe và khách hàng.

1.2.2. Phạm vi ứng dụng

Đề tài của chúng em được thực hiện song song với việc tham gia hỗ trợ cả nhóm thực hiện tự hành cho xe điện ở cấp độ 2 cho nên các tính năng phát triển cho đề tài của chúng em cũng nhằm mục đích phục vụ cho xe. Mục tiêu trọng tâm đề tài chúng em hướng đến là việc giám sát hoạt động của xe và của đội xe khi số lượng xe tự hành trong khu vực đó tăng lên.

Với các mục tiêu đề ra khi hoàn thành luận văn là:

- Thực hiện kết nối và trao đổi thành công dữ liệu giữa xe với server. Vì đối tượng nghiên cứu là xe tự hành hoạt động trong một khu vực cố định nhưng có diện tích không giới hạn nên mạng 4G được lựa chọn là phương thức kết nối internet nhằm đảm bảo kết nối được ổn định thay vì mạng wifi.
- Thiết kế giao diện web cho người giám sát và vận hành (trung tâm điều khiển) và giao diện cho người dùng (khách hàng) thực hiện các nhiệm vụ đặt xe, hay giao hàng...
- Từ các dữ liệu về GPS và các thuật toán về hoạch định đường đi xác định được lộ trình tối ưu cho từng xe để nâng cao hiệu quả công việc và giảm thiểu chi phí khi đưa vào vận hành thực tế.
- Thực hiện số hóa bản đồ cho phép xe có thể thực hiện lộ trình mà không cần hệ thống internet cũng như có được những thông tin cần thiết khác tại những khu vực mà Google Maps hoạt động không thực sự hiệu quả. (Phần này độc lập với việc giám sát dữ liệu của đội xe).
- Thiết kế được sản phẩm để có thể dễ dàng sử dụng, cũng như dễ dàng tích hợp trong những dự án tương tự khác, đặc biệt là trong các dự án về quản lý đội xe tự hành (Fleet Management System).

Khi hoàn thành được những mục tiêu trên, thì việc áp dụng xe tự hành vào các khu vực nhất định như resort, các khu liên hợp thể thao... sẽ dần thay thế các phương tiện có người lái truyền thống, từ đó nâng cao hiệu quả công việc, giảm thiểu chi phí nhân công, và làm tăng thêm sự hiện đại cho những khu vực cũng hướng đến sự tiện nghi và hiện đại đó.

1.2.3. Các bước thực hiện luận văn

- Tìm hiểu tổng quan về đề tài, yêu cầu và nhiệm vụ đặt ra.

- Tìm hiểu giải thuật Path Planning và các thuật toán tìm đường đi ngắn nhất (Astar, Dijkstra...) thông qua các bài báo khoa học. Nghiên cứu và xây dựng thuật toán phù hợp và tối ưu áp dụng riêng trên xe tự lái.
- Tìm hiểu các mô hình IoT, Webserver thông qua các bài báo khoa học. Lựa chọn mô hình phù hợp.
- Xây dựng server cơ bản, cơ sở dữ liệu để bắt đầu thực hiện việc truyền nhận dữ liệu giữa server và website, giữa server và phương tiện.
- Lựa chọn mô-đun GPS phục vụ cho giải thuật Path Planning, xử lý dữ liệu GPS để xác định vị trí phương tiện.
- Lựa chọn mô-đun 4G/LTE phục vụ cho việc truyền nhận dữ liệu qua internet. Thực hiện truyền nhận dữ liệu giữa mô-đun 4G và vi điều khiển.
- Thiết kế và xây dựng hộp điện điều khiển (nguồn, vi xử lý, mô-đun GPS, mô-đun 4G – gọi chung là mô-đun GPS-4G).
- Triển khai thực hiện thuật toán tìm đường đi ngắn nhất. Thực hiện giải thuật Path Planning.
- Thiết kế GUI điều khiển trên xe giúp người dùng tương tác với phương tiện.
- Thiết kế giao diện trang web ở trung tâm điều khiển và cho người dùng.
- Hoàn thiện server để xử lý các tác vụ trong quá trình xe hoạt động.
- Thực nghiệm lý thuyết trên phương tiện thực tế.
- Đánh giá, hiệu chỉnh đạt kết quả mong muốn và viết báo cáo.

1.3. Các nghiên cứu liên quan trong và ngoài nước

Có rất nhiều nghiên cứu thực hiện giải quyết bài toán hoạch định quỹ đạo chuyển động tối ưu cho phương tiện tự hành, đặc biệt đối với khu vực cục bộ thì bài toán này trở nên vô cùng quan trọng. Hơn nữa, khi hoạt động trong một khu vực cục bộ, người ta thường có xu hướng liên kết mọi thiết bị thành một thể thống nhất để dễ dàng quản lý và điều phối. Ngày nay, khi IoT đã trở thành

xu thế của thời đại thì mọi lĩnh vực của cuộc sống đều được kết nối với nhau qua internet. Từ đó các ứng dụng IoT cho ngành giao thông nói chung, hay cho xe tự hành nói riêng cũng đã có rất nhiều những thành tựu nổi bật và hướng tới những tiện nghi nhiều hơn cho cuộc sống con người.

1.3.1. Tình hình ngoài nước

Hiện nay các thế hệ xe tự hành trên thế giới vẫn đang ở mức phát triển từ cấp độ 4 lên cấp độ 5. Và trong khi thực hiện quá trình đó, các hãng công nghệ đều đang tập trung hướng tới cải thiện trải nghiệm người dùng như trải nghiệm của người ngồi trên xe, hay mức độ đóng góp mà khả năng tự hành mang lại. Trong đó những ứng dụng quản lý từ xa đang được áp dụng rất rộng rãi trong nhiều lĩnh vực khác nhau.

❖ Xe điện giao hàng Nuro R2 của hãng Nuro [1]



Hình 1-1 Xe điện giao hàng Nuro R2

Nuro R2 được thiết kế không có vô lăng, gương chiếu hậu bên hay bàn đạp cho nên rất dễ nhận ra. Với hệ thống giám sát và yêu cầu từ xa Nuro có khả năng thực hiện các tác vụ giao hàng mà không cần có tài xế ngồi trên xe. Từ 2020, Nuro R2 đã được Cơ quan an toàn giao thông quốc gia Mỹ (NHTSA) cho

phép chạy thử nghiệm ở bang California để cung cấp dịch vụ giao đồ cho các cửa hàng tạp hóa, nhà hàng và một số doanh nghiệp khác.

1.3.2. Tình hình trong nước

Tại Việt Nam, ứng dụng IoT để giám sát cho các đội xe ô tô, hay các phương tiện vận tải tương đối phổ biến và phát triển, trong khi đó lĩnh vực IoT cho xe tự hành thì chưa thực sự nổi trội do đa phần các dự án xe tự hành ở Việt Nam vẫn còn trong giai đoạn thử nghiệm và chưa được đưa vào sử dụng rộng rãi. Một số dự án đang được phát triển ở Việt Nam:

❖ Xe tự hành cấp độ 4 của Đại học Phenikaa (Hà Nội)



Hình 1-2 Xe tự hành cấp độ 4 của Đại học Phenikaa (Hà Nội)

Xe do nhóm nghiên cứu thuộc Tập đoàn Phenikaa chế tạo, chạy bằng điện, đạt 100 km một ngày với vận tốc trung bình 20 km/h, pin sạc đầy sau 7 tiếng. Xe được trang bị các tính năng tự hành chuẩn cấp độ 4 theo Hiệp hội Kỹ sư Xe hơi (SAE), như bản đồ 3D, các cảm biến Lidar và GPS phân giải cao, công nghệ học máy, học sâu. Tùy vào địa hình, hệ thống có thể quét thành bản đồ 3D cho khu vực 10 ha chỉ trong 30 phút, khả năng hiệu chỉnh vị trí xe trên bản đồ với độ chính xác dưới 1 cm.

Hiện tại, Phenikaa Maas đang hỗ trợ Phenikaa-X trong việc xây dựng ứng dụng điều khiển xe tự hành trên điện thoại. Trong một thành phố thông minh, chúng

ta có thể dùng điện thoại đặt một chiếc xe bất kỳ, chọn điểm đến mà không cần phải quan tâm đến việc xe này là của ai. Hệ thống điều hành của xe sẽ có thể thiết lập lộ trình di chuyển và đặt vận tốc tối đa từ xa, chuyển chức năng lái giữa một cầu và hai cầu, cũng như tự động di chuyển vào khu vực đỗ xe. Đặc biệt, xe tự hoàn toàn không có tay lái nên không cần hệ thống trợ lái, chế độ tự hành được dễ dàng thực hiện khi người dùng có thể tương tác với xe thông qua phần mềm được thiết kế riêng biệt.

❖ Ứng dụng quản lý đội xe TRACKING.VN của DSS



Hình 1-3 Giao diện phần mềm TRACKING.VN

Với thiết bị định vị của DSS, tích hợp lên đến 8 mô-đun quản lý nâng cao về phương tiện và đội xe, giúp việc quản lý phương tiện trở nên dễ dàng và chính xác hơn bao giờ hết. Không chỉ dừng lại ở việc theo dõi Km và vị trí, vận tốc của phương tiện. Với 8 mô-đun quản lý: Quản lý xe ra vào trạm BOT, quản lý đổ rút nhiên liệu, quản lý đến hạn các loại giấy tờ, bảo trì bảo dưỡng; quản lý điều phổi, điều vận; quản lý nâng cao; các mô-đun quản lý đặc thù với từng loại

xe,... DSS cung cấp rất nhiều dịch vụ giám sát hoạt động của đội xe, cũng như về giấy phép hoạt động của các đội xe, lái xe.

Với những dự án rất hứa hẹn trên, chúng ta có quyền lạc quan và thêm tự tin để tham gia vào công cuộc nghiên cứu và phát triển xe tự hành của đất nước.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Thuật toán hoạch định quỹ đạo cho xe tự hành

2.1.1. Bài toán quỹ đạo tối ưu

Bài toán tìm quỹ đạo di chuyển ngắn nhất là vô cùng quan trọng với đề tài xe điện tự hành hoạt động trong phạm vi cục bộ. Vì khi xe không di chuyển trên các tuyến đường lớn nên việc sử dụng API của Google Maps để thiết lập quỹ đạo chuyển động là không thực hiện được.

Với một thuật toán tìm đường tối ưu thì yêu cầu đặt ra là quỹ đạo được xác định phải ngắn nhất thời gian chạy thuật toán là nhanh nhất. Có một số thuật toán để tìm ra con đường nhanh nhất. Các thuật toán là A *, Dijkstra và tìm kiếm đầu tiên theo chiều rộng (BFS). Các thuật toán này là các thuật toán tìm đường tốt nhất [2]. Tuy nhiên, mỗi thuật toán này đều có điểm yếu và điểm mạnh riêng trong quá trình xác định đường đi nhanh nhất. Một nghiên cứu khác so sánh các thuật toán A *, BFS, Dijkstra, HPA * và LPA * trong các lối không gian 2D. Thuật toán BFS có kết quả chậm nhất. Kết quả này có thể được giải thích là do nguyên tắc hoạt động của thuật toán rất đơn giản và nó không sử dụng bất kỳ phương pháp Heuristics nào. Thuật toán của Dijkstra nhanh hơn BFS, nhưng chậm hơn các thuật toán khác. A * và LPA * có cùng giá trị biểu diễn. LPA * nhanh hơn trong các lối nhỏ hơn (64, 128, 256), nhưng A * nhanh hơn trong các lối lớn hơn (512, 1024). Dẫn đến kết luận rằng LPA * phù hợp hơn cho các bài toán tìm đường nhỏ hơn, trong khi A * được sử dụng tốt hơn để giải các bài toán lớn hơn [3]. Với mục tiêu của đề tài thì xe điện tự hành sẽ hoạt động trong khu vực cục bộ với diện 325×190 (mét vuông), sau khi thực hiện tạo bản đồ số kích thước lối sẽ rơi vào khoảng (1450, 800). Như vậy dựa vào các kết quả và so sánh của các nghiên cứu đi trước ta sẽ chọn thuật A * để hoạch định đường đi ngắn và nhanh nhất cho phương tiện.

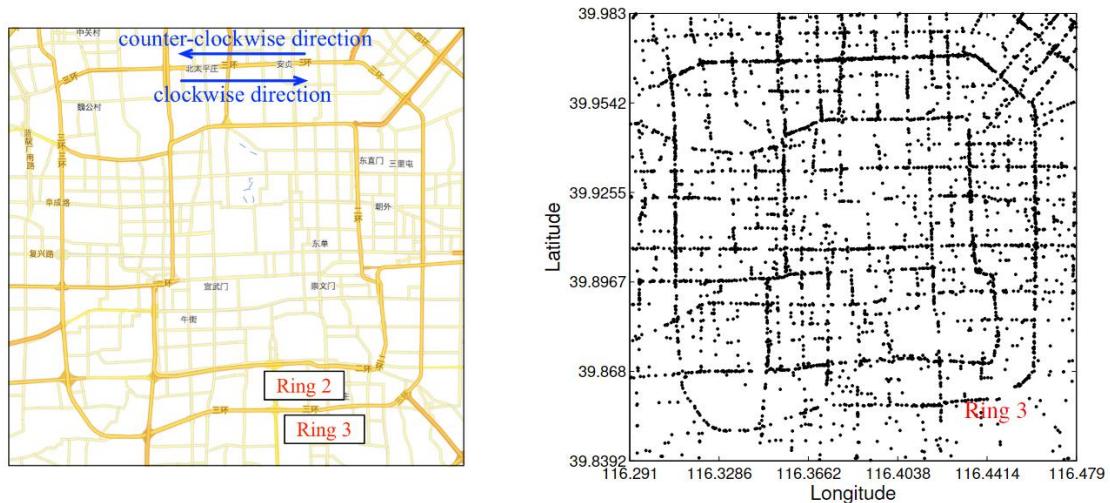
2.1.2. Thuật toán A *

Thuật toán A-Star được sử dụng để tìm đường đi ngắn nhất từ vị trí đầu đến vị trí cuối. Thuật toán sẽ tính toán kết hợp các chi phí từ điểm bắt đầu đến điểm hiện tại và phương pháp Heuristic ước tính vị trí hiện tại đến vị trí kết thúc làm cơ sở để giải bài toán hoạch định quỹ đạo tối ưu một cách hiệu quả. Cốt lõi của thuật toán sẽ tìm được quỹ đạo sao cho hàm $f(n) = g(n) + h(n)$ đạt giá trị nhỏ nhất, trong đó $g(n)$ là chi phí của đường dẫn từ nút bắt đầu đến nút thứ n , $h(n)$ là ước tính heuristic hoặc chi phí hoặc đường đi từ nút n đến mục tiêu, hàm $h(n)$ giúp việc tìm kiếm điểm đích có định hướng hơn, tối ưu thời gian và bộ nhớ để tính toán. Do đó, $f(n)$ ước tính tổng chi phí thấp nhất lời giải của quỹ đạo tối ưu đi qua nút n . Nút có giá trị $f(n)$ thấp nhất được chọn để mở rộng. Thuật toán kết thúc khi điểm mở rộng đó là điểm đích. Thuật toán A * hướng dẫn một con đường tối ưu đến một mục tiêu nếu hàm Heuristic $h(n)$ có thể chấp nhận được, có nghĩa là nó không bao giờ ước tính quá mức chi phí thực tế. Ví dụ: vì khoảng cách hàng không không bao giờ đánh giá quá mức khoảng cách đường cao tốc thực tế [4].

Để triển khai thực hiện thuật toán A * thông thường ta sẽ sử dụng hàng đợi ưu tiên (priority queue) để thực hiện việc lựa chọn lặp lại các nút có chi phí tối thiểu (ước tính) để mở rộng. Hàng đợi ưu tiên này được gọi là tập mở. Tại mỗi bước của thuật toán, nút có giá trị $f(n)$ thấp nhất bị lấy ra khỏi hàng đợi, các giá trị $f(n)$ và $g(n)$ của các nút lân cận của nó được cập nhật tương ứng và các nút lân cận này được thêm vào hàng đợi. Thuật toán tiếp tục cho đến khi một nút bị loại bỏ (do đó nút có giá trị $f(n)$ thấp nhất trong số tất cả các nút trong tập mở) là một nút mục tiêu. Giá trị $f(n)$ đó cũng là chi phí của đường đi ngắn nhất. Chi tiết các bước của thuật toán sẽ được trình bày ở phần giải thuật khi thực hiện thiết kế phần mềm hệ thống.

2.2. Phương pháp số hóa bản đồ cục bộ

Để thực hiện được thuật toán A * ta sẽ cần phải có được bản đồ của khu vực cục bộ mà xe sẽ hoạt động. Việc xây dựng bản đồ bằng cảm biến Lidar có được cân nhắc sử dụng, tuy nhiên mục tiêu phát triển lâu dài của xe tự hành mà nhóm đang thực hiện sẽ theo hướng vận hành hoàn toàn bằng trí tuệ nhân tạo và cảm biến Lidar sẽ không có trong hệ thống cảm biến của xe. Tuy là khu vực cục bộ nhưng xe vẫn hoạt động ngoài trời với diện tích tương đương với một khu du lịch, công viên, sân golf... có thể lên tới hàng chục nghìn mét vuông nên việc dùng Lidar hay Camera để tạo bản đồ cục bộ sẽ tốn rất nhiều thời gian và không thích hợp ở thời điểm hiện tại khi mới phát triển xe. Hơn nữa khi nhóm thực hiện đề tài, xe hoàn toàn chưa có yếu tố tự hành, khu vực mà xe hoạt động để kiểm tra quá trình điều khiển hay chạy các thuật toán ngoài thực tế lại thay đổi liên tục, yêu cầu hiện tại đặt ra là cần tạo ra một công cụ có thể giúp nhanh chóng tạo được bản đồ số mà nơi xe sẽ hoạt động. Nhóm đã tìm hiểu nhiều bài báo về cách thức thực hiện, trong đó “Mapping to Cells: A Simple Method to Extract Traffic Dynamics from Probe Vehicle Data” có đề xuất đến một phương án ánh xạ tới các ô để xây dựng một biểu đồ giao thông không gian cho mạng lưới đường cao tốc. Phương pháp này đơn giản vì nó hoàn toàn dựa trên dữ liệu tự thân và không có sự hỗ trợ của bất kỳ công cụ bổ sung nào như phần mềm [5].



Hình 2-1 Ánh xạ bản đồ

Dựa trên phương pháp này nhóm sẽ thu thập hình ảnh khu vực cục bộ từ vệ tinh để thực hiện số hóa lại bản đồ và đồng bộ với trở lại với vị trí kinh độ vĩ độ ngoài thực tế.

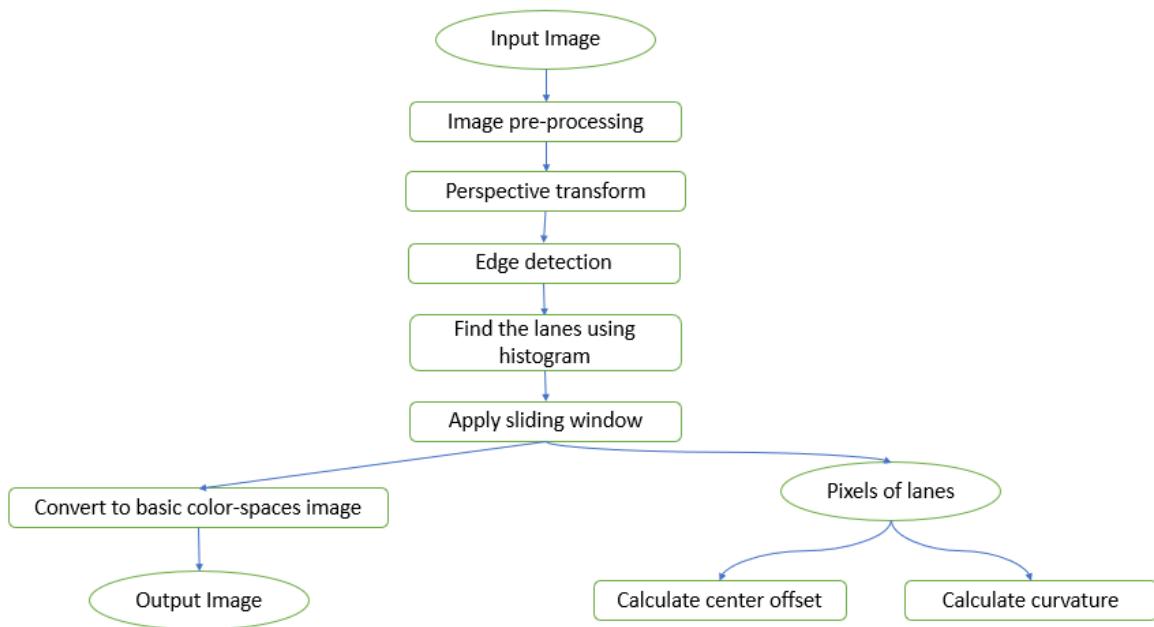
2.3. Phương pháp hồi quy tuyến tính ứng dụng cho xác định hướng di chuyển của xe

Quỹ đạo cần di chuyển của xe đã được xây dựng nhờ thuật toán A* và lưu trên bản đồ số hóa của khu vực cục bộ, mục tiêu tiếp theo là có thể ứng dụng lên xe điện tự hành, điều khiển xe đi theo quỹ đạo đã được hoạch định. Sử dụng tập dữ liệu trả về cảm biến GPS làm đầu vào cho phương pháp hồi quy tuyến tính để xác định phương hướng di chuyển hiện tại của xe, lấy vị trí hiện tại và một vị trí được hoạch định bởi quỹ đạo A* làm vector hướng chuyển động tiếp theo của xe, thực hiện tính góc lệch tạo bởi hai vector từ đó xác định đầu ra hướng di chuyển phù hợp sau đó là thiết lập tốc độ di chuyển để xe có thể tăng tốc khi đi thẳng hay giảm tốc để rẽ hướng an toàn. Thuật toán bám làn đường, phát hiện vật thể, bộ điều khiển trung tâm MPC sẽ giúp xe đi đúng làn đường, tránh né vật cản để di chuyển đến đích. Đầu ra của phương pháp hồi quy tuyến tính sẽ là một trong những đầu vào của bộ điều khiển MPC giúp xe nhận biết

được tiếp theo nên đi thẳng hay rẽ trái, rẽ phải từ đó thay đổi tốc độ và hướng đi để xe hoạt động một cách an toàn nhất.

2.4. Phương pháp phát hiện làn đường

Để xe có thể tự động bám làn đường trong quá trình tự hành, yêu cầu cơ bản là giữ cho xe luôn đi ở giữa phần làn đường dành cho xe, tức là giữ cho vị trí giữa xe trùng với vị trí chính giữa làn đường hiện tại. Cùng với đó, việc phát hiện làn đường sẽ được dùng để tính độ cong của làn đường (bán kính cong). Theo đó, độ lệch giữa vị trí tâm xe và tâm làn đường, cùng với bán kính cong là hai thông số đầu vào của bộ điều khiển đánh lái thực hiện bám làn MPC (Model Predictive Control). Để có thể xác định được điểm chính giữa làn đường ta cần xác định được làn đường nhờ vào yếu tố dễ thấy nhất chính là vạch kẻ đường, hoặc trong trường hợp không kẻ vạch kẻ đường thì chúng ta sẽ dựa vào những yếu tố khác để phân biệt giữa làn đường và lề đường hoặc dải phân cách. Ở đây ta xét đến trường hợp đường có vạch kẻ đường.



Hình 2-2 Lưu đồ giải thuật sử dụng xử lý ảnh để phát hiện làn đường

Trình tự các bước thực hiện phát hiện làn đường sử dụng xử lý ảnh [6]:

- Đọc ảnh đầu vào từ camera.

- Thực hiện hiệu chỉnh hình ảnh đầu vào (làm tối, tăng độ tương phản) và biến đổi các giá trị màu trong các khung gian màu khác nhau để ảnh thuận tiện nhất cho việc phát hiện các vạch kẻ đường.
- Chọn ROI (region of interest) để tập trung xử lý phần có chứa làn đường. Sử dụng các phép biến đổi khung hình để được ảnh với góc nhìn từ trên xuống (“bird-eye view”).
- Áp dụng histogram (phổ màu) và thuật toán Sliding window (cửa sổ trượt) để xác định vị trí của biên trái, biên phải của làn đường (xác định vạch kẻ đường).
- Thể hiện vạch kẻ đường bằng các đường màu khác lên ảnh đầu và cho ra kết quả.
- Tính toán bán kính cong, và khoảng cách giữa xe và vị trí giữa làn đường từ vị trí của các vạch kẻ đường để làm thông số đầu vào cho bộ điều khiển bám làn đường.

2.5. Bộ điều khiển

2.5.1. Bộ điều khiển Proportional Integral Derivative (PID)

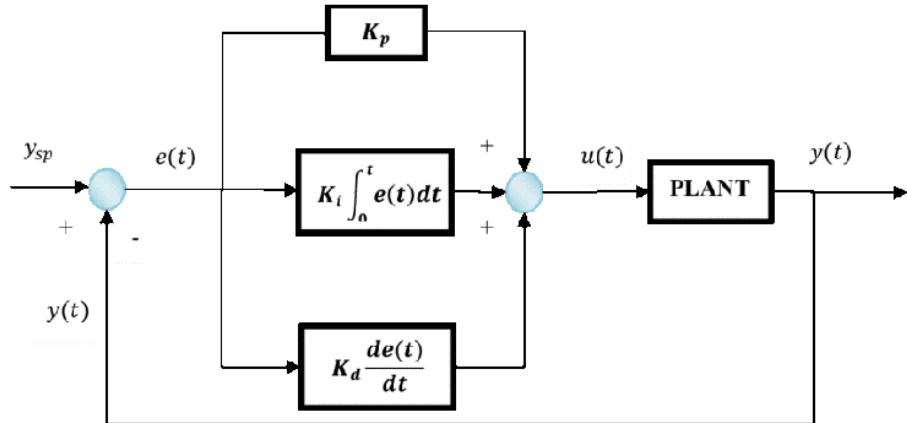
Hiện nay, bộ điều khiển vi tích phân tỉ lệ (Proportional Integral Derivative – PID) đang được sử dụng rộng rãi nhất trong các bộ điều khiển phản hồi. Trong trường hợp không có kiến thức cơ bản (mô hình toán học) về hệ thống điều khiển thì bộ điều khiển PID với các thông số được điều chỉnh thích hợp, là sẽ bộ điều khiển tốt nhất.

Giải thuật tính toán bộ điều khiển PID bao gồm 3 thông số riêng biệt, gồm giá trị tỉ lệ (P), tích phân (I) và đạo hàm (D). Khâu P xác định tác động của sai số hiện tại, khâu I xác định tác động của tổng các sai số quá khứ, và khâu D xác định tác động của tốc độ biến đổi sai số. Tổng chập của ba tác động này dùng để điều chỉnh quá trình thông qua một phần tử điều khiển.

Công thức tổng quát của bộ điều khiển PID liên tục

$$u_t = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$

Trong đó u_t là ngõ ra của bộ điều khiển ở thời điểm t , e_t là sai số giữa giá trị đặt và giá trị phản hồi cũng ở thời điểm t . K_p K_i K_d là các hằng số dương.



Hình 2-3 Sơ đồ bộ điều khiển PID

2.5.2. Bộ điều khiển Model Predictive Control (MPC)

Có nhiều nỗ lực nghiên cứu đã liên tục được tiến hành để phát triển các phương pháp lập quỹ đạo đường đi và bám theo quỹ đạo một cách hiệu quả cho phương tiện tự hành. Các công trình nghiên cứu ban đầu chủ yếu sử dụng các thuật toán lập kế hoạch dựa trên tìm kiếm trên bản đồ để tạo ra các đường đi ngắn nhất một cách hiệu quả và có chú ý đến các yếu tố của môi trường. Các phương pháp nổi tiếng như thuật toán Dijkstras, thuật toán A * và các biến thể của nó đã được sử dụng trước đây. Mặc dù đảm bảo tính hiệu quả và tính tối ưu của các phương pháp này, chất lượng của các quỹ đạo được hoạch định phụ thuộc rất nhiều vào độ phân giải của bản đồ. Bên cạnh đó, không thể dễ dàng tính đến những hạn chế của động lực phương tiện trong quá trình xây dựng quỹ đạo cho đường đi. Các công cụ xây dựng quỹ đạo cho đường đi sử dụng nội suy như sử dụng đường cong Clothoid, đường cong spline hay đường cong Bezier cũng đã được sử dụng rộng rãi để tính toán quỹ đạo online với lợi thế là

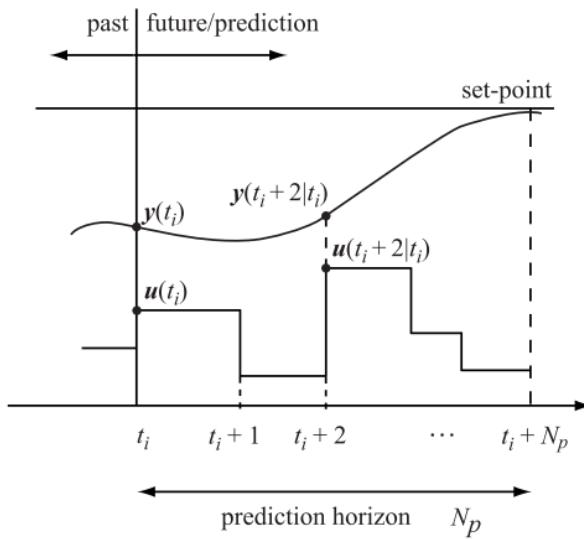
chi phí tính toán thấp, lượng tham số ít. Tuy nhiên, chúng không thể đảm bảo tính tối ưu của quỹ đạo được tạo ra và động lực học của phương tiện không được xem xét trong quá trình, vậy nên chúng thường cần phải đi kèm thuật toán làm mịn để tinh chỉnh các đường đi đã tạo sao cho chúng khả thi về mặt động lực học.

Các phương pháp tiếp cận dựa trên tối ưu hóa quỹ đạo ngày càng trở nên phổ biến trong việc lập kế hoạch đường đi cho xe tự lái, ý tưởng cốt lõi là xây dựng kế hoạch đường đi như một bài toán tối ưu hóa, có tính đến hiệu suất mong muốn của phương tiện và các ràng buộc liên quan. Do môi trường lái xe thường thay đổi phức tạp và ngẫu nhiên, và không thể dự đoán một cách chính xác hoàn toàn, nên phương pháp Điều khiển Dự báo dựa vào Mô hình (Model Predictive Control - MPC) thường được sử dụng để lập kế hoạch đường đi trực tuyến của xe tự lái trong những năm gần đây. Lý do là MPC có thể giải quyết một chuỗi bài toán tối ưu hóa quỹ đạo trong một khoảng thời gian hữu hạn nhờ tính toán đệ quy và có thể tính đến việc cập nhật trạng thái môi trường trong quá trình tính toán.

Bộ điều khiển dùng MPC tại mỗi thời điểm sẽ thực hiện lấy mẫu, thu thập các giá trị đầu vào và đầu ra hiện tại của đối tượng thực tế cũng như của mô hình để:

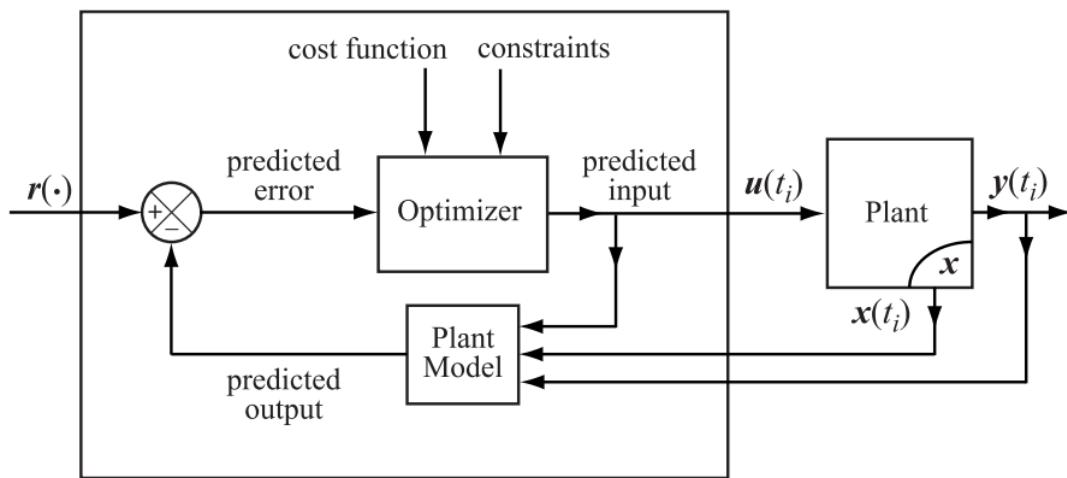
- Tính toán trong một phạm vi hữu hạn các tín hiệu điều khiển trong tương lai nhằm tối ưu hóa chỉ số hiệu suất nhất định và thỏa mãn các ràng buộc đối với tín hiệu điều khiển.
- Sử dụng tín hiệu điều khiển đầu tiên tính được làm tín hiệu điều khiển thực tế tại thời điểm hiện tại.

Ý tưởng của MPC được minh họa trong hình bên dưới, trong đó N_p là prediction horizon, là tín hiệu điều khiển dự đoán tại thời điểm $t + k$. Tương tự, là ngõ ra dự đoán tại $t + k$.



Hình 2-4 Ý tưởng điều khiển dùng Model Predictive Control

Kiến trúc tổng quan của một bộ điều khiển dựa vào mô hình Model Predictive Control:



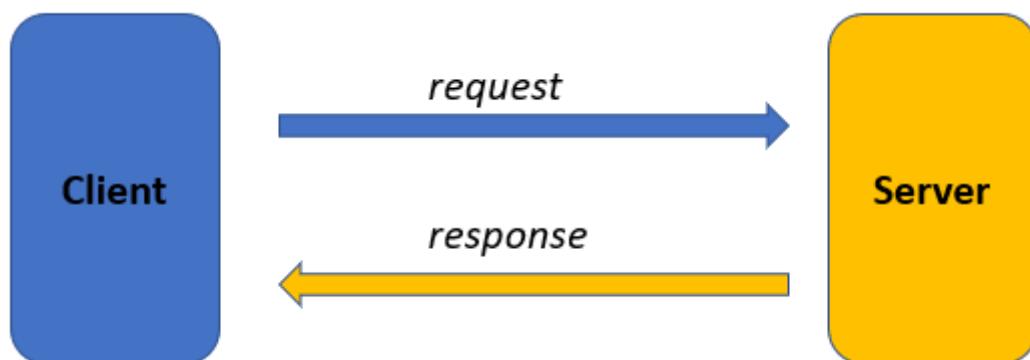
Hình 2-5 Kiến trúc tổng quan của Model Predictive Control

2.6. Mô hình webserver giám sát hoạt động của xe điện tự hành theo giao thức HTTP

2.6.1. Giao thức HTTP (HyperText Transfer Protocol) trong IoTs

HTTP (HyperText Transfer Protocol) là một giao thức rất quen thuộc và phổ biến trong lĩnh vực IoTs. Trước hết, HTTP là một giao thức có thể truyền nhận dữ liệu qua mạng, hoạt động theo cơ chế *request (yêu cầu)/response*

(phản hồi). Các client (trình duyệt, vi điều khiển, các thiết bị đầu cuối) gửi yêu cầu đến server và server sẽ trả về phản hồi các yêu cầu này.



Hình 2-6 Dạng yêu cầu/phản hồi của HTTP

❖ Hoạt động của HTTP

Dữ liệu HTTP truyền trên giao thức TCP, đảm bảo độ tin cậy của việc phân phối và chia nhỏ các yêu cầu và phản hồi dữ liệu lớn thành các phần mảng có thể quản lý được.

Ban đầu, client gửi một gói SYN đến Server và sau đó Server sẽ phản hồi với gói SYN-ACK để xác nhận việc nhận thành công. Tiếp theo, Client lại gửi một gói ACK, bao gồm một thiết lập kết nối – điều này cũng thường được gọi là bắt tay 3 bước. Ngoài ra, Client gửi một yêu cầu HTTP đến Server để tìm tài nguyên và đợi nó phản hồi một yêu cầu. Sau đó webserver sẽ xử lý yêu cầu, tìm tài nguyên và gửi phản hồi đến Client. Nếu Client không yêu cầu thêm tài nguyên, nó sẽ gửi gói FIN để đóng kết nối TCP.

❖ Ứng dụng của HTTP

Giao thức HTTP được sử dụng để khởi động World Wide Web để truyền dữ liệu dưới dạng văn bản, âm thanh, hình ảnh và video từ Server đến trình duyệt web của người dùng và ngược lại. HTTP hiện là nền tảng truyền dữ liệu của ứng dụng duyệt web ngày nay và được sử dụng rộng rãi trong hệ thống IoTs. Mặc dù giao thức Http có nhiều nhược điểm trong việc truyền dữ liệu và không phù hợp bằng các giao thức tối ưu khác như MQTT, CoAP, AMQP sử

dụng cho IoT, nhưng giao thức này vẫn phổ biến trong lĩnh vực nhà thông minh cũng như việc sử dụng nhiều trong bộ vi điều khiển và vi xử lý tiên tiến.

❖ *Ưu và nhược điểm của giao thức HTTP*

HTTP là giao thức IoT phổ biến bởi những ưu điểm nổi trội về khả năng tìm kiếm khi nó có khả năng truy cập đến cơ sở dữ liệu với một yêu cầu duy nhất; HTTP dễ lập trình do được mã hóa dưới dạng văn bản thuận tiện để theo dõi và triển khai hơn các giao thức sử dụng mã yêu cầu tra cứu - điều này khiến HTTP phù hợp với những người mới tiếp xúc với webserver. Bên cạnh đó, do là một giao thức đã được phát triển khá lâu đời nên HTTP cũng tồn tại một vài nhược điểm về kích thước các gói tin; các hạn chế trong các ứng dụng giao tiếp theo sự kiện hay yêu cầu về thời gian thực do HTTP hoạt động theo mô hình yêu cầu/phản hồi tức là khi một client gửi yêu cầu sẽ phải chờ server phản hồi mới có thể gửi một yêu cầu mới.

Các phương thức (method) truy vấn đến server của HTTP:

- GET là phương thức yêu cầu dữ liệu đơn giản và thường sử dụng nhất của HTTP. Phương thức GET yêu cầu server chỉ trả về dữ liệu bằng việc cung cấp các thông tin truy vấn trên URL, thông thường Server căn cứ vào thông tin truy vấn đó trả về dữ liệu mà không thay đổi nó.

https://fleetmonitoring.000webhostapp.com/searchGPSdata.php?frame_ID=1
| protocol | server | path | query

Hình 2-7 Ví dụ URL cho 1 GET request trong đề tài

Các thông tin truy vấn của gói tin chứa trong path và query.

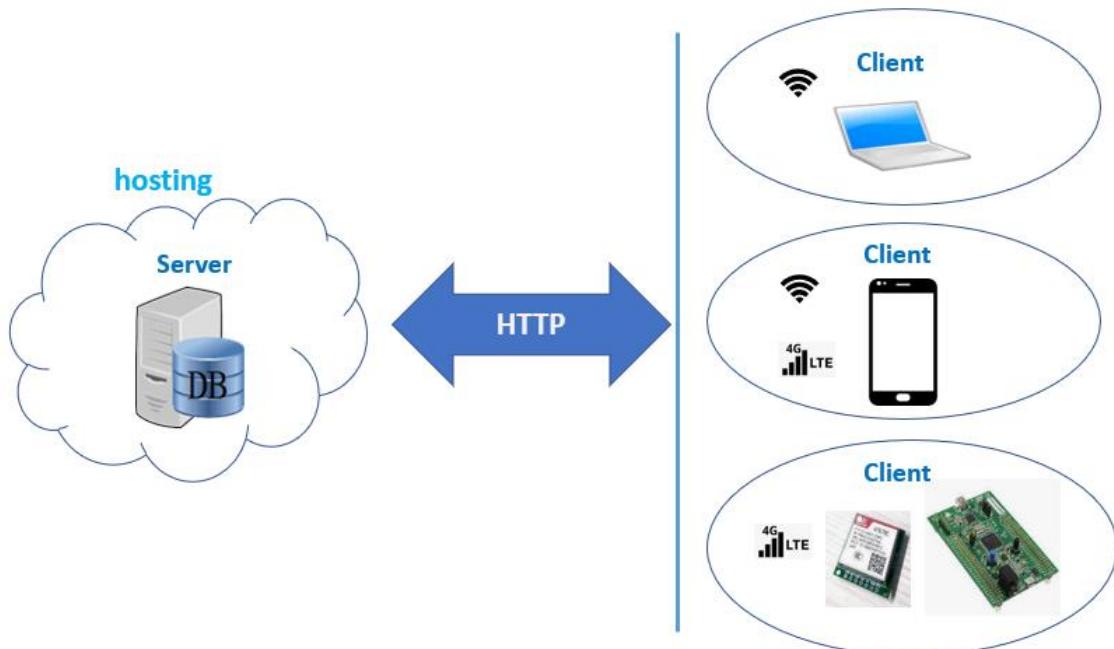
- POST tương tự như GET, nhưng POST có thể gửi dữ liệu về server. Khác với GET, nội dung của gói tin không hiển thị trên URL.
- PUT là phương thức yêu cầu tạo mới một dữ liệu, giống POST nhưng PUT đánh dấu cho server biết, nếu dữ liệu không tồn tại trong cơ sở dữ liệu thì tạo mới, hoặc sửa đổi nó.

- DELETE: tương tự như GET, nhưng báo cho Server biết về việc xóa dữ liệu thông qua URL.

Khi các yêu cầu được gửi đến server, server sẽ kiểm tra và xử lý sau đó trả về các phản hồi cho phía client. Trong các gói tin phản hồi của server luôn chứa các *status code* là trạng thái của truy vấn mà phía client thực hiện: như **200 OK** (truy vấn thực hiện thành công), hay các mã báo có lỗi như **400 Bad Request, 404 Not Found**.

2.6.2. Mô hình webserver giám sát hoạt động của xe điện tự hành theo giao thức HTTP

Tương tự như mô hình của một truy vấn HTTP, một mô hình webserver cũng có hai thành phần chính là server và client thực hiện giao tiếp với nhau theo mô hình request (yêu cầu)/response (phản hồi). Để các ứng dụng có thể truy cập mọi lúc mọi nơi thì server phải được thực thi trên một máy chủ trên mạng (hosting) – máy chủ hoạt động liên tục để các client có thể truy cập từ bất kỳ đâu miễn là client có kết nối internet. Trên máy chủ đó sẽ có tích hợp các hệ cơ sở dữ liệu để lưu trữ dữ liệu trong quá trình kết nối.



Hình 2-8 Mô hình webserver theo giao thức HTTP

Server và cơ sở dữ liệu được lưu trữ trên một máy chủ trên mạng (hosting) để các client có thể truy cập mọi lúc mọi nơi miễn là các thiết bị hay trình duyệt đó được kết nối mạng (wifi, 4G). Khi client gửi yêu một http request thì yêu cầu đó sẽ được gửi đến server xử lý, sau đó các truy vấn đến cơ sở dữ liệu (như tìm kiếm hay lưu thông tin) có thể được thực hiện, đồng thời server sẽ trả phản hồi về cho client. Cứ như vậy, quá trình trao đổi của một server HTTP được diễn ra.

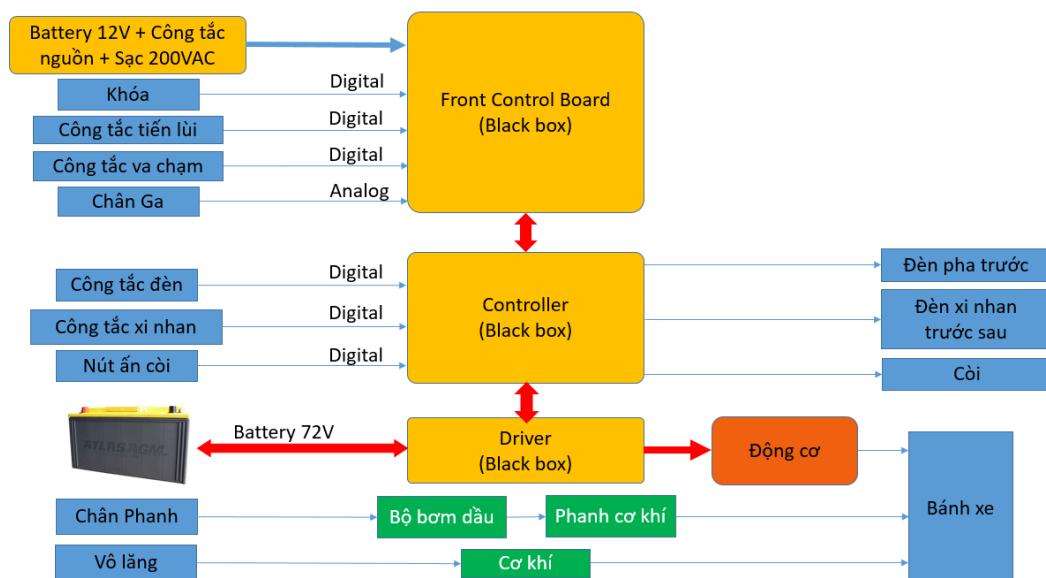
Đó là những lý thuyết chính cho một Webserver theo giao thức HTTP. Trong đề tài, việc giám sát dữ liệu hoạt động của xe tự hành hay các điều khiển cơ bản được gửi từ trung tâm vận hành đến xe cũng được thực hiện theo mô hình tương tự qua mạng di động 4G.

Chương 3. THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN CHO XE ĐIỆN TỰ HÀNH

3.1. Tổng quan hệ thống điều khiển trên xe điện



Hình 3-1 Hình ảnh thực tế xe điện 4 chỗ Sanyo

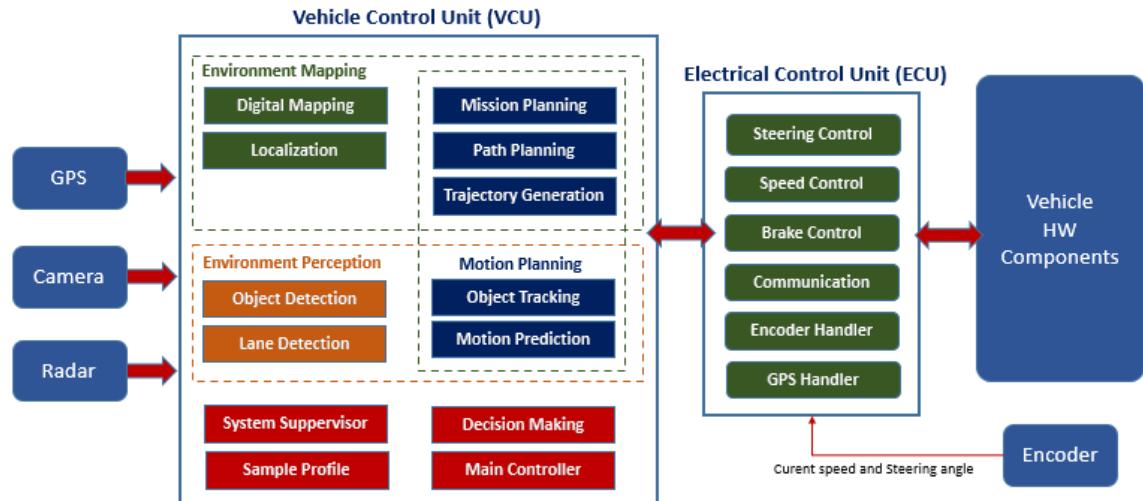


Hình 3-2 Tổng quan phần điều khiển trên xe

Để chuyển phần điều khiển bằng tay bằng điều khiển bằng điện thì các tín hiệu ngõ vào cơ khí cần được thay thế bằng tín hiệu điều khiển mới. Quan trọng bậc nhất là các tín hiệu công tắc tiến lùi, tín hiệu chân phanh, tín hiệu chân ga và góc đánh lái cho xe. Không những thế, để thỏa điều kiện điều khiển tự động, xe cần được trang bị thêm những thiết bị và cảm biến mới. Như vậy

một hoặc nhiều những mạch điện tử mới cần được phát triển để đọc các cảm biến và tạo ra các tín hiệu điều khiển nói trên.

3.2. Tổng quan hệ thống điều khiển trên xe điện tự hành

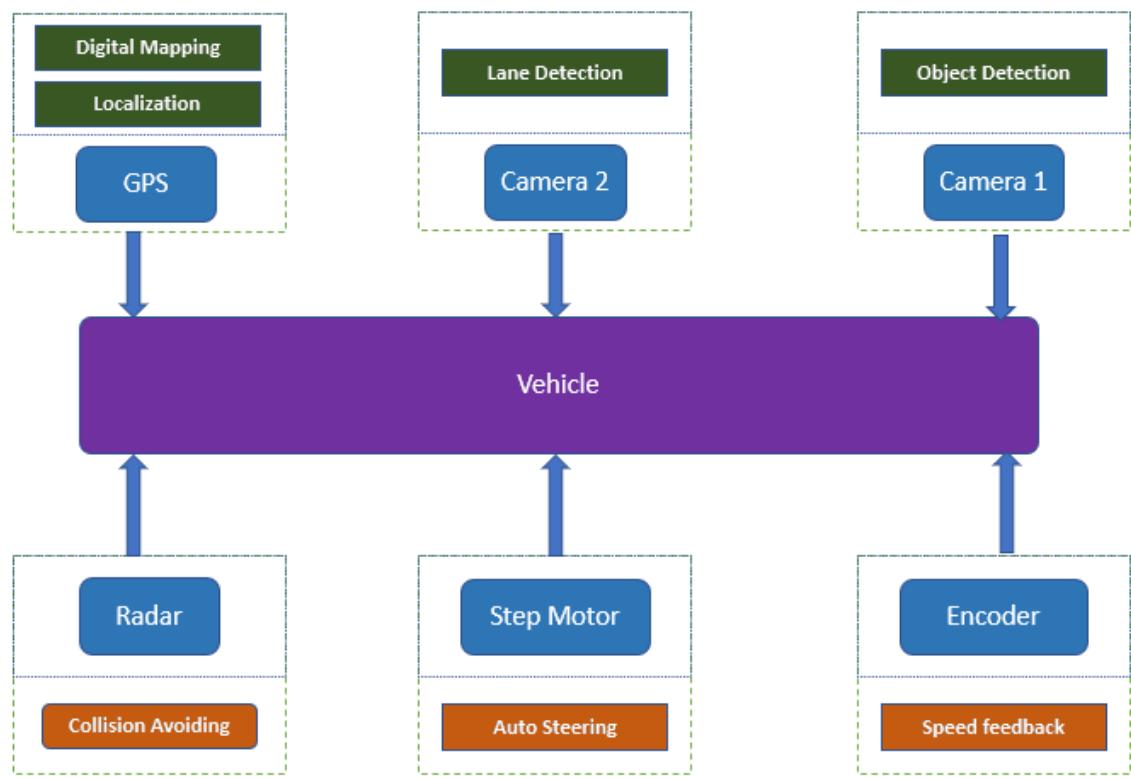


Hình 3-3 Tổng quan hệ thống điều khiển trên xe điện tự hành

Hệ thống điều khiển mới được thêm vào cho xe bao gồm hai tầng điều khiển chính. Thứ nhất, bộ điều khiển trung tâm là Vehicle Control Unit (VCU), nó là một máy tính có tốc độ xử lý tốt đủ khả năng thực hiện nhiều tác vụ cũng như các thuật toán mà những vi điều khiển khó lòng đáp ứng như các thuật toán về xử lý ảnh phát hiện làn đường, hay phát hiện vật thể... Tầng thứ hai là các board mạch điện tử - Electrical Control Unit (ECU) - có chức năng giao tiếp với bộ độ điều khiển trung tâm, nhận các lệnh điều khiển và điều khiển các thiết bị chấp hành trên xe.Thêm vào đó, các cảm biến được tích hợp thêm vào xe để đọc về các trạng thái hoạt động, những ECU sẽ đảm nhận luôn những công việc này. Trong thực tế, trên xe ô tô thường có rất nhiều những ECU như vậy, mỗi một ECU thường mang một chức năng chuyên biệt, như có ECU chỉ mang nhiệm vụ điều khiển thiết bị chấp hành motor, có ECU chỉ có nhiệm vụ đọc cảm biến, có ECU chỉ điều khiển hệ thống đèn điện trên xe.

Để cung cấp các dữ liệu đầu vào là các thông số hoạt động của xe hay góc đánh lái, các thông số về môi trường hoạt động của xe như làn đường, vật cản,

các cảm biến chuyên dụng được tích hợp vào xe để hỗ trợ thu thập những dữ liệu nêu trên.



Hình 3-4 Hệ thống các cảm biến được tích hợp vào xe điện hỗ trợ quá trình tự hành

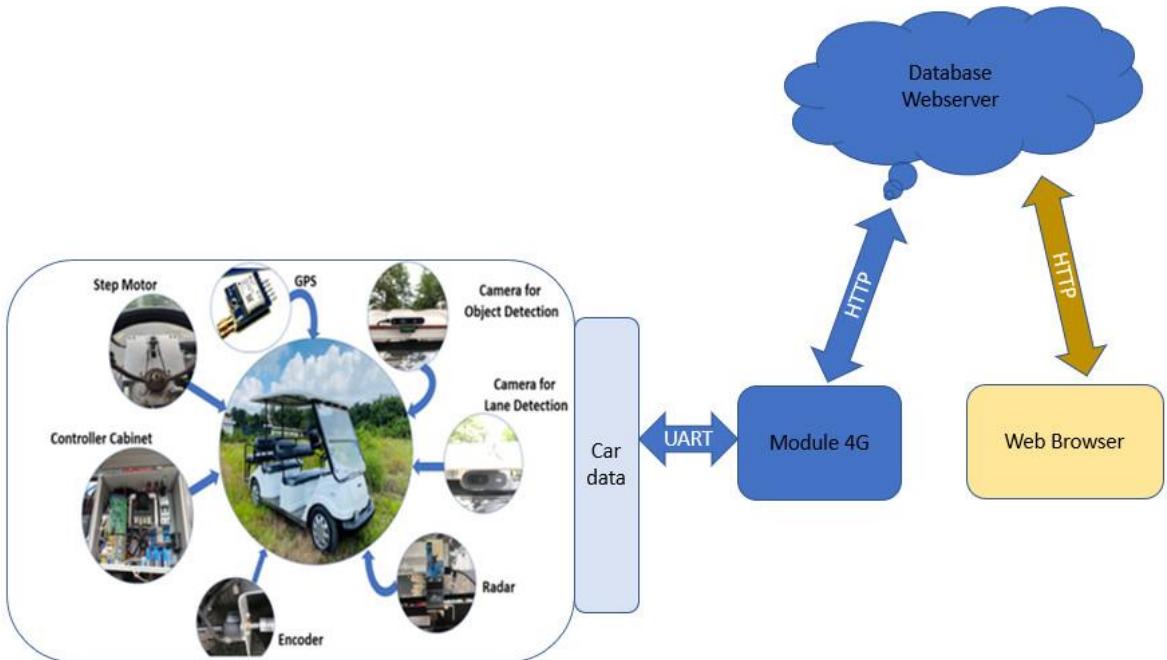


Hình 3-5 Hình ảnh thực tế của các cảm biến, và thiết bị tích hợp vào xe điện tự hành

Sau quá trình nhóm thực hiện tự hành cho xe, xe điện hiện đã được thay đổi tích hợp thêm một số mô-đun: động cơ bước -hộp số có trả về thông số tốc độ để điều khiển vô lăng, encoder để thực hiện đọc tốc độ của xe, camera để thực hiện phát hiện vật thể, camera thực hiện nhiệm vụ phát hiện làn đường, radar thực hiện phát hiện đối tượng để tránh va chạm, bộ định vị GPS để xác định vị trí của xe. Một tủ điện thực hiện kết nối các mô-đun lại với nhau và kết nối với hệ thống điện trên xe.

3.3. Tổng quan hệ thống giám sát hoạt động của xe tự hành qua mạng 4G

Sau quá trình thực hiện tự hành cho xe điện, ý tưởng tiếp theo của nhóm là thực hiện giám sát hoạt động và xa hơn là có thể điều khiển xe từ xa. Theo ý tưởng này, yêu cầu là phải thiết kế một hệ thống webserver để có thể kết nối giữa xe và trung tâm điều khiển.



Hình 3-6 Mô hình hệ thống IoT cho xe tự hành qua giao thức HTTP

Các dữ liệu của xe điện được quy định thành các gói tin UART gửi cho môđun 4G, cùng với các lệnh thực hiện để POST dữ liệu lên cơ sở dữ liệu trên mạng theo giao thức HTTP. Các dữ liệu được xử lý và lưu trữ trên cơ sở dữ liệu. Từ trình duyệt người dùng có thể truy cập vào trang web giám sát dữ liệu của các xe tự hành, khi đó trình duyệt sẽ gửi lên Webserver một HTTP request, phản hồi trả về nếu truy cập thành công là sẽ là một bản đồ trên đó thể hiện vị trí hiện tại của các xe.

Chương 4. THIẾT KẾ HỆ THỐNG PHẦN CỨNG

4.1. Sơ đồ kết nối phần cứng hệ thống truyền nhận dữ liệu qua mạng 4G

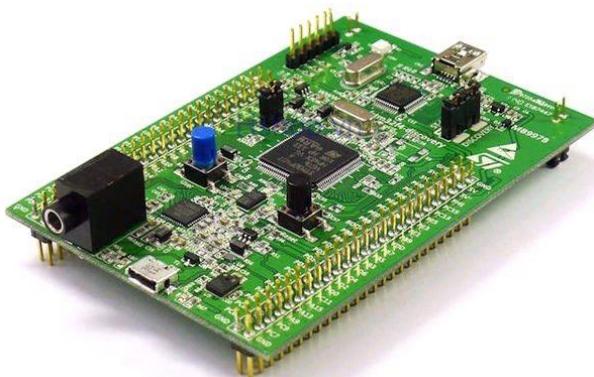


Hình 4-1 Sơ đồ tổng quan hệ thống phần cứng truyền nhận dữ liệu qua mạng 4G

Dữ liệu từ mô-đun GPS (hay lớn hơn là hệ thống dữ liệu của xe) được giao tiếp nối tiếp với một vi điều khiển có chức năng biến đổi hay đóng gói các dữ liệu đó thành các gói tin để có thể thực hiện truyền nhận dữ liệu đến server qua mô-đun 4G. Đồng thời dữ liệu vị trí cũng được gửi lên giao diện điều khiển để hiển thị vị trí hiện tại của phương tiện đồng thời làm dữ liệu để xây dựng phương pháp điều hướng theo quỹ đạo đã hoạch định.

4.2. Tổng quan các thiết bị

4.2.1. Vi xử lý STM32F407



Hình 4-2 Board STM32F407 Discovery

Kit STM32F4DISCOVERY thúc đẩy khả năng của vi điều khiển hiệu suất cao STM32F407, cho phép người dùng dễ dàng phát triển các ứng dụng. Kit bao

gồm một bộ công cụ gỡ lỗi nhúng ST-LINK, một gia tốc số ST-MEMS, một micro số, một audio DAC với trình điều khiển loa tích hợp lớp D, đèn LED, nút bấm và cổng micro-AB USB OTG.

Các đặc điểm tính năng chính

- Bộ vi xử lý STM32F407VGT6: 32-bit ARM Cortex – M4 với lõi FPU, bộ nhớ Flash 1MB, RAM 192KB với kiểu chân LQFP100.
- Trình gỡ lỗi ST-LINK/V2 trên STM32F4DISCOVERY.
- USB ST-LINK với khả năng kiểm tra lại và ba giao diện khác nhau:
 - Cổng gỡ lỗi.
 - Cổng COM ảo.
 - Lưu trữ khối lượng.
- Nguồn cung cấp: thông qua USB hoặc nguồn điện bên ngoài 3V – 5V.
- Gia tốc kế 3 trục LIS302DL hoặc LIS3DSH ST MEMS.
- Micro số đa hướng dùng cảm biến âm thanh MP45DT02 ST-MEMS.
- CS43L22 bộ chuyển đổi âm thanh/kỹ thuật số âm thanh công suất thấp (DAC) với trình điều khiển loa tích hợp lớp D.
- Tám đèn LED:
 - LD1 (đỏ/xanh) cho giao tiếp USB.
 - LD2 (đỏ) led báo nguồn 3.3V.
 - Bốn led của người dùng, LD3 (màu cam), LD4 (xanh lục), LD5 (màu đỏ) và LD6 (màu xanh lam).
 - 2 đèn led OTG USB LD7 (xanh lá cây) VBUS và LD8 (màu đỏ) quá dòng điện.
- Hai nút nhấn (người dùng và reset).
- USB OTG FS có cổng kết nối micro-AB.
- Header mở rộng I/O cho LQPF64 để kết nối nhanh với bo mạch gốc và dễ dàng sử dụng.

Bảng 4-1 Thông số kỹ thuật vi xử lý STM32F407

Thông số kỹ thuật	Mô tả
Nguồn cấp	3V – 5V DC
Bộ nhớ	1MB Flash, 192 KB RAM
Hỗ trợ mạch nạp và debug	ST – LINK/V2
Tần số xung Clock	168MHz

4.2.2. Thiết bị GPS

Phương tiện xe điện mà nhóm đang thực hiện đang trong giai đoạn đầu phát triển tính năng tự hành. Vì vậy, để tiếp cận nhanh chóng nhóm luôn hướng hướng đến mục tiêu thuận tiện, sử dụng các mô-đun có sẵn trên thị trường. Vấn đề thiết kế phần cứng để thu thập cơ sở dữ liệu sẽ tận dụng những board mạch đang lưu thông trong thị trường từ đó tiến hành thực nghiệm kiểm định lại sai số và tiến hành tích hợp nhiều phương pháp thiết kế lắp đặt để tăng độ chính xác và độ tin cậy cho tập cơ sở dữ liệu nhận được khi cần thiết. Qua quá trình tìm hiểu cũng như khảo sát về giá thành lẫn độ chính xác của dữ liệu thu thập đã chọn được mô-đun phù hợp sau:



Hình 4-3 Mô – đun GPS U-Blox NEO-7N

Một số thông số quan trọng trong Data Sheet của mô-đun được liệt kê trong bảng sau:

Bảng 4-2 Một số thông số quan trọng của thiết bị GPS NEO 7N

Thông số kỹ thuật	Mô tả
Nguồn cấp	3.3V đến 5V
Tần số L1	1575.42 MHz
Bộ nhận	56 kênh, GPS L1C/A, SBAS, L1C/A, QZSS L1C/A, Galileo E1B/C
Độ chính xác theo phương ngang	2.5m
Chuẩn kết nối	SMA và ipex
Tốc độ cập nhật tối đa	10Hz

Mô – đun GPS giúp theo dõi vị trí của phương tiện và thông báo người dùng khi xe đến đích, đồng thời thông qua tập dữ liệu trả về từ GPS ta có thể xác định được phương hướng di chuyển của xe. Từ đó xuất các tín hiệu hoạch

định di chuyển đến bộ điều khiển trung tâm đi thẳng, rẽ trái hay phải, dựa vào đó thì quyết định xe nên tăng tốc hay giảm tốc. Tín hiệu điều hướng này sẽ được đánh trọng số và làm đầu vào cho bộ điều khiển trung tâm của xe và đầu vào cho các giải thuật bám làn vây nên không yêu cầu quá cao về độ chính xác của vị trí mà quan trọng hơn là xác định được xu hướng chuyển động của xe. Đã có những ứng dụng thực hiện thành công trên các thiết bị GPS giá thành thấp [7], hay những nghiên cứu nhằm cải thiện độ chính xác cho các loại GPS này [8] [9].

Đồng thời mô – đun này được nhà sản xuất chú tâm phát triển, xây dựng giao diện giúp cấu hình, đặt lại các thông số phù hợp cho từng ứng dụng khác nhau. Thiết bị GPS sẽ trả về rất nhiều gói tin, tuy nhiên với mục tiêu chỉ lấy vị trí hiện tại, nhóm đã thực hiện cấu hình lại chỉ nhận một gói tin vị trí, nhằm giảm bớt lượng dữ liệu khi xử lý. Chu kỳ của các gói tin cũng có thể được đặt lại với tần số nhanh nhất lên đến 20Hz, thông số này sẽ được hiệu chỉnh khi tiến hành thực nghiệm.

4.2.3. Module 4G



Hình 4-4 Module 4G/LTE SIM A7670C

Dòng module SIM A7670 được sử dụng khá phổ biến trong cộng đồng sinh viên với mục đích truyền tải dữ liệu qua Internet. Chỉ cần nguồn cung cấp ổn định mô-đun sẽ hoạt động rất tốt.

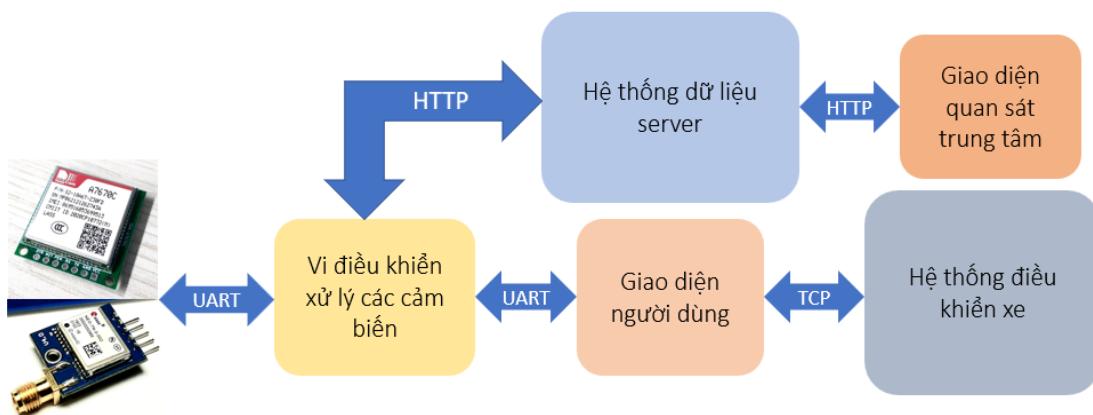
Bảng 4-3 Một số thông số quan trọng của Module 4G/LTE SIM A7670C

Thông số kỹ thuật	Mô tả
Kích thước (mm)	24.0*24.0*2.3
Nhiệt độ làm việc	-40°C ~ +85°C
Nguồn cấp (V)	3.4~ 4.2
Dòng tiêu thụ (mA) - (LTE)	3.8
Tốc độ truyền tải (Mbps) - (LTE)	10(DL)/5(UL)
Giao thức hỗ trợ	TCP/IP/IPV4/IPV6/Multi-PDP/FTP/FTPS /HTTP/HTTPS/DNS

Chương 5. THIẾT KẾ HỆ THỐNG PHẦN MỀM HOẠCH ĐỊNH QUÝ ĐẠO VÀ GIAO DIỆN ĐIỀU KHIỂN CHO XE TỰ HÀNH

5.1. Tổng quan kết nối hệ thống phần mềm

Để tài hướng đến phát triển một hệ thống có thể giám sát và điều khiển đội xe từ xa. Vì vậy dưới đây là phần phát triển của đề tài song song cùng lúc với các thành viên khác trong nhóm thực hiện chung trên xe tự hành.



Hình 5-1 Tổng quan hệ thống hoạch định quý đạo và gửi dữ liệu lên server

Hệ thống bao gồm vi điều khiển giúp xử lý các luồng thông tin từ cảm biến thông qua chuẩn UART. Các thông tin vị trí sẽ được hiển thị trên giao diện người dùng, đồng thời cũng sẽ được gửi lên hệ thống dữ liệu của server quan giao thức HTTP phục vụ quá trình giám sát từ xa. Giao diện sẽ nhận tín hiệu vị trí và địa điểm đích từ đó hoạch định quý đạo di chuyển cho xe, giao diện cũng hỗ trợ người dùng có thể trực tiếp chọn điểm bắt đầu và điểm đích cho quá trình di chuyển ngay trên xe. Sau khi hoạch định được quý đạo chuyển động hệ thống bắt đầu hoạch định chuyển động, giao diện sẽ gửi hướng và vận tốc đến bộ điều khiển trung tâm thông qua kết nối TCP trong mạng LAN. Quá trình bám làn và thiết lập góc đánh lái sẽ do thuật toán phát hiện làn đường kết hợp với bộ điều khiển MPC tại trung tâm phụ trách. Khi đến điểm đích kết thúc quá

trình di chuyển, giao diện sẽ hiển thị thông báo cho người dùng và gửi tín hiệu đến trung tâm cho xe dừng lại.

Robot Operating System – ROS là hệ điều hành mã nguồn mở dành cho robot, là một framework được dùng rất rộng rãi trong lĩnh vực robotic với nhiều ưu điểm. Nó tạo ra một nền tảng phần mềm có thể hoạt động trên rất nhiều robot khác nhau mà không cần sự thay đổi quá nhiều trong chương trình phần mềm. ROS cung cấp môi trường phát triển giúp các phần mềm kết nối với nhau một cách dễ dàng, nó giúp cho việc sử dụng lại và tích hợp nhiều mã nguồn mở trở nên đơn giản hơn nhiều

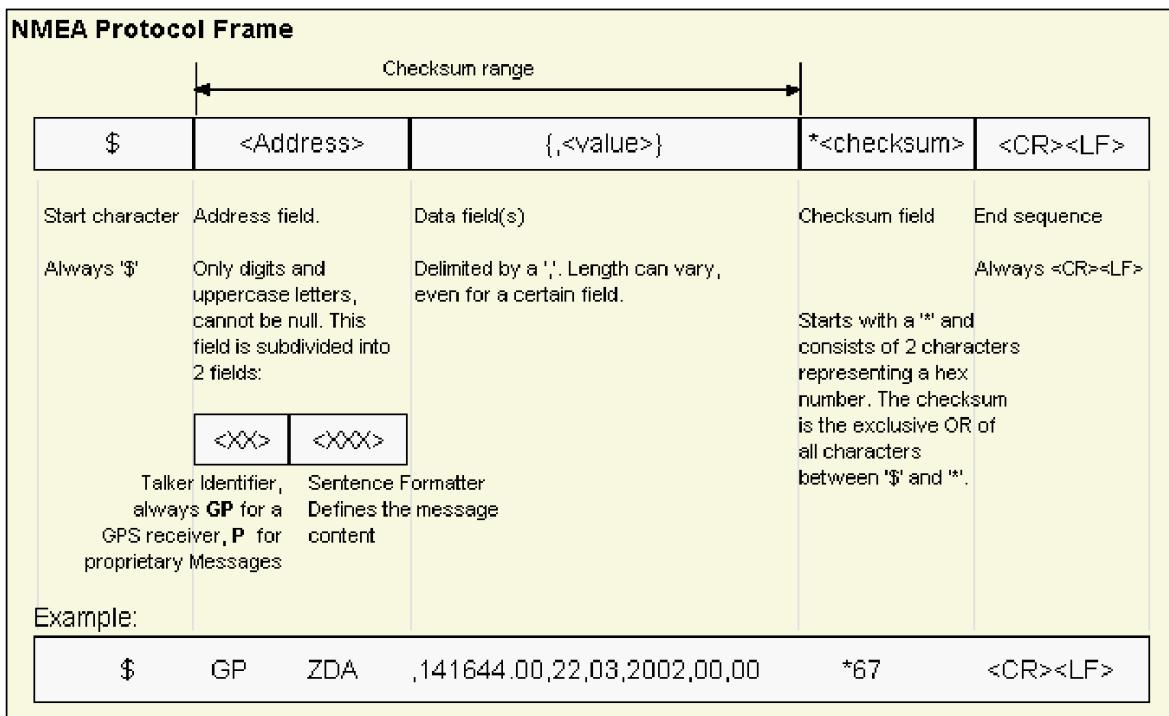


Hình 5-2 Robot Operating System

Giao diện điều khiển được xem như một node của hệ thống và gửi lệnh điều khiển đến trung tâm thông qua kết nối TCP.

5.2. Xử lý dữ liệu từ cảm biến GPS

Các bản tin NMEA được gửi bởi bộ thu GNSS dựa trên giao thức NMEA, sau đó mô-đun GPS NEO 7M sử dụng chuẩn giao tiếp truyền nhận nối tiếp bất đồng bộ (UART) để truyền các gói tin đó đến ngõ ra.



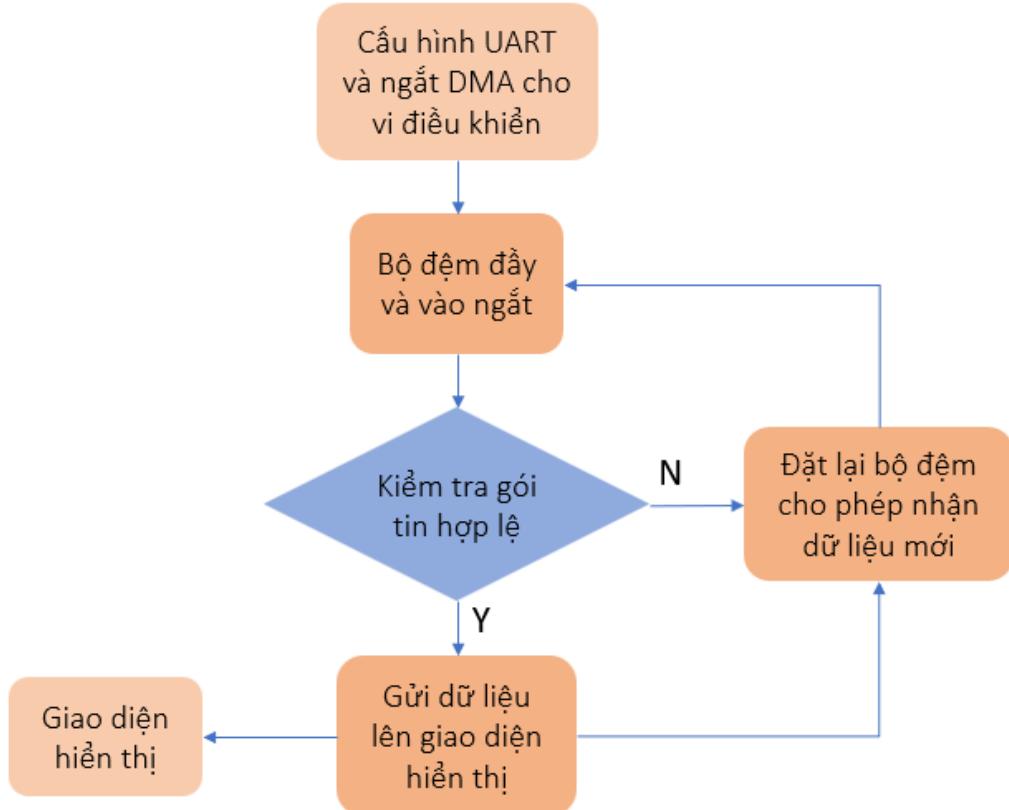
Hình 5-3 Cấu trúc một gói tin giao thức NMEA [10]

\$GPGLL,1045.58122,N,10639.71576,E,023523.00,A,A*6F{0D}{0A}

Hình 5-4 Cấu trúc gói tin GPGLL của giao thức NMEA

Đây là gói tin cần được xử lý, theo chuẩn NMEA nó bao gồm ký tự bắt đầu '\$', trường địa chỉ "GPGLL" để mô tả nội dung của gói tin cụ thể ở đây là kinh độ và vĩ độ của điểm hiện tại, với phần giá trị của vĩ độ thì hai số đầu tiên là phần độ các số tiếp theo là phần phút, với phần giá trị của kinh độ thì ba số đầu tiên là phần độ các số tiếp theo là phần phút, nên cần phải thực hiện tính toán chuyển đổi để được tọa độ đúng, cuối cùng sẽ là kiểm tra và xuống dòng kết thúc gói tin.

Như vậy, để nhận được gói tin ta sẽ thực hiện cấu hình UART và ngắt nhận DMA của vi điều khiển để nhận dữ liệu gửi về từ GPS. Vấn đề cần xử lý với mô-đun này là những lúc nó mất tin hiệu và không trả về giá trị kinh độ vĩ độ. Sơ đồ dưới đây thể hiện các xử lý.



Hình 5-5 Quy trình xử lý dữ liệu nhận được từ GPS

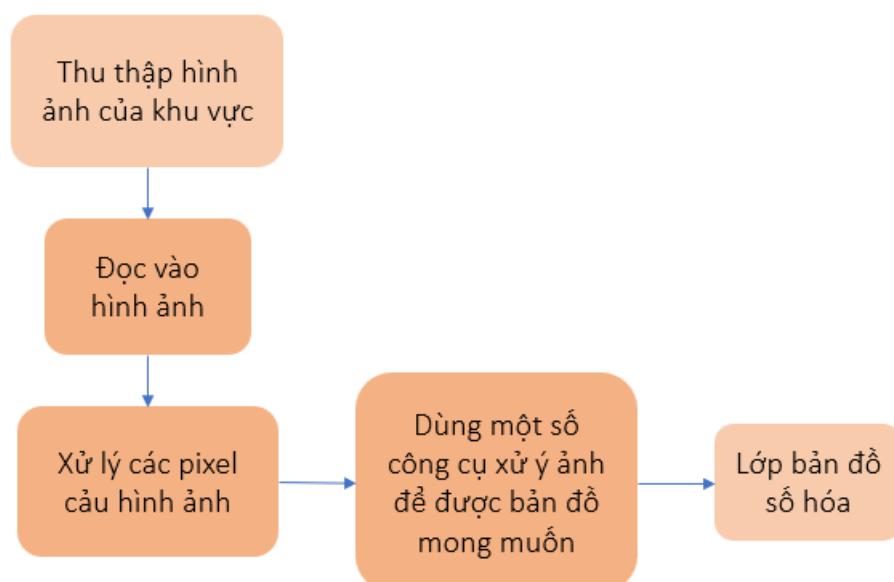
Với mô-đun hiện tại đã cấu hình chỉ gửi về một frame mình mong muốn để lấy dữ liệu kinh độ vĩ độ “GPGLL”. Gói tin này có độ dài 52 bytes nên kích thước của bộ đệm để nhận dữ liệu GPS cũng sẽ độ dài tương tự, tần số trả về được cấu hình là 100ms cho một gói tin. Với sự hỗ trợ ngắt DMA từ vi điều khiển mỗi khi nhận được gói tin đúng độ dài đó thì sẽ diễn ra quá trình ngắt. Ngắt sẽ thực hiện việc gửi dữ liệu này qua máy tính và hiển thị lên giao diện định vị được vị trí hiện tại.

5.3. Xây dựng và hiển thị bản đồ số

5.3.1. Xây dựng bản đồ số khu vực cục bộ

Mục tiêu lâu dài và lý tưởng của bản đồ số là xây dựng được một bản đồ chi tiết nhất cho khu vực đó mà Google Maps không thể hiện được (các đặc trưng từ đường đi đến các địa điểm của khu vực đó). Muốn được như vậy thì cần phải có được bản thiết kế của khu vực đó sau đó đồng bộ nó với tọa độ trên

Google Maps. Tuy nhiên nhóm không tìm được một địa điểm như vậy để chạy thực nghiệm xe. Vì vậy, nhóm đã chọn một khu vực khu dân cư, sau đó dùng hình ảnh trên Google Maps để thực hiện số hóa và đồng bộ tọa độ với tọa độ thực. Lớp bản đồ số này chỉ bao gồm các con đường mà xe sẽ hoạt động, sau đó các thuật toán tìm đường và hoạch định chuyển động sẽ được nhúng vào lớp bản đồ này.



Hình 5-6 Quy trình thực hiện số hóa khu vực cục bộ

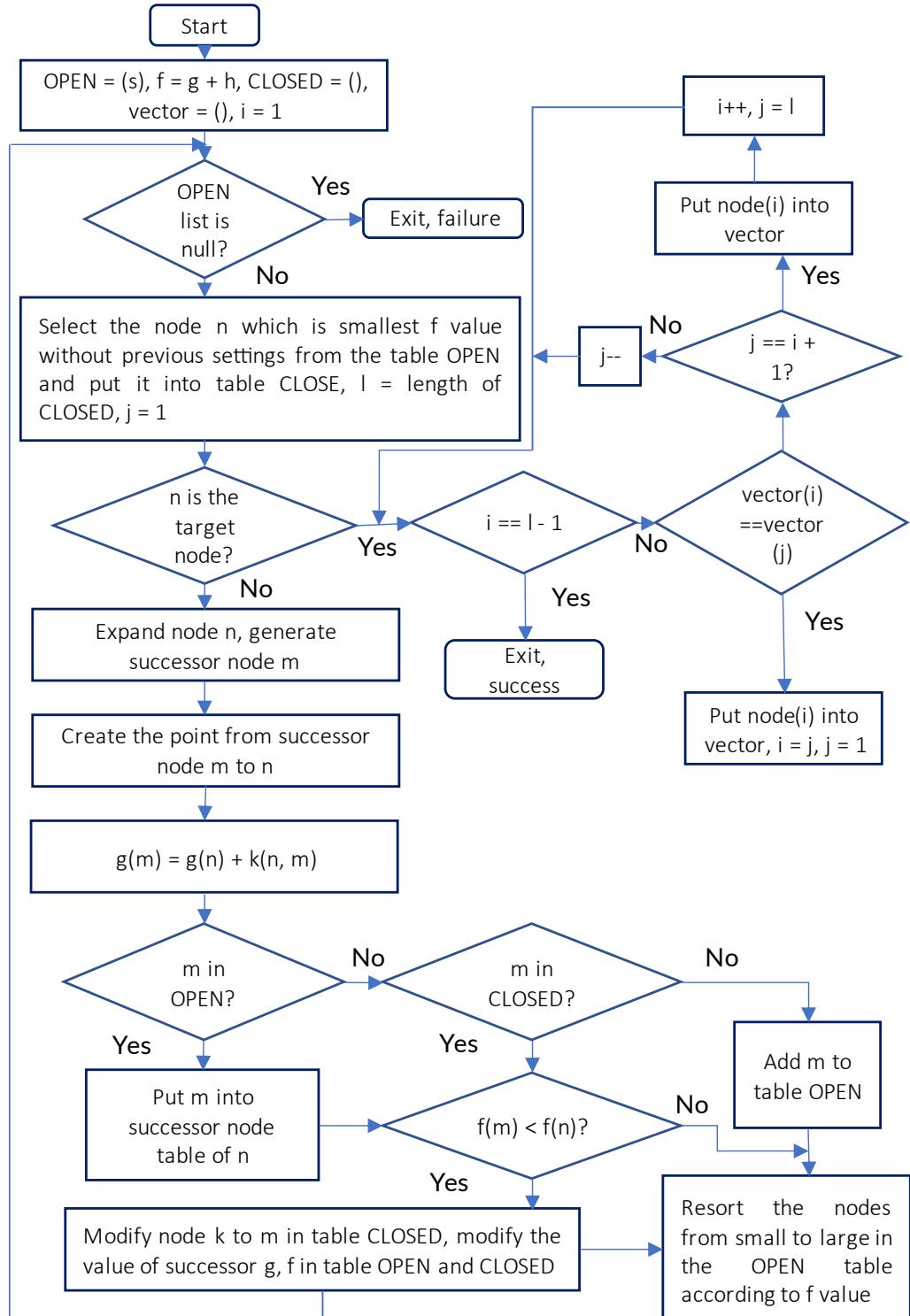
Nhóm sẽ thực khảo sát khu vực thực nghiệm xe, khu vực phải đủ thoáng đáng và ít xe qua lại để đảm bảo sự an toàn khi thực nghiệm. Tiếp theo là thu thập hình ảnh của khu vực đó thông qua Google Maps. Thực hiện một công cụ giúp đọc được hình ảnh, hỗ trợ vẽ đường đi hay các đặc trưng cần thiết, có thể thay đổi được giá trị các pixel hình ảnh để phân biệt giữa đường đi và các khu không thể đi. Cuối cùng là áp dụng một số lệnh xử lý ảnh để đạt được lớp bản đồ mong muốn mà hỗ trợ tối ưu nhất cho các thuật toán hoạt động trên nó.

5.3.2. Hiển thị bản đồ

Framework Qt hỗ trợ mô-đun *QgraphicsScene* giúp hiển thị và thao tác dễ dàng trên các hình ảnh. Nó giúp hiển thị bản đồ đã thực, hiển thị vị trí hiện tại của xe, đường đi A * khi đã hoạch định thành công.

5.4. Thực hiện thuật toán A *

Thuật toán A * sẽ được nhúng trực tiếp vào giao diện hiển thị. Nhóm tự thực hiện lập trình thuật toán dựa trên giải thuật của một bài báo khoa học đã cho kết quả phù hợp với mục đích đề tài của nhóm. Dưới đây là sơ đồ giải thuật của thuật toán.



Hình 5-7 Quá trình tìm kiếm của thuật toán A * [11]

Sơ đồ trên là giải thuật tìm đường đi ngắn nhất của thuật toán A *. Như đã trình bày ở phần lý thuyết, mục tiêu của thuật toán là tìm được đường đi

ngắn nhất giữa hai điểm dựa trên việc xác định giá trị nhỏ nhất của hàm $f(n) = g(n) + h(n)$. Chi tiết thực hiện như sau [12]:

- i. Đẩy nút ban đầu s vào bảng OPEN, $\text{OPEN} = (s)$. $\text{CLOSED} = ()$, tại thời điểm này, bảng CLOSED là một bảng trống. Do đó, hàm tại vị trí s sẽ được biểu diễn $f(s) = 0 + h(s)$ có.
- ii. Lặp lại quá trình sau cho đến khi tìm ra nút đích. Nếu bảng OPEN trống, nghĩa là $\text{OPEN} = ()$, thì thoát, tức là không tìm ra đích.
- iii. Để chọn và đánh dấu nút có giá trị f nhỏ nhất mà không cần thiết lập trước đó khỏi bảng OPEN, hãy xóa nút đó khỏi bảng OPEN và đưa vào bảng CLOSED.
- iv. Nếu nút tốt nhất là nút đích, nghĩa là đã đến đích, quá trình tìm kiếm đã thu được giải pháp thành công và thuật toán kết thúc. Đường dẫn tốt nhất từ nút ban đầu đến nút n được nhận bằng cách theo dõi con trỏ của chuỗi chính.
- v. Nút n sẽ được mở rộng nếu nút n tốt nhất không phải là nút đích. Tập nút $M = (m)$ được tạo ra bao gồm tất cả các nút kế thừa của nút n mà không phải là tổ tiên của nút n. Nút m được thêm vào như một nút kế thừa của nút n vào đồ thị tìm kiếm G.
- vi. Nút m được thêm vào bảng OPEN nếu m không xuất hiện trong danh sách OPEN hoặc trong bảng CLOSED.
- vii. Nút k bị xóa khỏi bảng OPEN và nút m được thêm vào bảng OPEN nếu m có một nút trùng lặp k trong bảng OPEN và $f(m) < f(k)$.
- viii. Kiểm tra nút kế nhiệm có trong bảng CLOSED hay không. Nếu nó không có trong bảng OPEN, các hoạt động như sau được thực hiện nếu nút kế nhiệm có nút trùng lặp trong bảng CLOSED.
 - a. Thay đổi nút k thành nút m trong bảng CLOSED.
 - b. Sửa đổi giá trị g, f kế vị trong bảng OPEN và CLOSED có chứa nút k trong chuỗi các phần tử kế tiếp.
- ix. Sắp xếp lại các nút từ nhỏ đến lớn trong bảng OPEN theo giá trị của f .

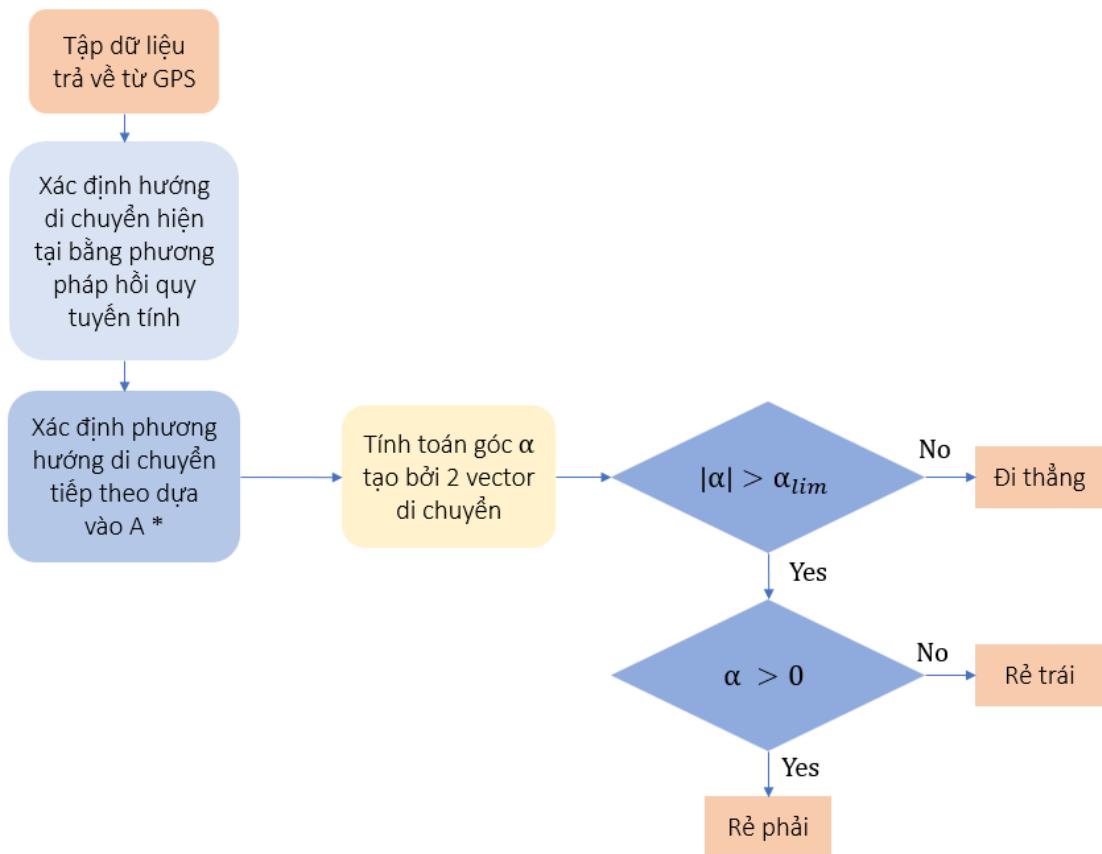
x. Trở lại bước *ii* bắt đầu lại vòng lặp.

Thuật toán để trích xuất một đường dẫn khả thi từ bảng CLOSED được mô tả theo các bước sau [13]:

- i.** Đặt nhãn nút cuối $i = 1$, độ dài n của bảng CLOSED được giải quyết, để làm cho nhãn nút $j = n$, tạo một vectơ trống.
- ii.** Nếu $i = n - 1$, chuyển sang **vi**. Nếu không, so sánh nút i với nút j . Nếu i và j khác nhau, chuyển sang **iii**. Nếu i và j giống nhau, chuyển sang **iv**.
- iii.** Nếu $j = i - 1$, thì trích xuất nút i , đẩy vào vectơ và chuyển sang **v**, nếu không thì tính $j--$, chuyển sang **ii**.
- iv.** Trích xuất nút i và đẩy vào vectơ, $i = j, j = n$, để chuyển sang **ii**.
- v.** Tính $i++$, gán $j = n$, chuyển sang **ii**.

Khi quá trình trích xuất nút hoàn thành, chuỗi nút được lưu trữ trong vectơ sẽ là đường dẫn khả thi.

5.5. Phương pháp hồi quy tuyến tính



Hình 5-8 Sơ đồ giải thuật điều hướng theo quỹ đạo đã hoạch định

Các trường hợp điều hướng cho xe lấy góc α tạo bởi vector vận tốc và vector chỉ điều hướng để làm cơ sở.

Do sai số vị trí trả về từ GPS là khá lớn nên không thể chỉ dựa vào một vài điểm mà xác định được xu hướng chuyển động hay vị trí chính xác của xe. Vì vậy ta cần thu thập và lưu giữ các dữ liệu của vị trí trước đó, sau đó dùng thuật toán hồi quy tuyến tính để dựng một đường thẳng tốt nhất dựa trên tập dữ liệu đó, mà đặc trưng của nó chính là hệ số góc. Hệ số góc giúp đánh giá được phương hướng chuyển động hiện tại của xe.

Phương trình xấp xỉ tập dữ liệu có dạng [14]

$$y = \hat{\alpha} + \hat{\beta}x$$

$$\hat{\beta} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$\hat{\alpha} = \bar{y} - (\hat{\beta}\bar{x})$$

trong đó,

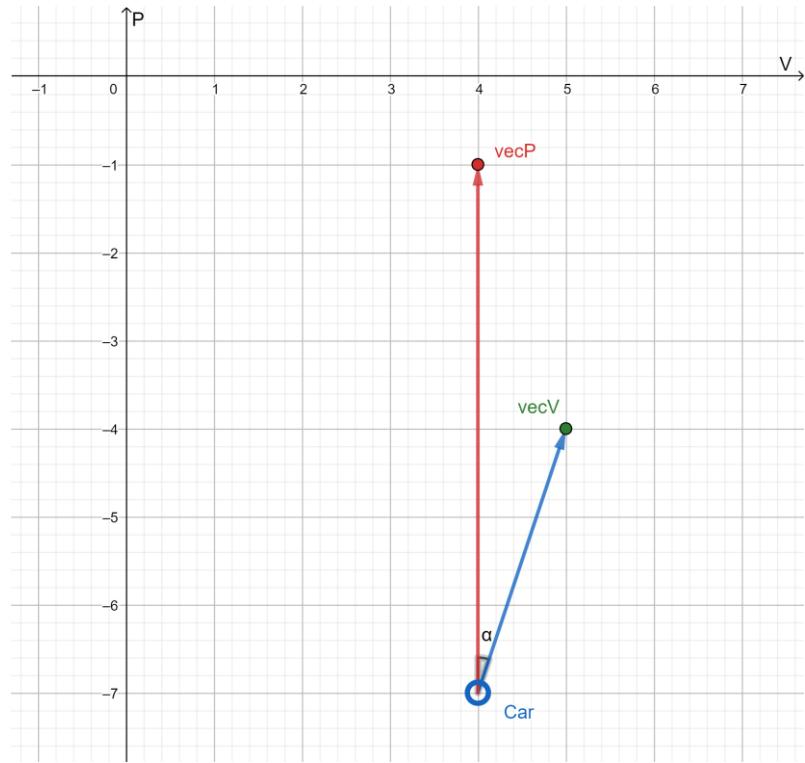
\bar{x}, \bar{y} là giá trị trung bình của x_i, y_i tương ứng

$\hat{\beta}$ là hệ số góc

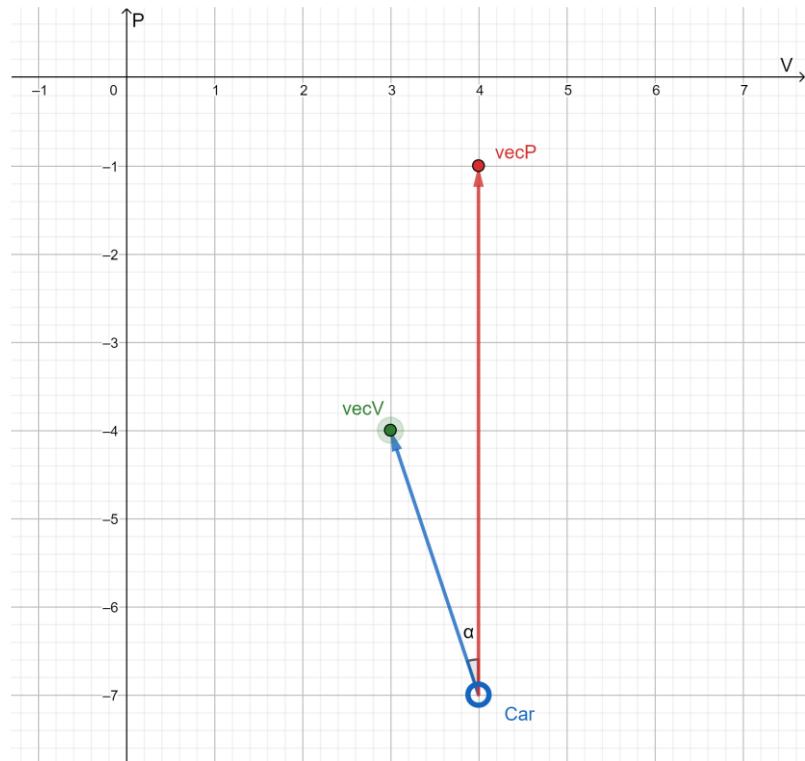
$\hat{\alpha}$ là giá trị cắt tại trục y

Như vậy dựa vào hệ số góc $\hat{\beta}$ mà ta tính toán được sẽ xác định được phương di chuyển của xe, đồng thời dựa vào các giá trị tọa độ lưu trữ trước đó đến thời điểm hiện tại ta sẽ xác định được chuyển động của phương tiện.

Các trường hợp điều hướng cho xe: góc α tạo bởi vector vận tốc (hướng sẽ được xác định bởi tập dữ liệu trả về của GPS, độ lớn hiện tại của vector vận tốc được tính toán dựa vào encoder) và vector điều hướng (vector điều hướng được xác định dựa vào vị trí mới nhất và vị trí thuộc cần đến cách cách đó một khoảng có thể cấu hình được và vị trí đó thuộc quỹ đạo A *).



Hình 5-9 Góc tạo bởi vector chuyển động và vector điều hướng $\alpha < 0$



Hình 5-10 Góc tạo bởi vector chuyển động và vector điều hướng $\alpha > 0$

Việc xuất tín hiệu điều hướng sẽ phụ thuộc vào độ lớn và giá trị của góc α , ngưỡng độ lớn của góc α dùng để quyết định ra tín hiệu cần rẽ hướng hay đi

thẳng sẽ được điều chỉnh cho phù hợp trong quá trình thực nghiệm; khi đáp ứng được điều kiện rẽ, giá trị của góc α sẽ quyết định hướng rẽ là trái hay phải. Độ lớn góc α được tính bằng hàm $\cos^{-1}()$ giữa hai vector, giá trị của góc α được tính bằng hàm $\sin^{-1}()$. Như vậy, điều kiện để xe xuất tín hiệu điều hướng như sau:

$$\begin{cases} \alpha > 0 \text{ và } |\alpha| > \alpha_{lim}, & (\text{rẽ phải}) \\ \alpha < 0 \text{ và } |\alpha| > \alpha_{lim}, & (\text{rẽ trái}) \\ |\alpha| < \alpha_{lim}, & (\text{đi thẳng}) \end{cases}$$

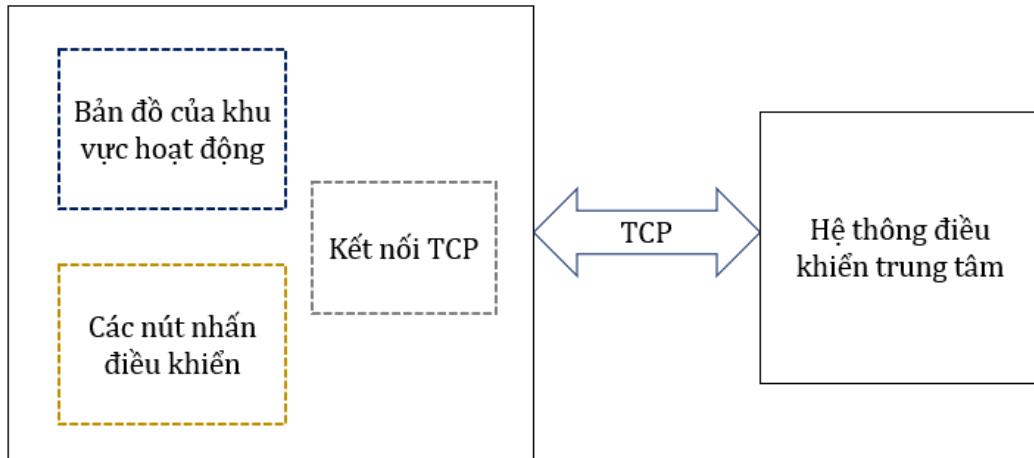
Với phương tiện xe điện tự hành mà nhóm đang thực hiện, bộ điều khiển trung tâm MPC sẽ điều khiển góc đánh lái của xe bảo đảm xe đi đúng làn đường và rẽ đúng hướng. Bộ điều khiển sẽ lấy đầu vào từ các cảm biến: camera, radar, GPS; tín hiệu đầu ra của các cảm biến này sẽ được đánh trọng số và làm đầu vào cho bộ MPC từ đó tính toán được góc đánh lái cho xe. Vì vậy, tín hiệu điều hướng này sẽ làm đầu vào cho bộ điều khiển trung tâm của xe chính là bộ MPC, việc cải thiện được tín hiệu điều khiển này sẽ góp phần quan trọng cho việc xác định phương hướng di chuyển hiện tại của xe và ra các lệnh rẽ hướng hay đi thẳng cho quá trình chuyển động tiếp theo để đến đích.

5.6. Thiết kế giao diện điều khiển

5.6.1. Tổng quan giao diện điều khiển

Giao diện sẽ hiển thị:

- Bản đồ mà khu vực xe đang hoạt động.
- Vị trí hiện tại, đường đi và trạng thái của xe.
- Các nút nhấn giúp ta có thể tự điều khiển xe trong quá trình thực nghiệm.
- Giao diện sẽ được kết nối với hệ thống điều khiển trung tâm trong mạng LAN cục bộ thông qua giao thức TCP/IP.



Hình 5-11 Tổng quan giao diện người dùng

Nhóm đã chọn Framework Qt Creator C++ để thực hiện thiết kế giao diện. Đây là một framework hỗ trợ mạnh mẽ không chỉ xây dựng giao diện chuyên nghiệp trong ngành công nghiệp thiết kế giao diện xe ô tô nói chung và xe điện tự hành nói riêng. Qt cũng hỗ trợ xây dựng phát triển thuật toán với ngôn ngữ C++, Python, hay các mô-đun giúp truy cập lên sever để ứng dụng cho dự án muốn lưu trữ dữ liệu hay quan sát điều khiển từ xa.

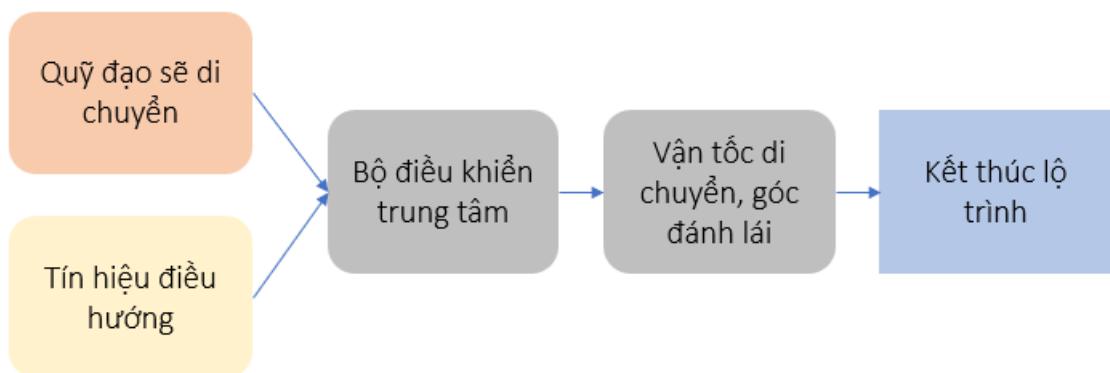
5.6.2. Thiết kế các tính năng điều khiển xe

Ngoài việc hiển thị bản đồ giúp quan sát được vị trí hiện tại của xe, giao diện sẽ có các tính năng cơ bản để có con người có thể điều khiển xe khi thực nghiệm. Các nút nhấn bắt đầu chạy và dừng lại, nút nhấn chuyển chế độ chạy tự động hay con người sẽ điều khiển. Các nút dùng để bật tắt đèn hay còi xe, các thanh trượt giúp điều khiển vận tốc xe, điều khiển góc đánh lái của vô-lăng. Thao tác chọn điểm bắt đầu vào điểm kết thúc của quá trình di chuyển. Nút nhấn bắt đầu thực hiện thuật toán A * khi đã chọn hai điểm và cho phép hiển thị đường đi khi đã tìm kiếm đường đi thành công.

Như vậy để các tín hiệu đó được gửi đến bộ điều khiển trung tâm Qt có hỗ trợ mô-đun *QtcpSocket*. Khi các máy tính được kết nối chung mạng LAN

quông qua router, các tín hiệu điều khiển từ giao diện sẽ được gửi đến trung tâm thông qua giao thức TCP.

5.7. Ứng dụng hệ thống lên xe điện tự hành



Hình 5-12 Ứng dụng hoạch định quỹ đạo và phương pháp điều hướng lên xe tự hành

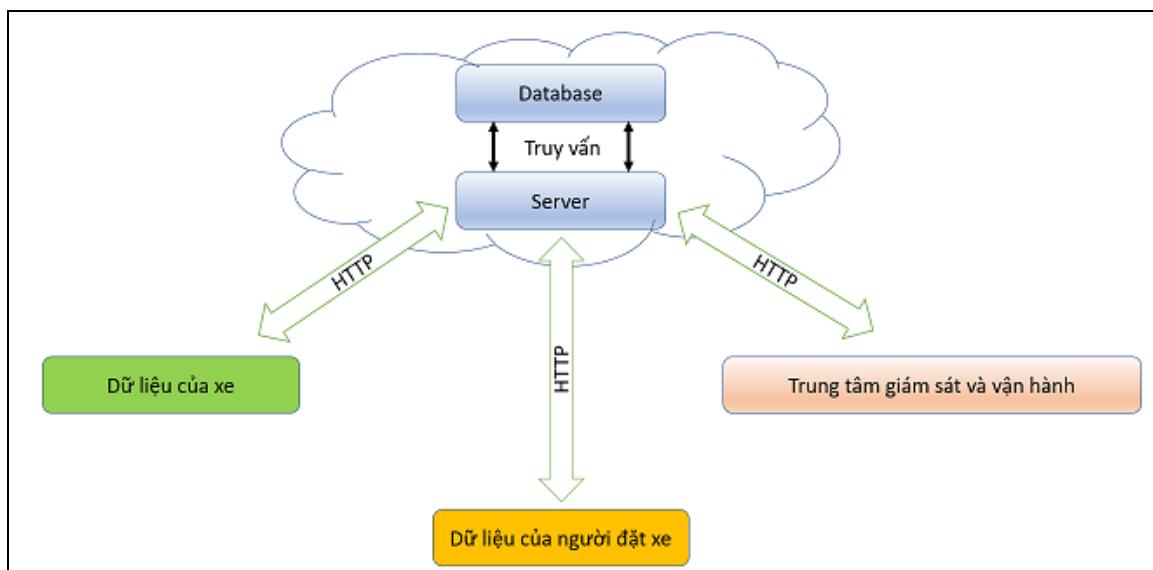
Với giao diện điều khiển đặt trên xe, người dùng thực hiện chọn điểm đầu cuối của quá trình di chuyển trên bản đồ; trong đó điểm đầu ứng với vị trí hiện tại, điểm cuối là vị trí muốn đến. Quỹ đạo di chuyển A * giữa hai điểm sẽ được thiết lập và hệ thống xuất tín hiệu điều khiển bắt đầu quá trình di chuyển. Tín hiệu GPS được cập nhật liên tục và hiển thị trên bản đồ của giao diện, tập dữ liệu vị trí sẽ lưu từ 10 đến 20 mẫu tọa độ di chuyển, các mẫu trước đó sẽ liên tục được loại bỏ và liên tục cập nhật vị trí mới nhất. Dựa trên tập dữ liệu đó sẽ xác định được hướng xe đang di chuyển, dựa vào vị trí mới nhất và vị trí cần đến (thuộc quỹ đạo A *) xác định hướng di chuyển tiếp theo từ đó xuất tín hiệu điều hướng chuyển động cho xe. Tín hiệu đến bộ điều khiển trung tâm, tốc độ sẽ được điều khiển bởi bộ PID vòng kín với hồi tiếp vận tốc hiện tại được tính toán dựa vào encoder. Thuật toán phát hiện làn đường sẽ làm đầu vào cho bộ điều khiển MPC giúp điều khiển góc đánh lái đảm bảo rằng xe đi đúng làn đường quy định; tín hiệu điều hướng là một trong những đầu vào của bộ MPC giúp bộ điều khiển quyết định giữ góc đánh lái đi thẳng, đánh lái rẽ trái hay phải tại các khu vực ngã ba, ngã tư. Cuối cùng, khi xe tiếp cận vị trí đích đến

trong tầm bán kính 2m đến 3m, hệ thống thông báo xe đã đến vị trí mong muốn và xuất tín hiệu điều khiển cho xe dừng lại.

Chương 6. THIẾT KẾ HỆ THỐNG WEB SERVER GIÁM SÁT VÀ VẬN HÀNH HOẠT ĐỘNG XE ĐIỆN TỰ HÀNH

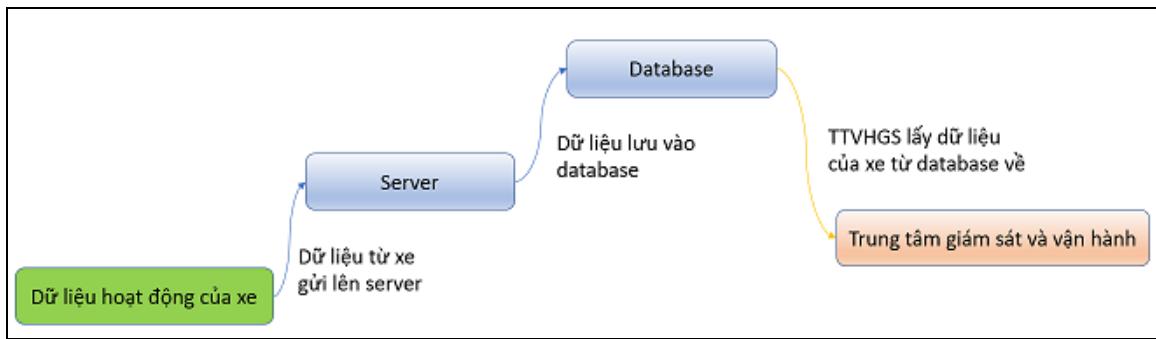
6.1. Tổng quan về hệ thống giám sát và vận hành hoạt động xe tự hành

Hệ thống giám sát và vận hành bao gồm một hệ thống kết nối dữ liệu qua mạng ở đây là mô hình truyền nhận dữ liệu giữa xe tự hành với trung tâm vận hành theo giao thức HTTP.



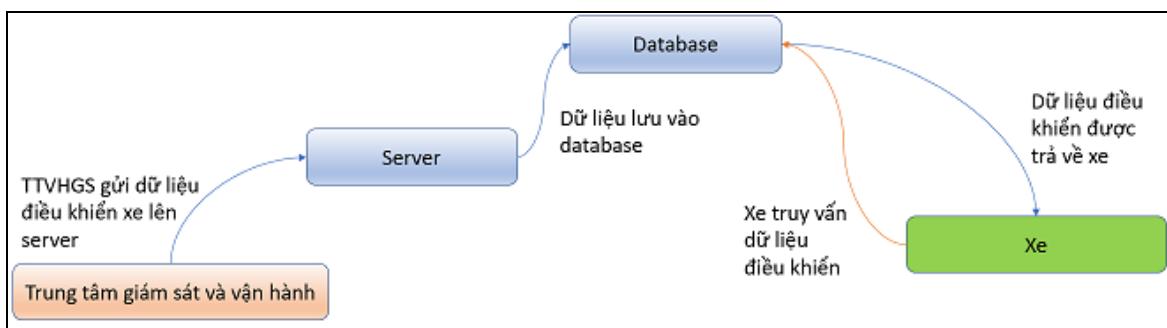
Hình 6-1 Mô hình hệ thống giám sát và vận hành hoạt động của xe (đội xe) tự hành

Dữ liệu của xe (vị trí, trạng thái, tốc độ, lượng pin) được gửi lên server, sau đó các thông tin được lưu trữ vào cơ sở dữ liệu. Với các thông tin này, trung tâm giám sát có thể yêu cầu dữ liệu từ server về để có thể theo dõi hoạt động của các xe từ dữ liệu đã được lưu trữ trên CSDL. Trung tâm giám sát và vận hành cũng có thể gửi các yêu cầu đến xe theo chiều ngược lại: yêu cầu được gửi lên server và lưu vào CSDL, sau đó xe sẽ yêu cầu đến server để truy xuất các dữ liệu này để thực hiện nhiệm vụ. Một luồng dữ liệu khác là khách hàng là người đặt xe với các thông tin về điểm đón, điểm đến của chuyến đi, thông tin này được gửi về server và lưu vào CSDL để các xe truy xuất đến. Các luồng dữ liệu được thể hiện trong các lưu đồ giải thuật dưới đây.



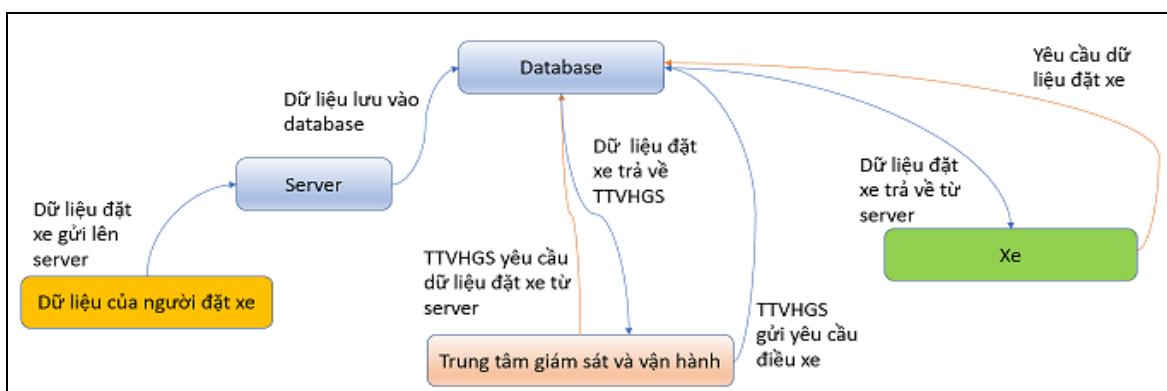
Hình 6-2 Dữ liệu từ xe gửi lên server và trung tâm giám sát lấy dữ liệu về

Dữ liệu hoạt động của xe được gửi về server và lưu vào CSDL. Trung tâm giám sát và vận hành yêu cầu dữ liệu hoạt động của xe từ server, server trả về các thông tin hoạt động của xe đã được lưu trên CSDL.



Hình 6-3 Dữ liệu điều khiển gửi đến xe

Trung tâm vận hành gửi yêu cầu điều khiển xe, yêu cầu đã được đưa lên server và lưu vào CSDL. Xe thực hiện truy vấn đến dữ liệu điều khiển, server trả về thông tin điều khiển đã được lưu trên CSDL của xe được gửi về server và lưu vào CSDL. Trung tâm giám sát và vận hành yêu cầu dữ liệu hoạt động của xe từ server, server trả về các thông tin hoạt động của xe đã được lưu trên CSDL.



Hình 6-4 Dữ liệu đặt xe với điểm đến cuối cùng là yêu cầu xe thực hiện lộ trình được đặt

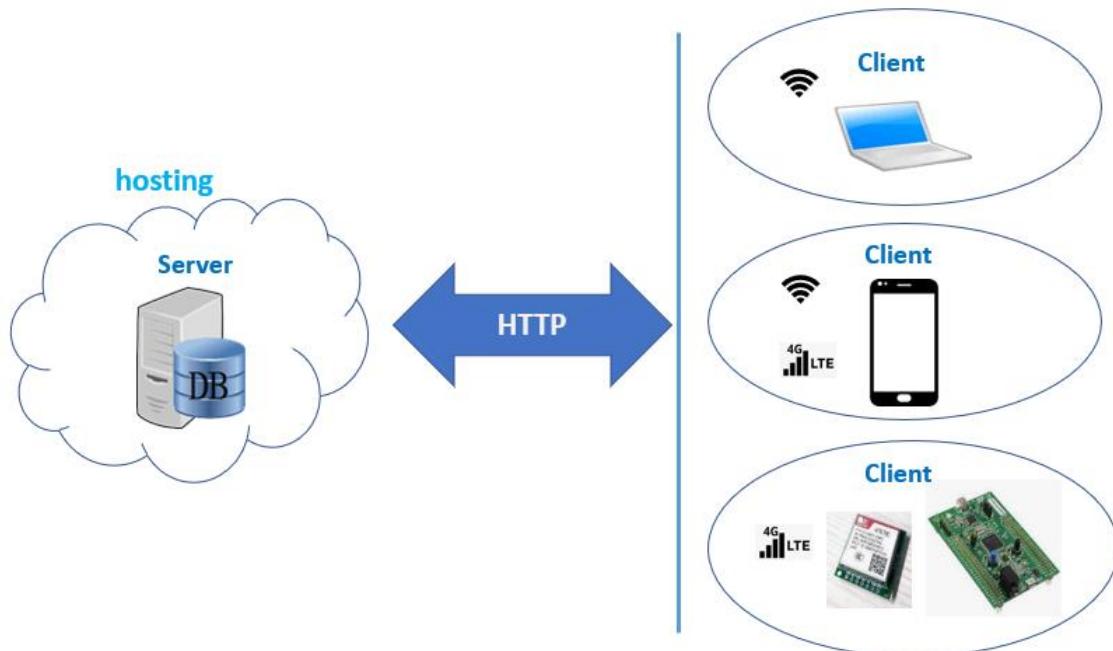
Người đặt xe nhấn thực hiện đặt xe, các thông tin đặt xe được gửi về server và lưu vào CSDL. TTGSVH truy vấn các thông tin đặt xe từ server, server trả về các thông tin đặt xe đã được lưu trên CSDL. TTGSVH xử lý và chọn xe thực hiện, sau đó gửi yêu cầu điều xe (đặt xe) lên server. Xe thực hiện truy vấn thông tin điều xe (đặt xe) từ server, server trả về thông tin điều xe (đặt xe) đã lưu trên CSDL, xe thực hiện yêu cầu.

6.2. Xây dựng Server

Để các phương tiện có thể kết nối đến server mọi lúc, mọi nơi chỉ cần kết nối internet, thì trước hết phải có một máy chủ (server) trên internet hoạt động liên tục, các dữ liệu sẽ được đặt trên máy chủ đó để các client có thể truy cập mọi lúc. Ở đây, nhóm chúng em sử dụng máy chủ do 000webhost cung cấp. 000webhost là đơn vị cung cấp hosting miễn phí, thuộc sự quản lý của Hostinger. 000webhost có rất nhiều gói hosting với những tính năng cũng như chi phí khác nhau phù hợp với từng dự án. Với gói dùng thử (Free Web Hosting) của 000webhost sẽ có các thông số quan trọng sau:

- Dung lượng ổ đĩa: 300MB.
- Băng thông: 3GB.
- Số lượng website: 1 với tên miền bị giới hạn theo cấu trúc: “websitename.000webhostapp.com”.
- Cơ sở dữ liệu MySQL: 2 với dung lượng bộ nhớ tối đa của mỗi cơ sở dữ liệu là 1 GB.

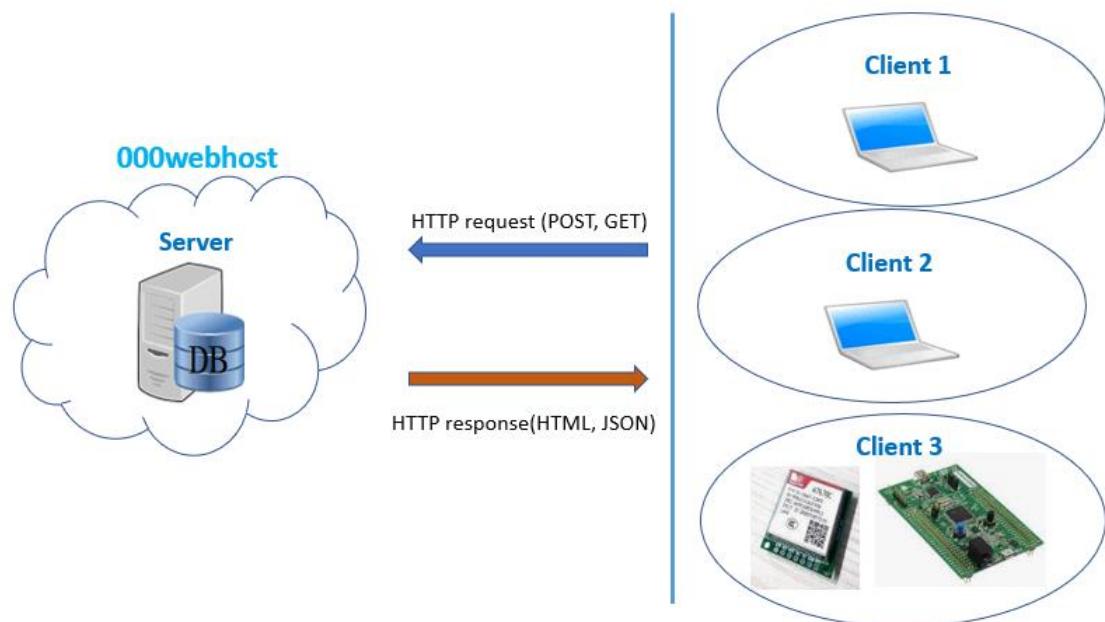
Với những tính năng mà gói Free Web Hosting cung cấp sẽ là đủ đối với các ứng dụng có lượt truy cập đến server vừa phải, như đồ án đại học, còn với các ứng dụng yêu cầu cao hơn, chúng ta sẽ phải trả một khoản phí tương ứng tùy theo yêu cầu dịch vụ.



Hình 6-5 Mô hình kết nối hệ thống dữ liệu qua mạng

Cách thức hoạt động

Các client ở đây là vi điều khiển được kết nối internet qua mạng 4G hay các trình duyệt web, khi các Client gửi một HTTP request đến server thì server sẽ gửi về một response tương ứng.

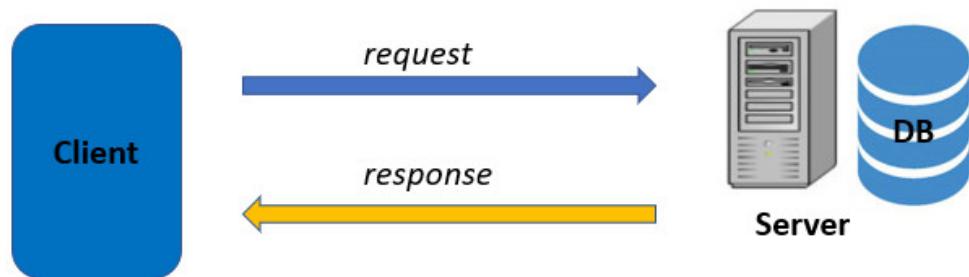


Hình 6-6 Cách thức trao đổi dữ liệu

6.3. Thiết kế cơ sở dữ liệu

6.3.1. Tổng quan cơ sở dữ liệu

000webhost hỗ trợ cơ sở dữ liệu MySQL. MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.



Hình 6-7 Cách thức truy cập MySQL

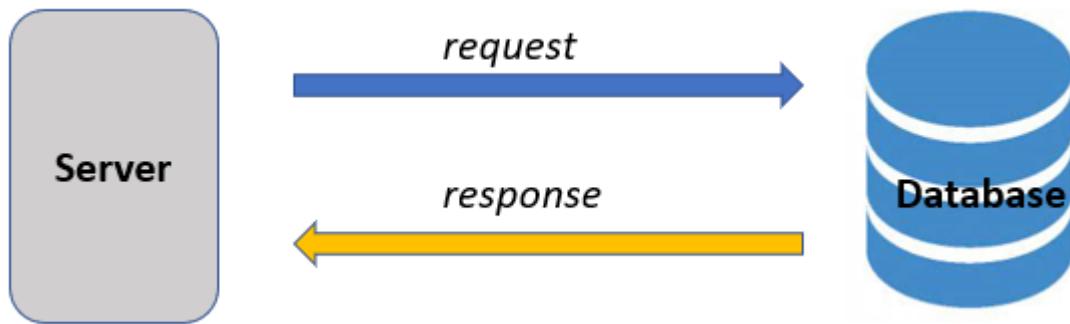
Hình ảnh trên giải thích cấu trúc cơ bản về việc giao tiếp giữa client-server model. Một máy client sẽ liên lạc với máy server trong một mạng nhất định. Mỗi client có thể gửi một request từ giao diện người dùng (Graphical user interface – GUI) trên màn hình, và server sẽ trả về kết quả như mong muốn. Miễn là cả hai hiểu nhau. Cách vận hành chính trong môi trường MySQL cũng như vậy:

- MySQL tạo ra bảng để lưu trữ dữ liệu, định nghĩa sự liên quan giữa các bảng đó.
- Client sẽ gửi yêu cầu SQL bằng một lệnh đặc biệt trên MySQL.
- Ứng dụng trên server sẽ phản hồi thông tin và trả về kết quả trên máy client.

Máy chủ do 000webhost cung cấp cơ sở dữ liệu với bộ nhớ 1GB (đối với gói Free Web Hosting) sử dụng phpMyAdmin để thực hiện quản trị cơ sở dữ

liệu. Do đẽ tài sủ dụng PHP đẽ thực hiện phần back-end cho nên sủ dụng phpMyAdmin cũng là một điểm vô cùng phù hợp.

Server trên máy chủ trên mạng khi thực hiện truy vấn đến cơ sở dữ liệu cũng sẽ đóng vai trò như một client yêu cầu dữ liệu từ database.



Hình 6-8 Server đóng vai trò là client khi truy cập vào database

6.3.2. Cấu trúc cơ sở dữ liệu

Bảng 6-1 Cấu trúc cơ sở dữ liệu tổng

Bảng	Hành động		Hàng	Kiểu	Bảng mã đổi chiều	Kích thước
booked_info	★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa		6	InnoDB	utf8_unicode_ci	16,0 KiB
fleet_info	★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa		2	InnoDB	utf8_unicode_ci	32,0 KiB
gps_data_list	★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa		56	InnoDB	utf8mb4_general_ci	16,0 KiB
task_control	★ Duyệt Cấu trúc Tìm kiếm Chèn Rỗng Xóa		2	InnoDB	utf8_unicode_ci	16,0 KiB
4 bảng	Tổng		66	InnoDB	utf8_unicode_ci	80,0 KiB

Cơ sở dữ liệu bao gồm 4 bảng dữ liệu cơ sở: *booked_info*, *fleet_info*, *gps_data_info*, *task_control*. Mỗi bảng đều có dữ liệu về *timestamp* là thời gian cập nhật dữ liệu.

Bảng 6-2 Bảng *fleet_info* lưu các thông số hoạt động của xe

car_ID	car_status	car_speed	car_position	car_task	car_power
1	ON	15	10.45124, 106.76541	Test Updata	80
2	ON	13	10.45221, 106.76536	Delivery	65

Bảng *fleet_info* lưu trữ thông tin về các trạng thái hoạt động của từng xe: kí hiệu của xe (car_ID), vị trí (car_position), tốc độ (car_speed), lượng pin còn lại (car_power), nhiệm vụ đang thực hiện (car_task).

Bảng 6-3 Bảng *gps_data_info* lưu trữ dữ liệu vị trí của xe

frame_ID	car_gps_ID	latitude	longitude	velocity	timestamp
1	2	1045.57040	10639.64824	0	2022-05-19 18:25:33
2	2	1045.56949	10639.64343	0	2022-05-19 18:25:37

Bảng *gps_data_info* chịu trách nhiệm chính cho dữ liệu về vị trí của mỗi xe gồm kinh độ (longitude) và vĩ độ (latitude) và đi kèm theo tốc độ của các xe (velocity) phân biệt với nhau bởi kí hiệu của xe (car_gps_ID).

Bảng 6-4 Bảng *booked_info* lưu trữ thông tin đặt xe

guest_ID	guest_num	lat_a	lon_a	lat_b	lon_b	timestamp
8	0384370746	10.79745	106.74354	10.79758	106.74502	2022-05-19 18:38:27
9	0384370746	10.79790	106.74331	10.79812	106.74478	2022-05-19 18:38:27

Bảng *booked_info* lưu trữ dữ liệu đặt xe của khách hàng, bao gồm thứ tự đặt (guest_ID), số điện thoại (guest_num), điểm đón gồm vĩ độ và kinh độ (lat_a, lon_a), điểm đến (lat-b, lon_b).

Bảng 6-5 Bảng *task_control* lưu trữ thông tin điều xe

car_do_ID	guest_num	lat_a_do	lon_a_do	lat_b_do	lon_b_do
2	038370746	10.26354	106.76521	10.26187	106.76318

Bảng *task_control* lưu trữ dữ liệu về nhiệm vụ giao cho các xe sau khi trung tâm vận hành tính toán để lựa chọn xe tối ưu thực hiện yêu cầu của khách hàng. Bảng gồm các thông tin về kí hiệu xe sẽ thực hiện nhiệm vụ (car_do_ID), số điện thoại khách hàng (guest_num), và lộ trình gồm điểm đón (lat_a_do, lon_a_do), điểm đến (lat_b_do, lon_b_do).

Khi client gửi các yêu cầu truy vấn dữ liệu đến server, server sẽ kiểm tra các truy vấn này theo các \$key để tìm kiếm đến dữ liệu mong muốn đó trên CSDL.

6.4. Thiết kế giao diện web

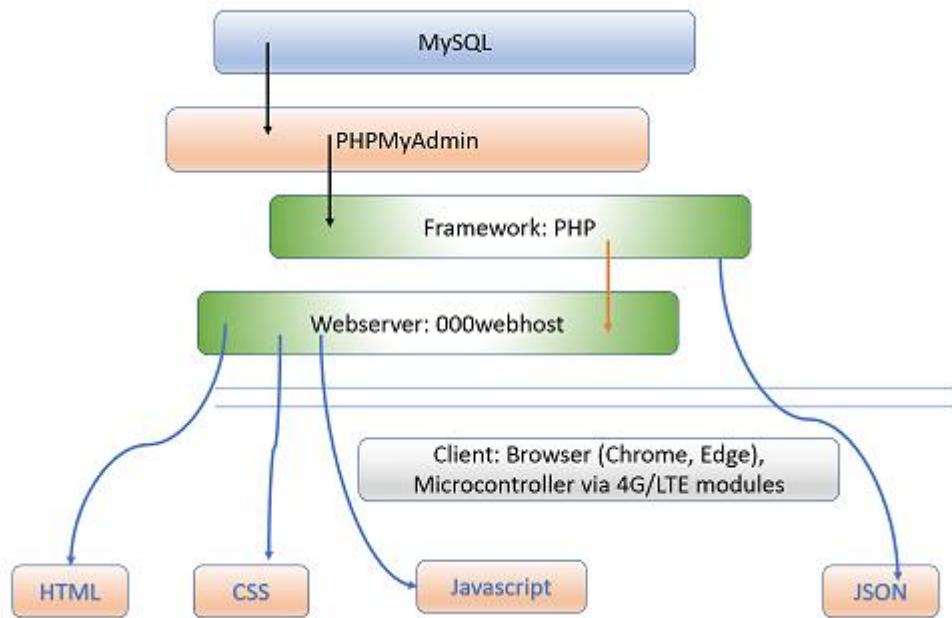
Về phần thiết kế giao diện, sẽ gồm hai giao diện web chính: giao diện cho người giám sát ở trung tâm giám sát và vận hành, và giao diện dành cho người đặt xe.

Trên giao diện giám sát sẽ có thể thực hiện các chức năng sau:

- Giám sát dữ liệu về hoạt động của xe (đội xe): vị trí của từng xe, trạng thái hoạt động, lộ trình đang đi, lượng pin còn lại của mỗi xe...
- Lưu trữ dữ liệu về danh sách khách hàng đặt xe, sau đó xử lý dữ liệu để chọn xe phù hợp nhất để thực hiện nhiệm vụ.
- Có khả năng gửi yêu cầu cụ thể cho từng xe: yêu cầu bật/tắt xe, yêu cầu sạc pin...

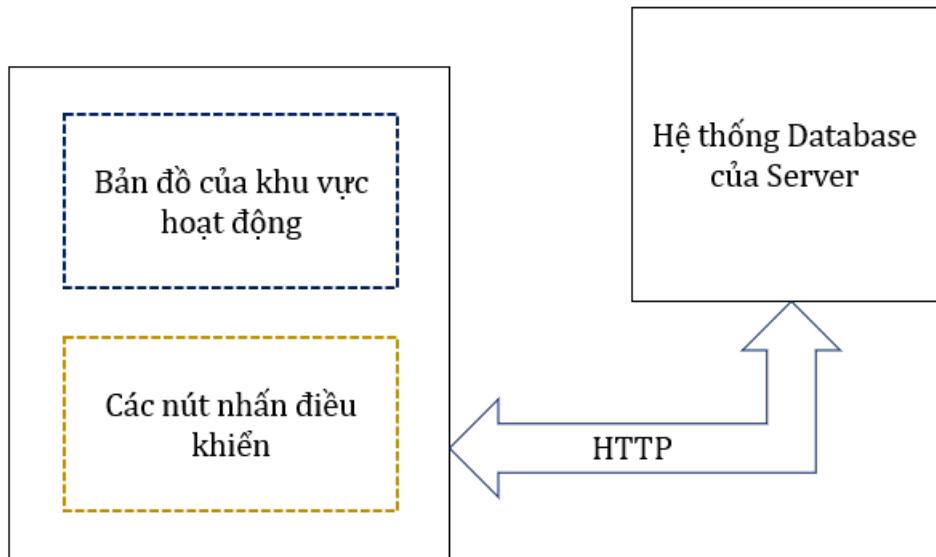
Trên giao diện đặt xe sẽ có các chức năng sau:

- Nhập thông tin đặt xe và thực hiện đặt xe.
- Chọn lộ trình mong muốn trên bản đồ và thực hiện đặt xe.
- Theo dõi vị trí hiện tại của bản thân.
- Theo dõi lộ trình đã đặt.
- Sử dụng các ngôn ngữ lập trình web HTML, CSS, Javascript để thiết kế giao diện front-end của trang web. Trên đó hiển thị đầy đủ các thông tin cần thiết cũng như những thao tác có thể thực hiện để tác động lên xe.
- Sử dụng PHP framework cho back-end để xử lý các luồng dữ liệu yêu cầu hay phản hồi từ client đến server và ngược lại



Hình 6-9 Cách thức hoạt động của PHP Framework

6.4.1. Thiết kế giao diện giám sát cho người vận hành

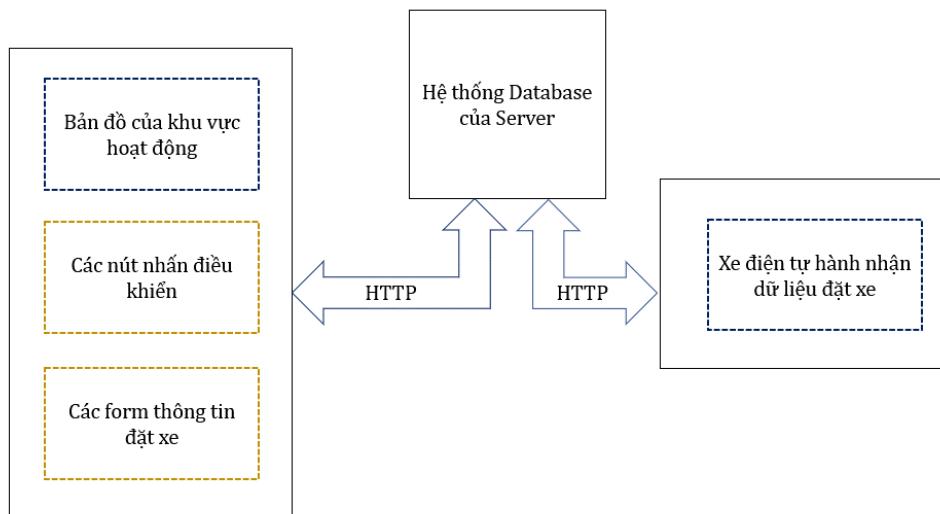


Hình 6-10 Giao diện giám sát từ xa thông qua giao thức HTTP

Hướng đến mục tiêu lâu dài là kết nối, giám sát, điều khiển từ xa cho đội xe tự hành trong khu vực cục bộ, nhóm đã thực hiện một giao diện giám sát và điều khiển từ xa. Ban đầu để dễ tiếp cận nhóm đã chọn giao thức HTTP. Dữ liệu

vị trí sẽ được gửi lên server thông qua mô-đun 4G được gắn trên xe. Qt cũng sẽ hỗ trợ mô-đun *QnetworkAccessManager* giúp truy cập đến cơ sở dữ liệu của server, lấy về vị trí hiện tại của phương tiện và hiển thị lên màn hình quan sát. Mục tiêu cao nhất là có thể điều khiển được xe bắt đầu chạy hay bắt buộc dừng lại khi phát hiện sự cố bất thường tại trung tâm quan sát. Tuy nhiên, trong quá trình thực nghiệm cho thấy độ trễ của giao thực HTTP quá cao, điều này không đảm bảo được tính an toàn khi quyết định điều khiển xe được đưa ra nên nhóm vẫn chưa thực hiện tính năng điều khiển mà chỉ dừng lại ở giám sát vị trí của phương tiện. Việc phát triển tính năng điều khiển sẽ phát triển thêm sau này khi tìm được giao thức phù hợp hơn.

6.4.2. Thiết kế giao diện đặt xe cho khách hàng

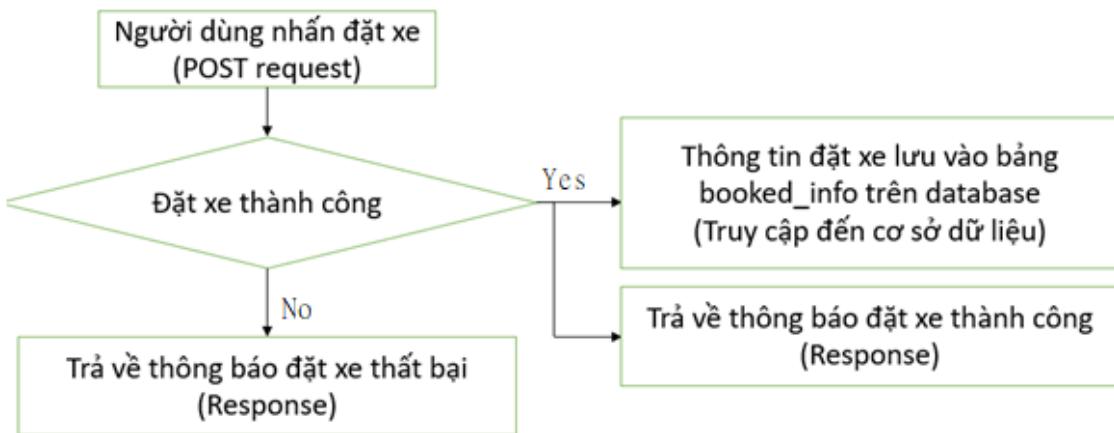


Hình 6-11 Giao diện đặt xe cho khách hàng

Trên giao diện web dành để đặt xe, khách hàng nhập các thông tin chuyến đi, sau đó dữ liệu được gửi về server và lưu vào CSDL. Các xe điện tự hành sẽ truy vấn đến CSDL để lấy dữ liệu đặt xe. Nếu dữ liệu đặt xe trùng với thông tin của xe (kí hiệu xe) thì xe thực hiện chuyến đi.

6.5. Lưu đồ giải thuật của hệ thống các luồng dữ liệu

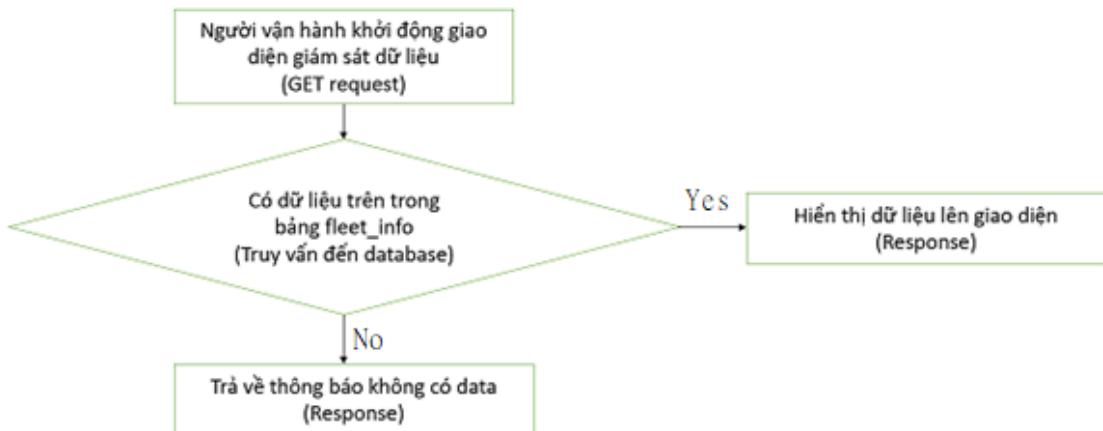
- ❖ *Client là người dùng thực hiện đặt xe*



Hình 6-12 Sơ đồ giải thuật khi Client là người dùng đặt xe

Khi người dùng truy cập đến trang web dành để đặt xe, sau khi người dùng nhập các thông tin cần thiết và nhấn nút “Đặt xe” một POST request được khởi tạo. Server sẽ xử lý request đó bằng cách kiểm tra các header, và các \$KEY, nếu đúng thủ tục thì dữ liệu đặt xe sẽ được lưu vào cơ sở dữ liệu và server trả về thông báo đặt xe thành công cho người đặt xe biết; ngược lại một thông báo đặt xe thất bại sẽ được trả về.

❖ *Client là người vận hành yêu cầu dữ liệu của đội xe*



Hình 6-13 Sơ đồ giải thuật khi Client là người vận hành yêu cầu dữ liệu của đội xe

Trên giao diện giám sát, khi người giám sát nhấn khởi động giám sát dữ liệu, một GET request được khởi tạo. Sau đó server sẽ kiểm tra request đó nếu đúng sẽ truy vấn đến cơ sở dữ liệu để xem có dữ liệu mà phía người vận hành

(client) yêu cầu hay không, nếu có thì hiển thị dữ liệu đó là các chấm đỏ lên trên giao diện; ngược lại trả về thông báo không có dữ liệu.

Từ thông tin về danh sách đặt xe, và dữ liệu của đội xe mà server sẽ tính toán để chọn ra xe và lộ trình để tối ưu việc thực hiện yêu cầu của khách hàng. Các thông tin này được lưu vào bảng task_control sau đó các xe sẽ truy vấn vào cơ sở dữ liệu này để thực hiện nhiệm vụ dựa vào hàm tối ưu thời gian.

$$t = \min \left(f \left(car_{status}, car_{position}, car_{task}, s(A - B) \right) \right)$$

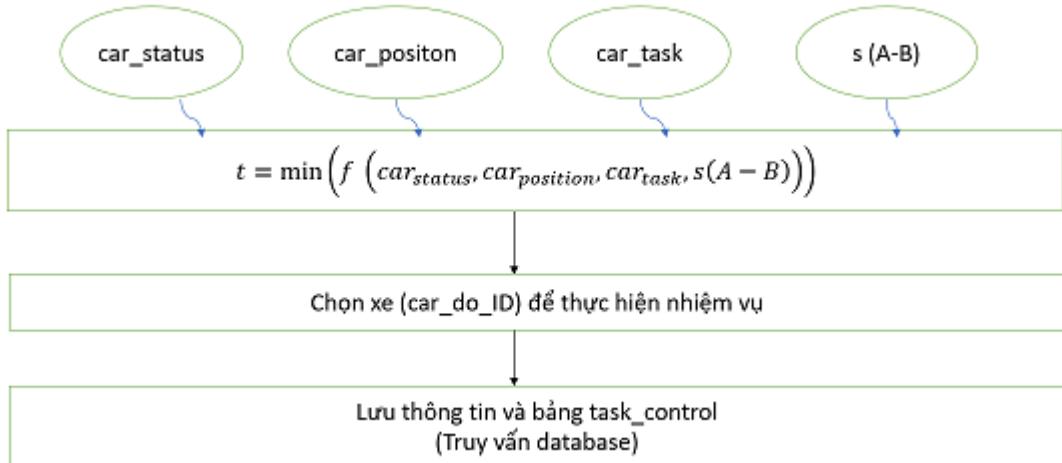
trong đó:

car_{status} : trạng thái hoạt động của các xe,

$car_{position}$: vị trí hiện tại của xe,

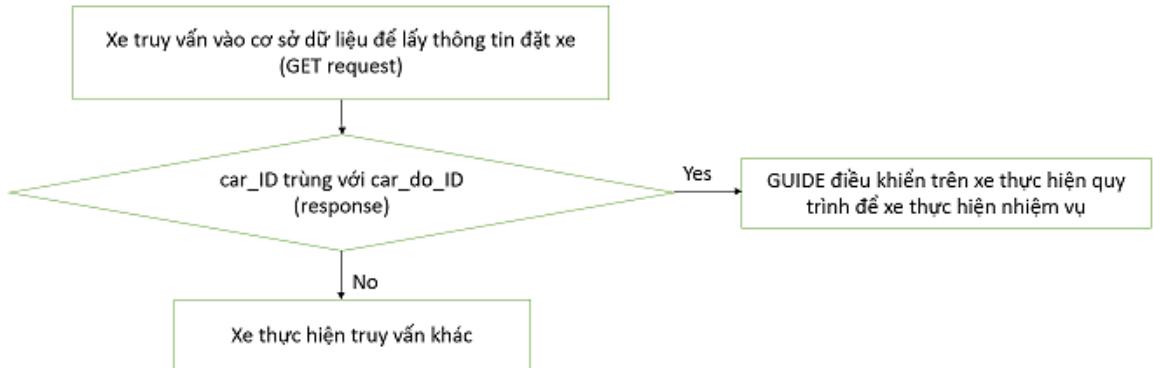
car_{task} : nhiệm vụ xe đang thực hiện (hoặc không có),

$s(A - B)$: quãng đường từ A đến B



Hình 6-14 Giải thuật chọn xe tối ưu thực hiện nhiệm vụ

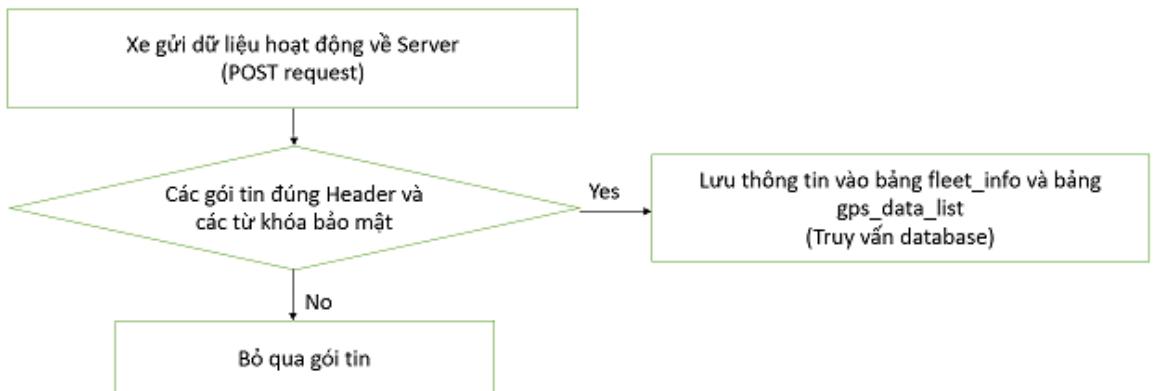
- ❖ Client là xe (vi điều khiển) yêu cầu thông tin đặt xe từ server qua mô-đun 4G



Hình 6-15 Sơ đồ giải thuật khi Client là xe (vi điều khiển) yêu cầu thông tin đặt xe từ server qua mô-đun 4G

Vi điều khiển thực hiện gửi các gói tin đến mô-đun 4G để thực hiện một GET request để lấy dữ liệu về thông tin đặt xe với từ khóa kiểm tra là ký hiệu của xe đó (car_ID). Server sẽ kiểm tra trên cơ sở dữ liệu – bảng task_control nếu như có car_do_ID giống với car_ID thì sẽ trả về một phản hồi chứa thông tin của chuyến xe đó để phần xe thực hiện nhiệm vụ; nếu không đúng xe sẽ thực hiện truy vấn khác

❖ Client là xe (vi điều khiển) gửi dữ liệu về server qua mô-đun 4G



Hình 6-16 Sơ đồ giải thuật khi Client là xe (vi điều khiển) gửi dữ liệu về server qua mô-đun 4G

Vi điều khiển thực hiện gửi các gói tin đến mô-đun 4G để thực hiện một POST request để đưa dữ liệu về các thông số của xe. Server sẽ kiểm tra các header và từ khóa, nếu đúng sẽ lưu thông tin đó vào cơ sở dữ liệu còn nếu sai thì sẽ bỏ qua gói tin.

6.6. Kết nối dữ liệu giữa xe và server qua mô-đun 4G

6.6.1. Khảo sát mô-đun 4G SIMA7670C

❖ Các lệnh config cho mô-đun SIMA7670C [15]

Màu xanh dương là dữ liệu gửi xuống mô-đun từ chân Tx của USB-UART xuống chân Rx của mô-đun, màu lục là phản hồi từ mô-đun gửi qua chân Rx của mô-đun lên chân Rx của USB-UART

```
AT
AT
OK
ATE0
ATE0
OK
```

❖ Các lệnh thực hiện HTTP Post request

```
AT+HTTPINIT
OK
AT+HTTPPARA="CONTENT","application/x-www-form-urlencoded"
OK
AT+HTTPPARA="URL","https://fleetmonitoring.000webhostapp.com/createGPSdata.php"
OK
AT+HTTPDATA=67,1000
DOWNLOAD
car_gps_ID=2&longitude=10639.67639&latitude=1045.59172&velocity=16
OK
AT+HTTPACTION=1
OK
+HTTPACTION: 1,200,20
AT+HTTPREAD=0,20
OK
```

```
+HTTPREAD: 20  
Posted successfully.  
+HTTPREAD: 0
```

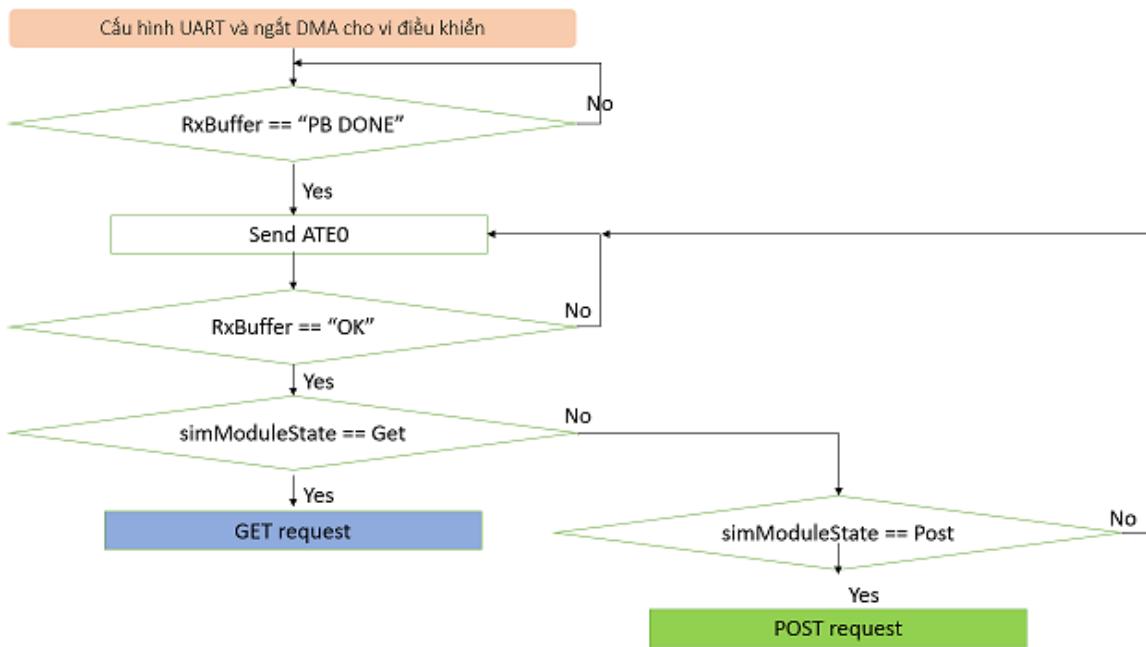
Trong điều kiện bình thường, mỗi POST request thường mất khoảng 3-5s để nhận được phản hồi là đã thực hiện request thành công +HTTPACTION: 1,200,20 (200 chính là status code OK trong giao thức HTTP). Còn trong các trường hợp xảy ra lỗi thì ta sẽ phải gửi lại lệnh hoặc config lại mô-đun.

❖ *Các lệnh thực hiện HTTP GET request*

```
AT+HTTPINIT  
  
OK  
  
AT+HTTPPARA="URL","https://fleetmonitoring.000webhostapp.com/searchGPSdata.php?frame_ID=1"  
  
OK  
  
AT+HTTPACTION=0  
  
OK  
  
+HTTPACTION: 0,200,99  
  
AT+HTTPREAD=0,99  
  
OK  
  
+HTTPREAD: 99  
{"frame_ID":"1","car_gps_ID":"2","latitude":"1047.96435","longitude":"10644.59685","velocity":"0"}  
+HTTPREAD: 0
```

Trong điều kiện bình thường, mỗi GET request thường mất khoảng 3-5s để có thể đọc được phản hồi từ SERVER (khi nhận được +HTTPACTION: 0,200,99 (200 chính là status code OK trong giao thức HTTP)) thì mới có thể thực hiện lệnh AT+HTTPREAD để đọc nội dung của gói tin trả về từ SERVER. Còn trong các trường hợp xảy ra lỗi thì ta sẽ phải gửi lại lệnh hoặc config lại mô-đun.

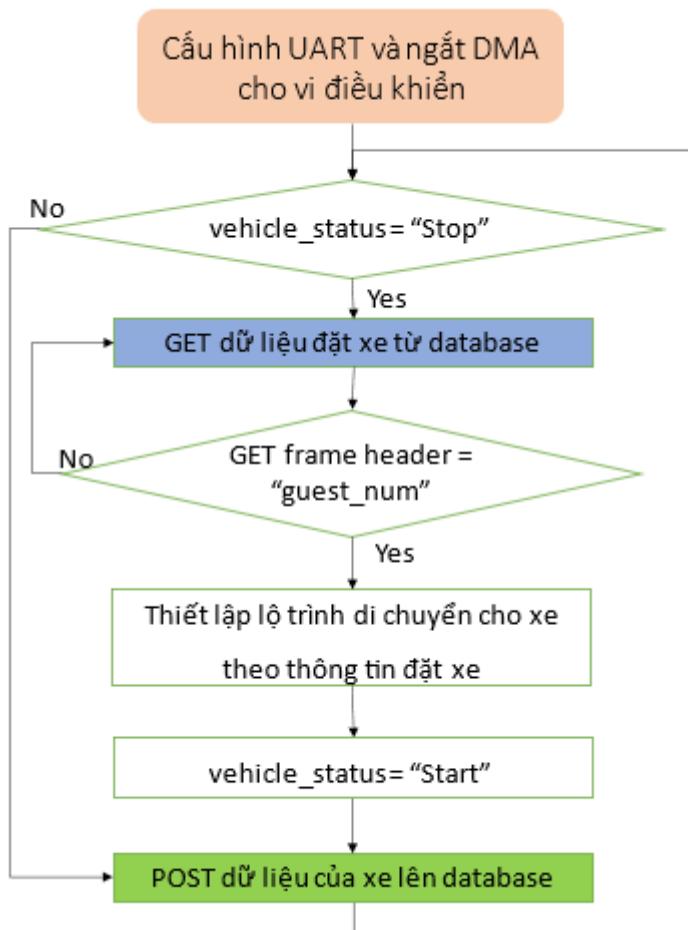
6.6.2. Lưu đồ giải thuật thực hiện POST, GET request từ vi điều khiển đến server qua mô-đun 4G.



Hình 6-17 Lưu đồ giải thuật thực hiện POST, GET request từ vi điều khiển qua mô-đun 4G

Khi khởi động, vi điều khiển sẽ chạy các hàm để khởi tạo hoạt động cho xe và mô-đun GPS-4G. Sau đó một UART kết nối với mô-đun 4G sẽ bắt đầu kiểm tra các gói tin. Nếu nhận được PB DONE thì vi điều khiển bắt đầu gửi các lệnh để xác định phương thức GET hay POST qua biến simMô-đunState. Nếu là GET request thì vi điều khiển sẽ lấy dữ liệu về thông tin đặt xe, nếu là POST request thì vi điều khiển sẽ thực hiện gửi dữ liệu về vị trí của xe lên server lưu vào cơ sở dữ liệu.

6.7. Ứng dụng thực tế của hệ thống vào xe tự hành



Hình 6-18 Lưu đồ giải thuật ứng dụng thực tế của hệ thống vào xe tự hành

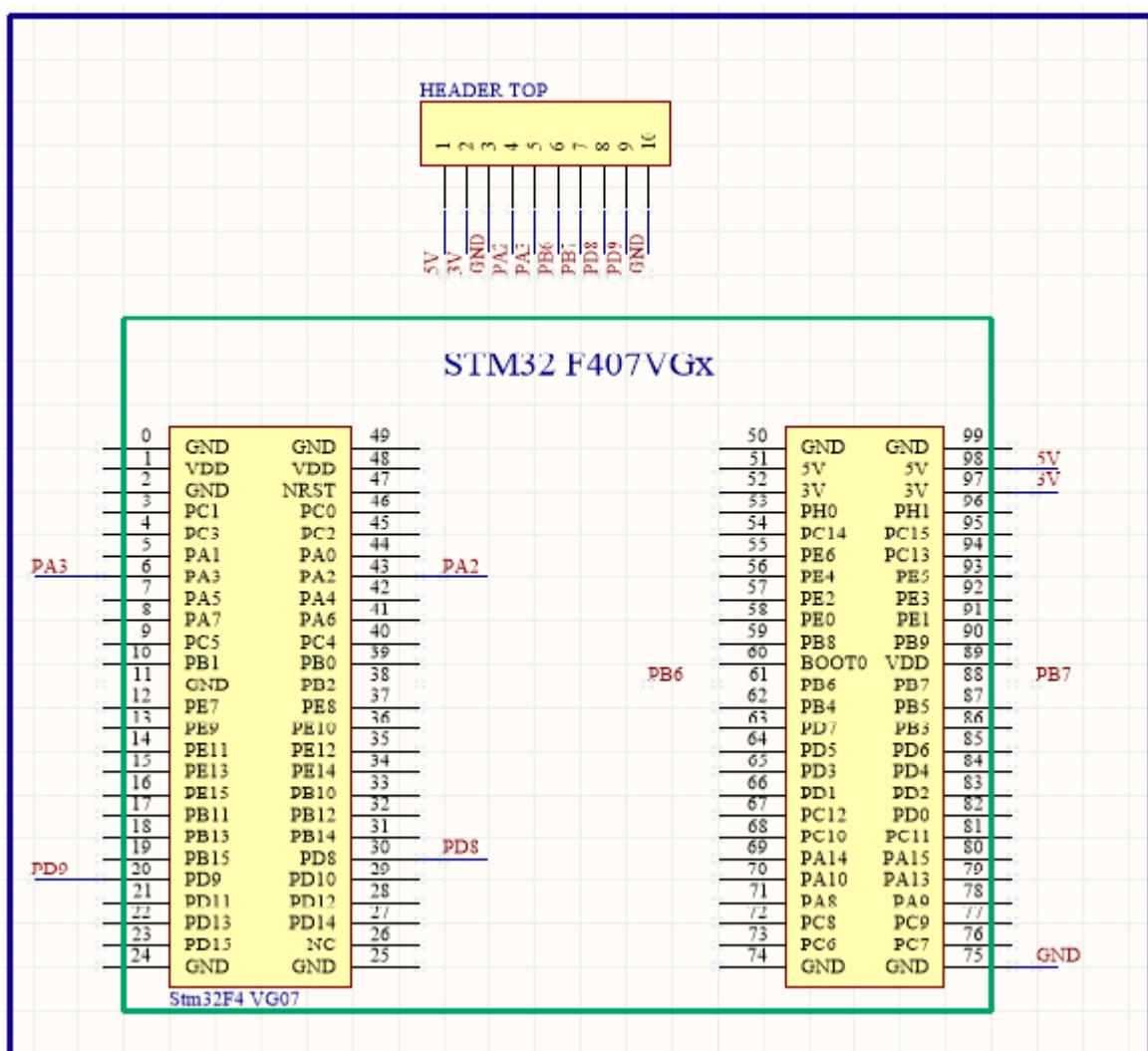
Sau khi khởi động hệ thống, hệ thống sẽ kiểm tra biến trạng thái của xe, nếu xe đang ở trạng thái “Stop” (đứng yên) thì sẽ gửi GET request để yêu cầu thông tin đặt xe từ server, nếu gói tin trả về có chứa thông tin của khách hàng thì hệ thống hoạch định quỹ đạo sẽ hoạch định lộ trình ngắn nhất cho quãng đường giữa 2 điểm đặt, sau đó xe sẽ chuyển sang trạng thái “Start” để bắt đầu thực hiện chuyến đi, khi đó xe sẽ chuyển sang POST dữ liệu hoạt động (vị trí, tốc độ) của xe lên database. Khi xe đứng yên, dữ liệu về vị trí hiện tại của xe đã được lưu trữ trước khi xe chuyển sang trạng thái “Stop”.

Chương 7. KẾT QUẢ THỰC HIỆN

7.1. Kết quả thực hiện phần cứng

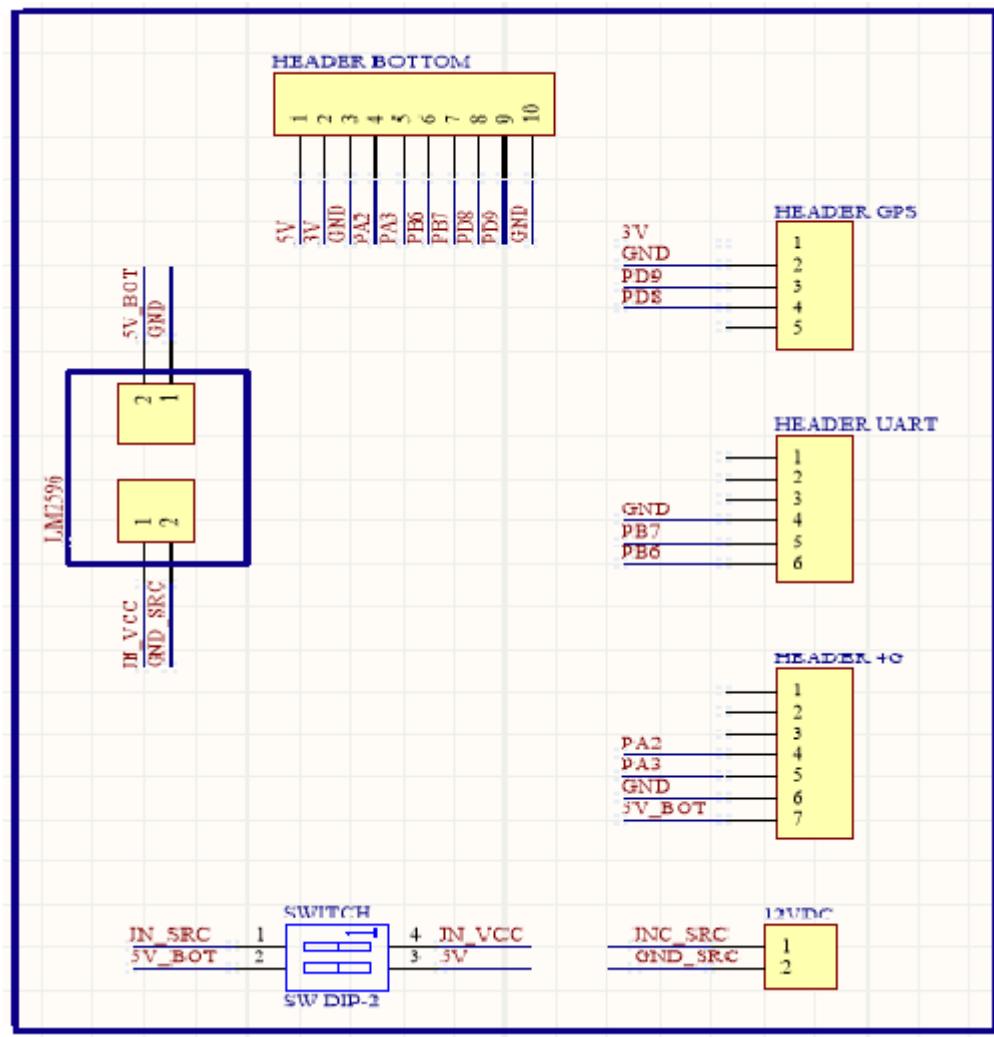
Phần cứng quản lý định vị và điều khiển từ xa gồm 2 lớp mạch (trên và dưới)

Lớp trên gắn STM32F407 và header ra chân của các ngoại vi cần thiết để kết nối xuống lớp dưới.



Hình 7-1 Mạch lớp trên của mô-đun GPS-4G

Lớp dưới gắn mạch giảm áp DC-LM2576, mô-đun GPS, mô-đun 4G cùng header ra chân PB6, PB7 để kết nối với USB-UART.



Hình 7-2 Mạch lớp dưới của mô-đun GPS-4G

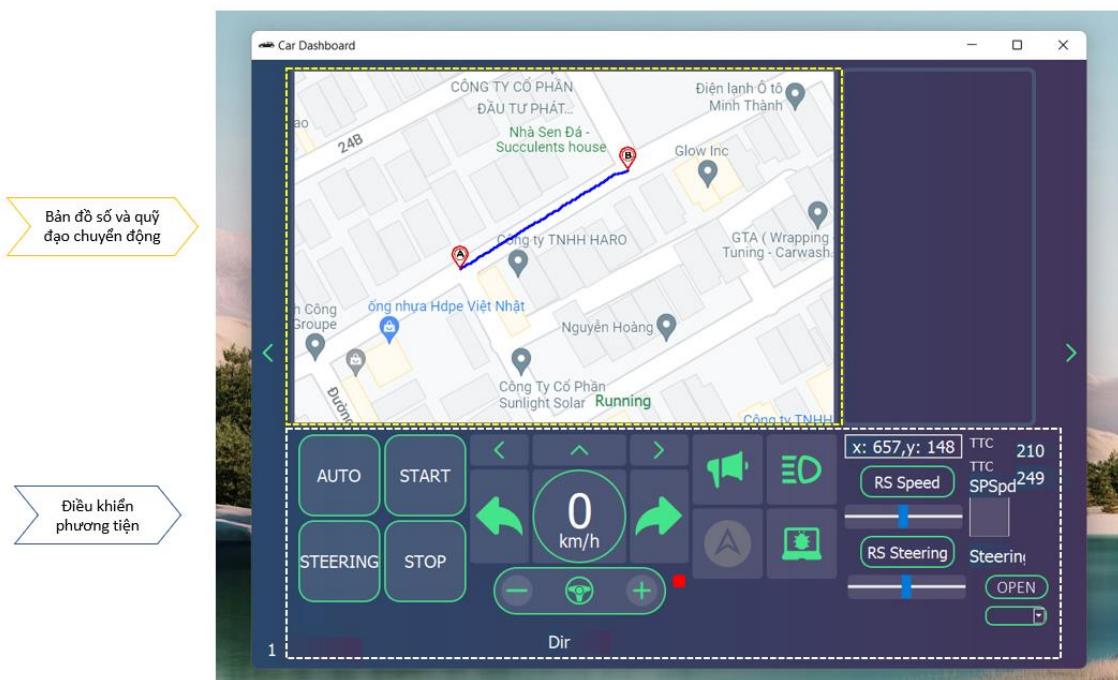


Hình 7-3 Hộp điện GPS-4G hoàn chỉnh

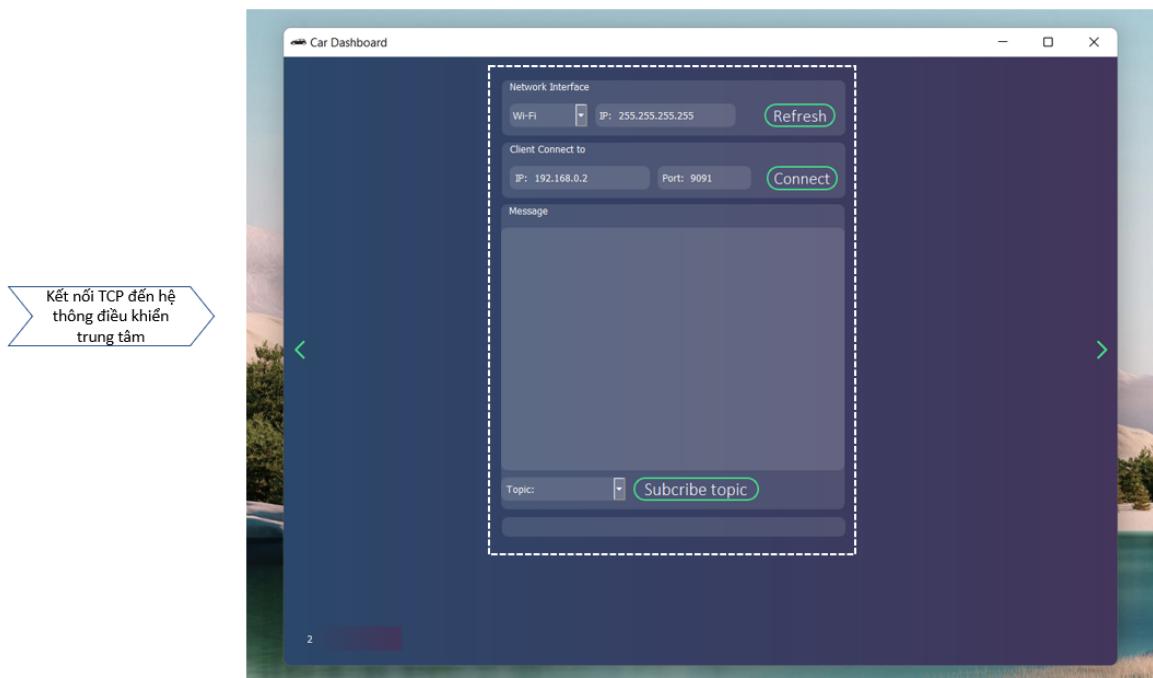
Trên vỏ hộp được bố trí các công tắc nguồn, các vị trí kết nối cho các ăng-ten GPS, ăng-ten 4G và USB-UART.

7.2. Kết quả thiết kế giao diện điều khiển cho xe điện tự hành

Giao diện hiển thị do nhóm tự phát triển trên Framework Qt C++. Giao diện có chức năng hỗ trợ người dùng chọn quỹ đạo di chuyển, người dùng có thể sử dụng giao diện để thực hiện điều khiển xe thông qua kết nối TCP giữa GUI với hệ thống điều khiển trung tâm.



Hình 7-4 Giao diện chính để điều khiển xe



Hình 7-5 Thực hiện kết nối TCP với trung tâm điều khiển

Dánh giá, nhận xét

Giao diện hoạt động ổn định, các tính năng đáp ứng được nhu cầu người dùng. Hiện tại giao diện vẫn chạy trên một máy tính riêng đặt trên hệ thống xe điện, hướng phát triển sắp tới sẽ thêm các chức năng cần thiết, loại bỏ các chứng năng chỉ sử dụng trong quá trình khảo nghiệm. Dàn tích hợp giao diện lên màn hình hiển thị của xe điện tự hành.

7.3. Kết quả thực hệ thống phần mềm hoạch định quỹ đạo

7.3.1. Kết quả xử lý dữ liệu cảm biến GPS

Sau khi lập trình vi điều khiển xử lý gói tin và trích xuất được vị trí, nhóm tiến hành khảo sát lại độ chính xác của GPS trong môi trường thực tế.

Đối với thiết bị GPS sẽ rất nhạy với nhiễu, vì vậy khi thực khảo sát cần chọn khu vực thoáng đãng, tránh các tòa nhà cao tầng, chọn ngày có thời tiết đẹp, ít mây để tránh nhiễu từ trường, đồng thời chọn khu vực khảo sát dễ đo đạc để có thể so sánh đối chiếu kết quả. Sau đó tiến hành kết nối GPS với vi điều khiển, chờ GPS có tín hiệu và tín hiệu đó được hội tụ và ổn định, gửi tọa độ này

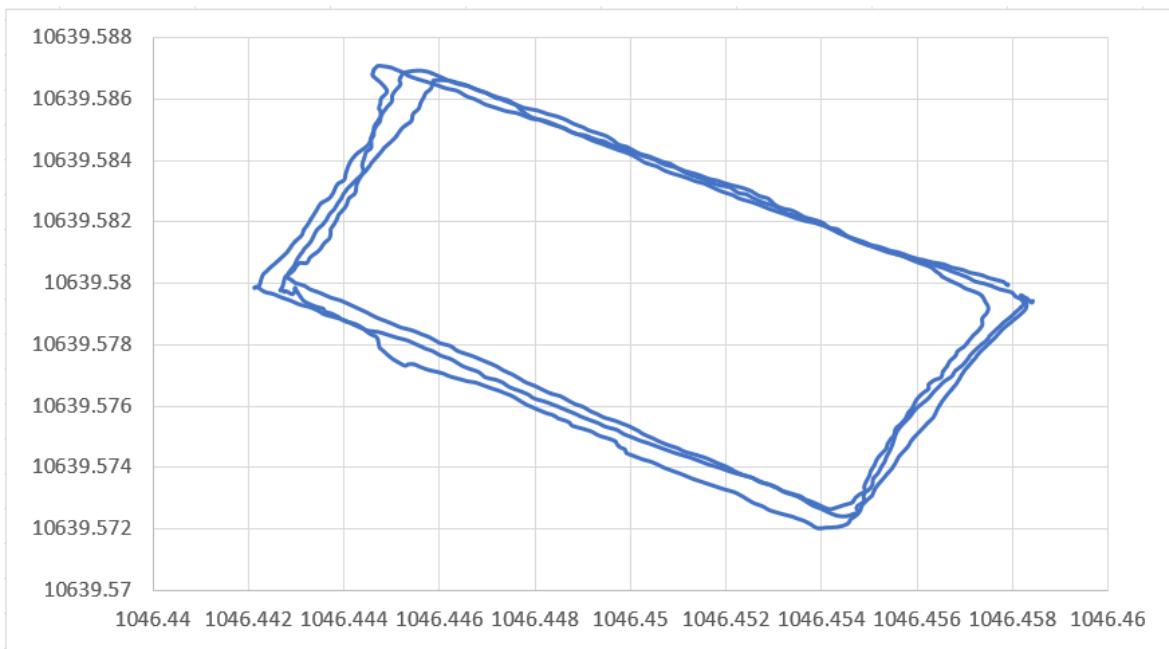
lên máy tính và xuất sang một tập tin excel để phục vụ thực hiện quá trình vẽ đồ thị khảo sát.



Hình 7-6 Địa điểm khảo sát GPS

Đây là sân bóng rổ của trường Đại học Bách Khoa TP. HCM, một khu vực thoáng đãng đủ để tín hiệu GPS ổn định. Khu vực có kích thước:

Thực hiện đi 3 vòng quanh sân bóng để thu thập dữ liệu và đánh giá kết quả



Hình 7-7 Dữ liệu GPS

Số liệu trên đây là tọa độ thực khi GPS trả về do đó ta sẽ sử dụng công thức **Haversine** [16] tính quỹ đạo cung tròn tạo bởi 2 tọa độ, đây cũng chính là khoảng cách địa lý thực tế theo đường chim bay:

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1) * \cos(\varphi_2) * \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 * \sin^{-1}(\sqrt{a})$$

$$d = R * c$$

Trong đó,

φ (rad) là vĩ độ

λ (rad) là kinh độ

R (km) là bán kính Trái Đất ($R_{mean} = 6371km$)

d (km) là khoảng cách địa lý

Bảng 7-1 Tính toán kích thước sân dựa trên dữ liệu trả về từ GPS

	Chiều dài (m)	Chiều rộng (m)

Lần 1	28.97147	14.57101
Lần 2	26.19221	14.02635
Lần 3	26.41356	15.12887
Trung bình	27.19241	14.57541

Thực hiện tính sai số tuyệt đối của mỗi lần đo:

$$\Delta P_k = |\bar{P} - P_k|$$

Bảng 7-2 Sai số tuyệt đối của các lần đo

	Chiều dài (m)	Chiều rộng (m)
Lần 1	1.77906	0.0044
Lần 2	1.0002	0.54906
Lần 3	0.77885	0.55346

Sai số ngẫu nhiên được tính theo công thức:

$$\overline{\Delta P} = \frac{\Delta P_1 + \Delta P_2 + \dots + \Delta P_n}{n}$$

Tuy phép đo chỉ được thực hiện 3 lần ($n < 5$), để khách quan giá trị sai số ngẫu nhiên sẽ là giá trị sai số tuyệt đối cực đại trong 3 lần đo.

	Chiều dài (m)	Chiều rộng (m)
Sai số ngẫu nhiên	1.77906	0.55346

	Chiều dài (m)	Chiều rộng (m)
Kích thước sân đo bằng thước	27.825	14.892
Kích thước sân tính toán từ dữ liệu GPS	27.19241	14.57541
Sai lệch so với kích thước thực tế	0.63259	0.31659

Đánh giá, nhận xét:

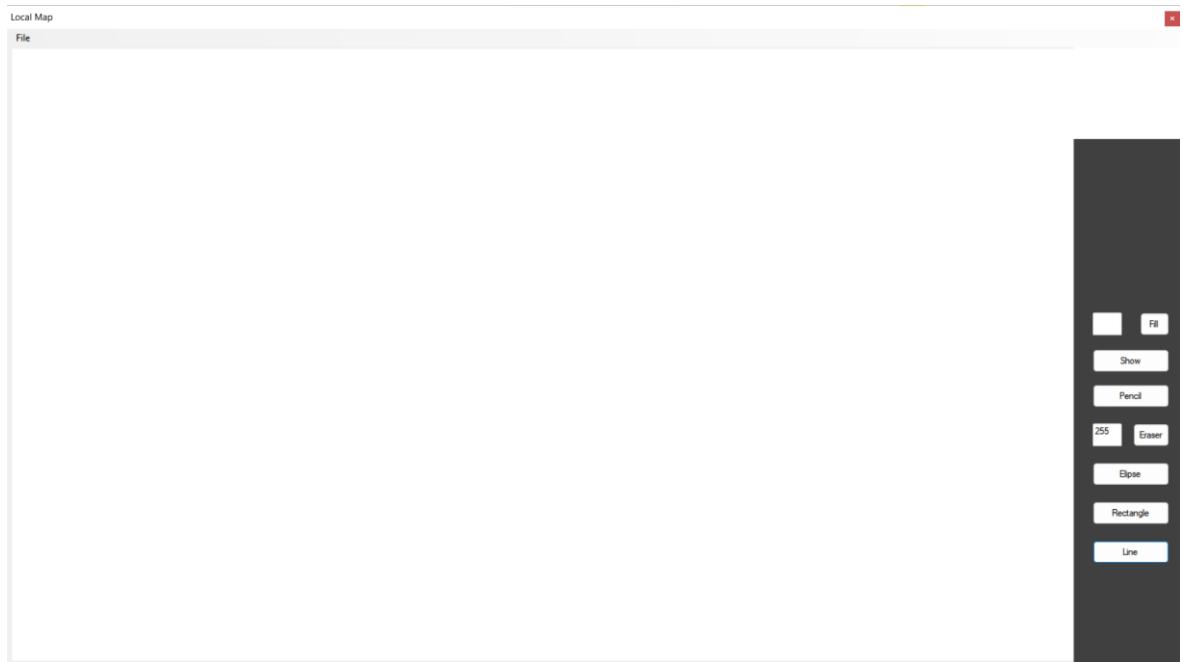
- Với sai lệch giữa phép đo sử dụng GPS và kích thước thực tế dao động từ 0.3 đến 0.6 mét này có thể chấp nhận trong quá trình quan sát di chuyển theo quỹ đạo đã hoạch định trong những điều kiện thực hiện khảo sát như trên.
- Tuy nhiên sai số ngẫu nhiên có thể lên đến 1.7m vì vậy cần tìm nơi thoáng đãng khi hoạch định quỹ đạo chuyến động. Cần tìm những con đường có ít xe qua lại để đảm bảo an toàn trong quá trình thực nghiệm.
- Ngoài ra độ chính xác của GPS còn phụ thuộc vào các yếu tố không gian, môi trường. Nếu chọn nơi thực nghiệm xe tương tự như khu vực khảo sát GPS, ít nhà cao tầng, thoáng đãng, điều kiện thời tiết tốt thì sự sai lệch giữa quỹ đạo thực tế và quỹ đạo di chuyển dao động từ 0.3m đến 0.6m là hoàn toàn có thể chấp được.

7.3.2. Kết quả thực hiện số hóa bản đồ

7.3.2.1. Công cụ thực hiện số hóa

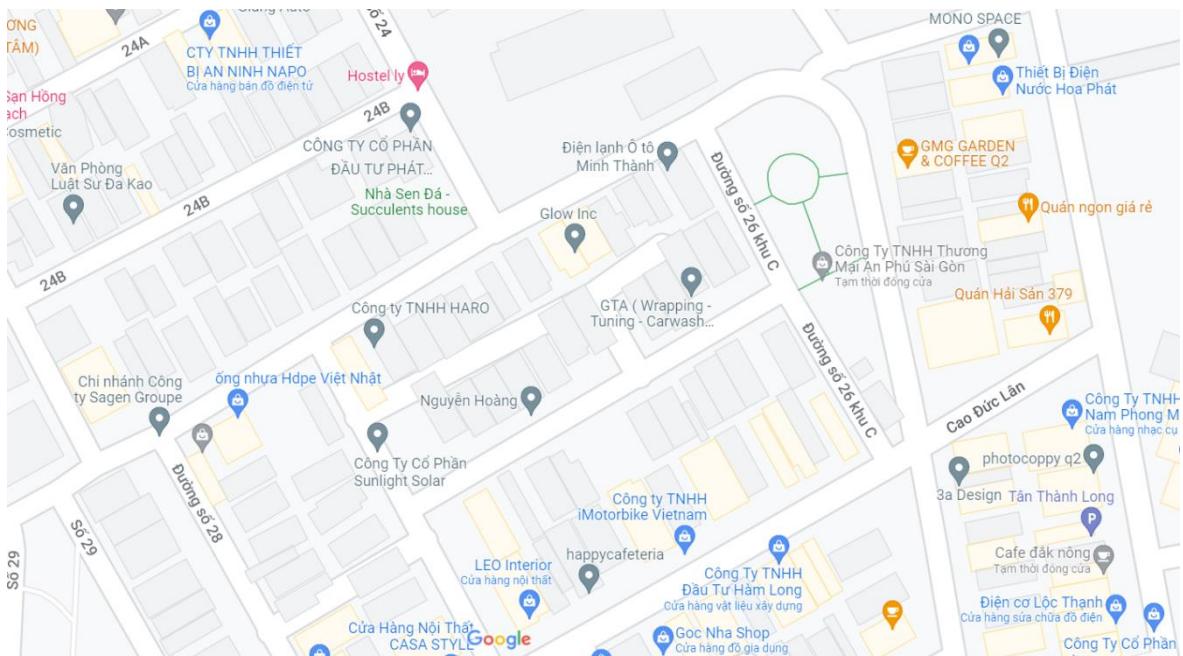
Như phương án được đề xuất phía trên nhóm thực hiện thiết kế một công cụ giúp số hóa lại bản đồ của khu vực cục bộ. Công cụ này cho phép thay đổi giá

trị các pixel theo mong muốn của người thiết kế. Ở đây chỉ tạo lớp bản đồ đơn giản để thể hiện con đường mà xe sẽ di chuyển.

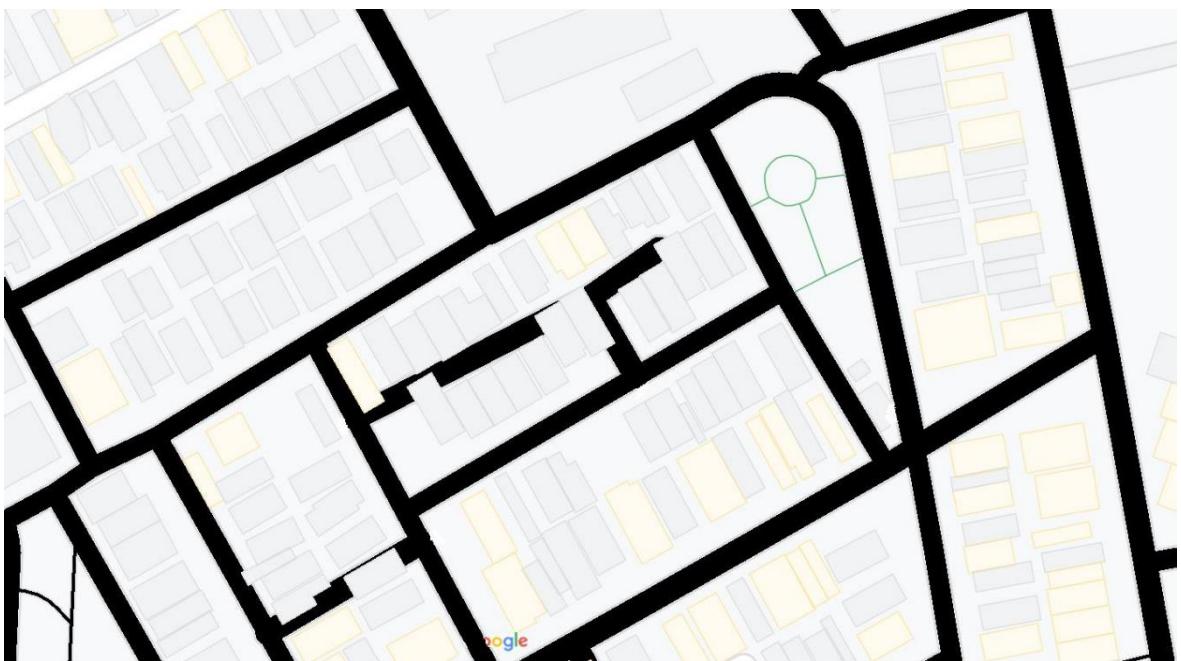


Hình 7-8 Công cụ hỗ trợ xây dựng bản đồ số

Công cụ trên sẽ đọc dữ liệu bản đồ thu thập từ Google Maps và có các tính năng hỗ trợ vẽ các đường thẳng hay đường cong để xây dựng bản đồ.

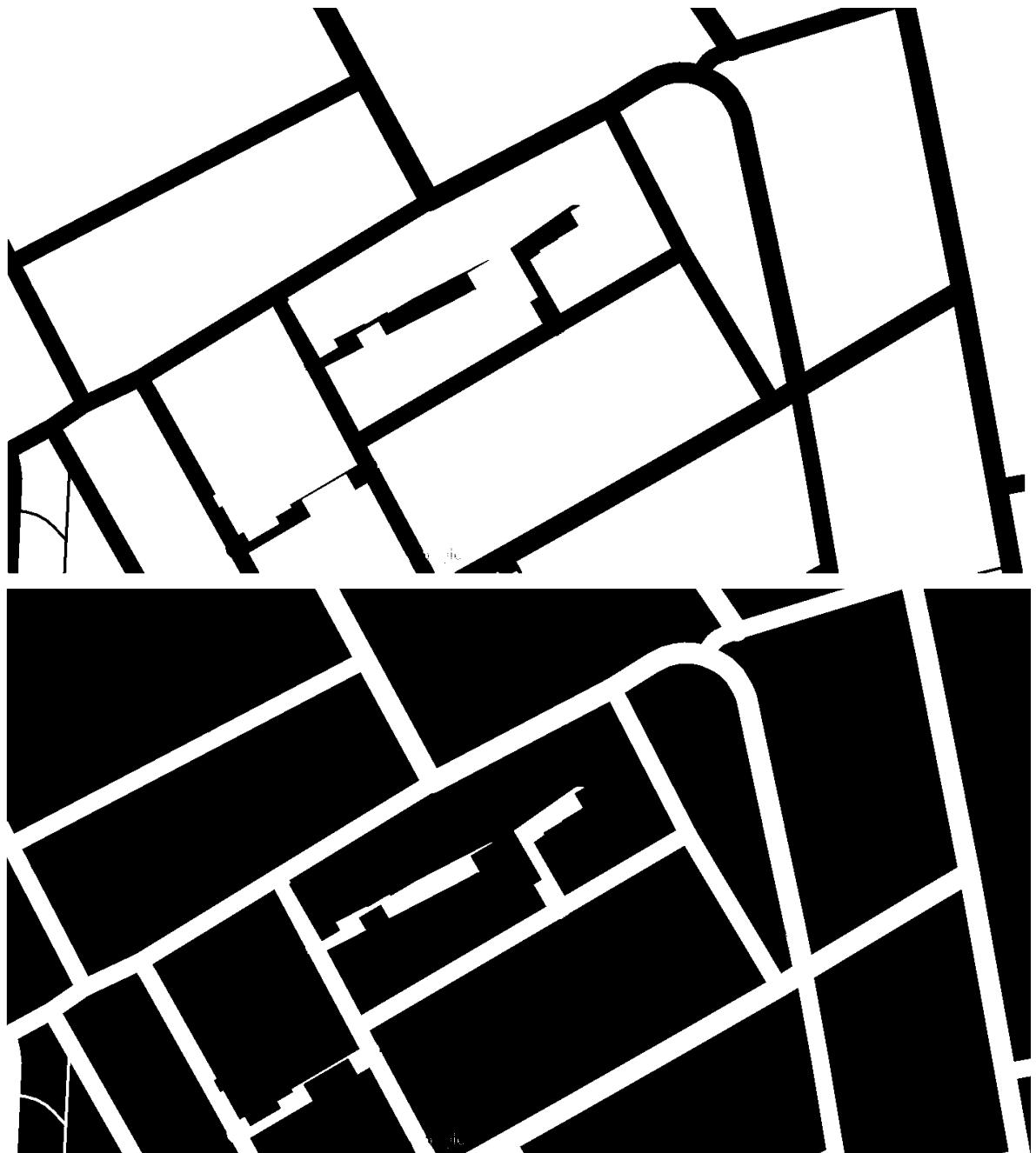


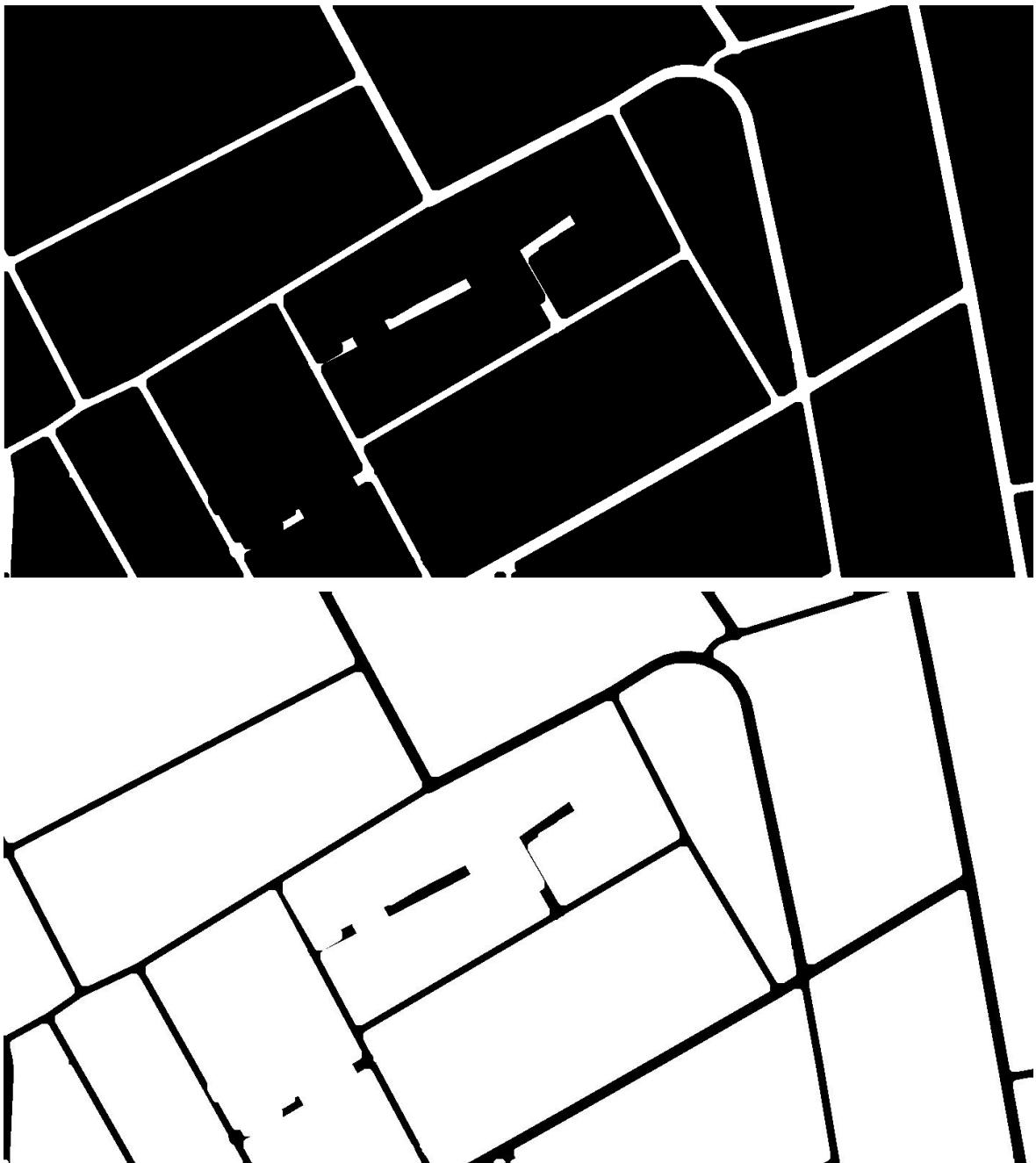
Hình 7-9 Bản đồ thu thập được từ Google Maps



Hình 7-10 Bản đồ sau khi qua công cụ số hóa







Hình 7-11 Lớp bản đồ thu được sau khi thực hiện số hóa qua các bước trên

Đây sẽ là lớp bản đồ áp dụng thuật toán A * để tìm quỹ đạo chuyển động cho xe. Việc làm cho các đường đi nhỏ lại giúp quỹ đạo hoạch định đúng với mong muốn hơn, giảm đi khối lượng tính toán. Mục tiêu phát triển cao hơn của việc xây dựng bản đồ số là có thể xây dựng thêm nhiều lớp bản đồ khác nhau để thể hiện đặc trưng chi tiết nhất cho khu vực đó để phục vụ cho mục tiêu phát triển lâu dài sau này.

7.3.2.2. Đồng bộ tọa độ thực tế và bản đồ số hóa

Khi thực hiện thu thập dữ liệu ta sẽ lấy tọa độ hai điểm đầu cuối của khu vực sau đó tính toán lại để đồng bộ với tọa độ của bản đồ số.

Với bản đồ trên ta có 2 tọa độ đầu cuối sau:

$$10.79691, 106.74219$$

$$10.79525, 106.74522$$

Do tọa độ trả về từ GPS chỉ chính xác đến 5 chữ số thập phân. Vì vậy để đảm bảo tính khách quan của độ chính xác, tọa độ đầu cuối này cũng chỉ lấy đến 5 chữ số thập phân. Tọa độ trên tương ứng với bản đồ số có kích thước 1420x790. Như vậy ứng với mỗi pixel theo chiều ngang thì kinh độ sẽ thay đổi

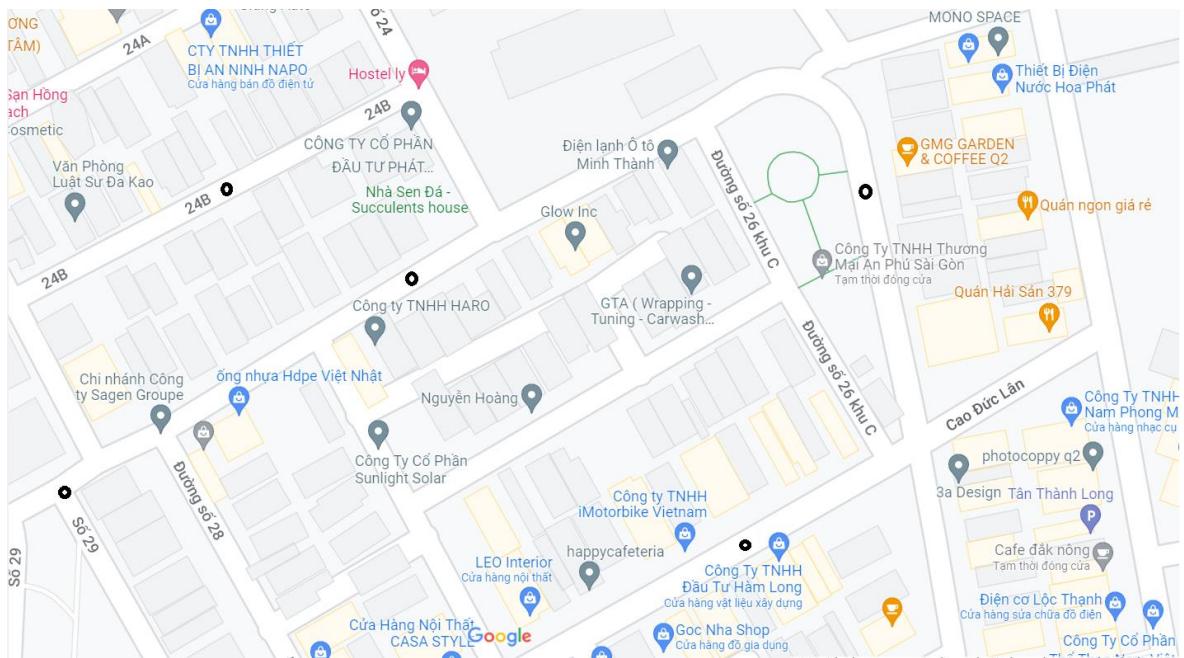
$$\frac{106.74522 - 106.74219}{1420} = 0.0000021338^{\circ}$$

, ứng với mỗi pixel thay đổi theo chiều dọc thì vĩ độ sẽ thay đổi

$$\frac{10.796901 - 10.79521}{790} = 0.0000021405^{\circ}$$

7.3.2.3. Khảo sát độ chính xác của bản đồ xây dựng

Chọn tọa độ 5 điểm từ bản đồ số hóa (5 vòng tròn đen), thực hiện tính toán dựa vào công thức tuyến tính trên và vẽ lại vị trí trên bản đồ của Google Maps.

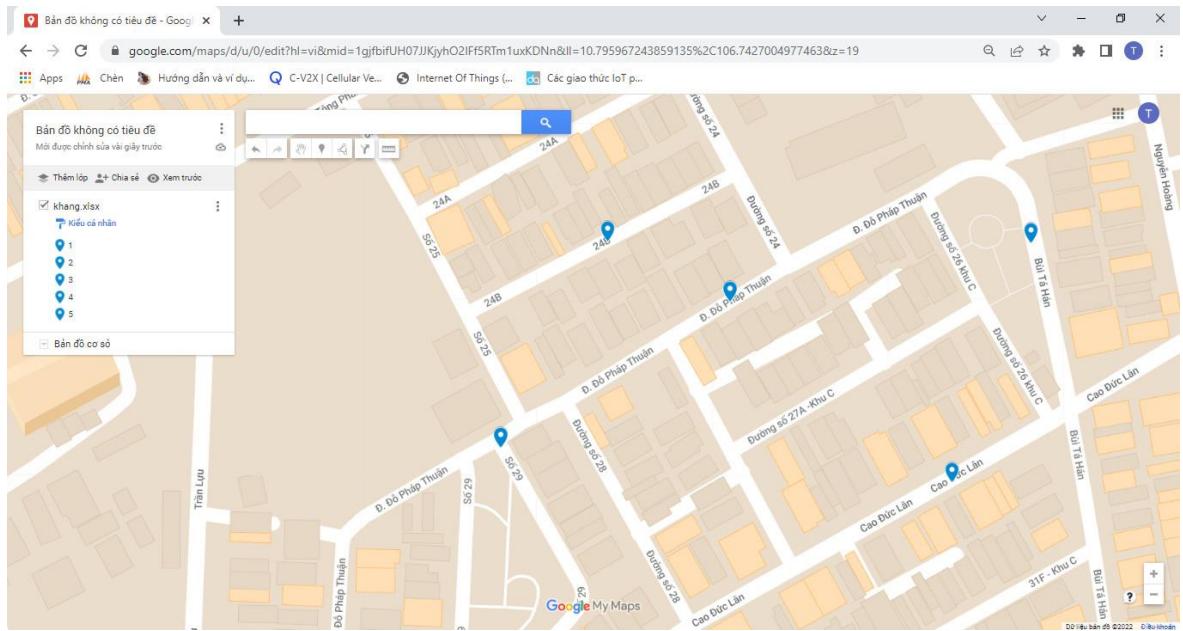


Hình 7-12 Chọn 5 điểm khảo sát bản đồ số

Bảng 7-3 Bảng khảo sát độ chính xác bản đồ số

	x	y	Tọa độ thực
Điểm 1	264	219	(10.79643, 106.74275)
Điểm 2	488	327	(10.79620, 106.74323)
Điểm 3	1039	223	(10.79642, 106.74441)
Điểm 4	68	588	(10.79564, 106.74234)
Điểm 5	894	651	(10.79551, 106.74409)

Sau khi thực hiện tính toán, ta vẽ tọa độ lên lên Google Maps để so sánh kết quả



Hình 7-13 Tọa độ 5 điểm của bản đồ số được vẽ lên Google Maps

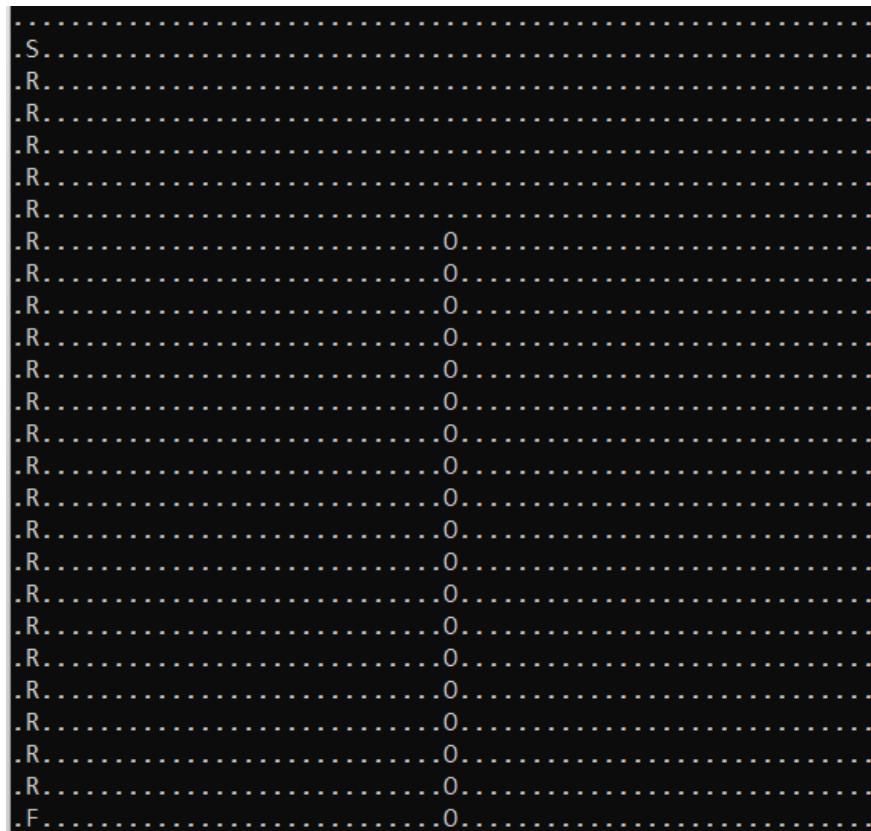
Bằng cách quan sát trực quan so sánh vị trí ta thấy vị trí các điểm trên bản đồ số gần như trùng khớp với vị trí tọa độ thực trên bản đồ. Điều này làm tăng độ tin cậy cho bản đồ số trong quá trình sử dụng.

7.3.3. Kết quả thực hiện thuật toán A *

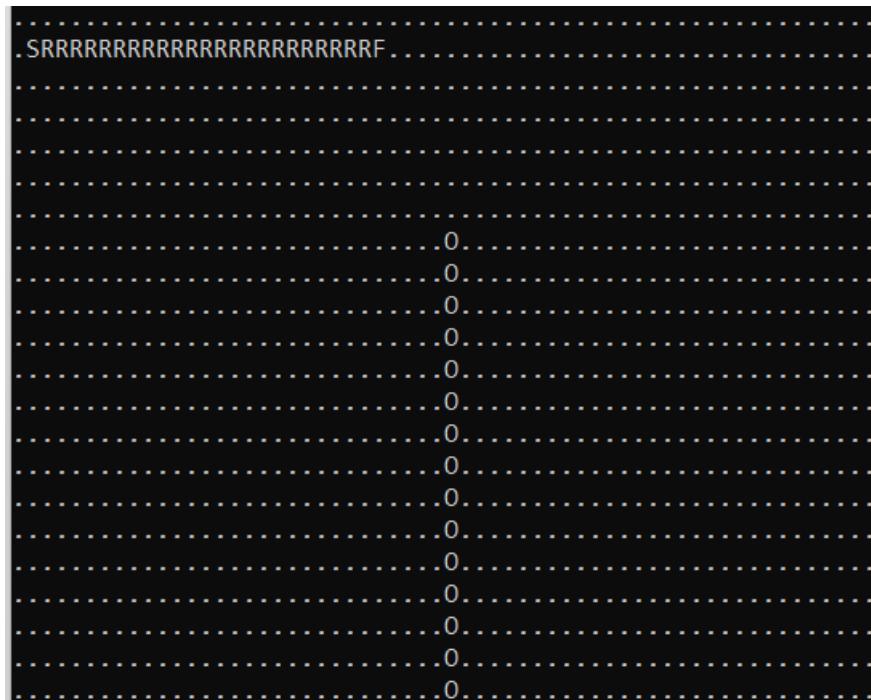
Nhóm chọn ngôn ngữ C++ để thực hiện thuật toán A *. Dựa trên giải thuật từ nhiều nguồn trên Internet, và giải thuật cốt lõi được trình bày phía trên nhóm đã thực hiện và được các kết quả sau:

7.3.3.1. Kiểm tra sơ bộ kết quả

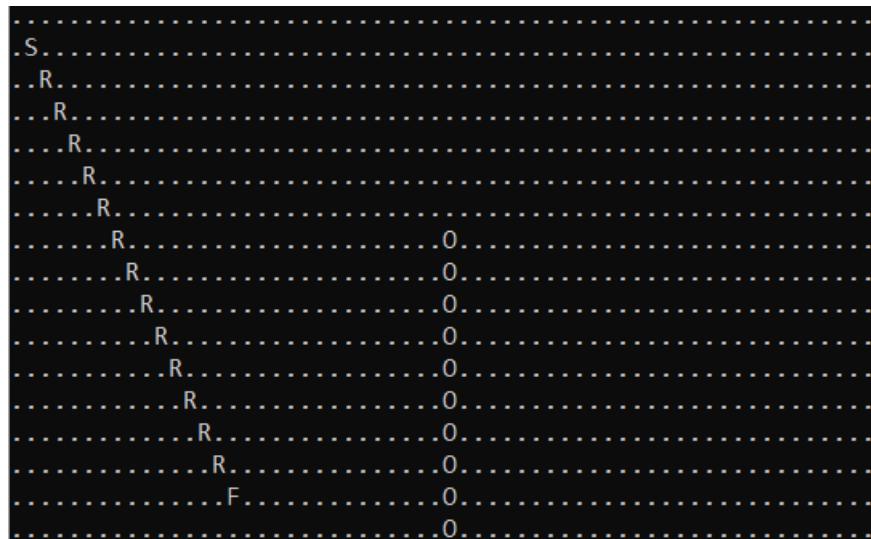
Nhóm thực hiện lập trình C++ và giải thuật tìm đường đi A* trên một lối đồ thị vào hiển thị lên “terminal” của Qt. Trong đó ‘S’ là điểm bắt đầu, ‘R’ là điểm của đường đi A *, ‘F’ là điểm đích, ‘0’ là các vật cản trên đường đi.



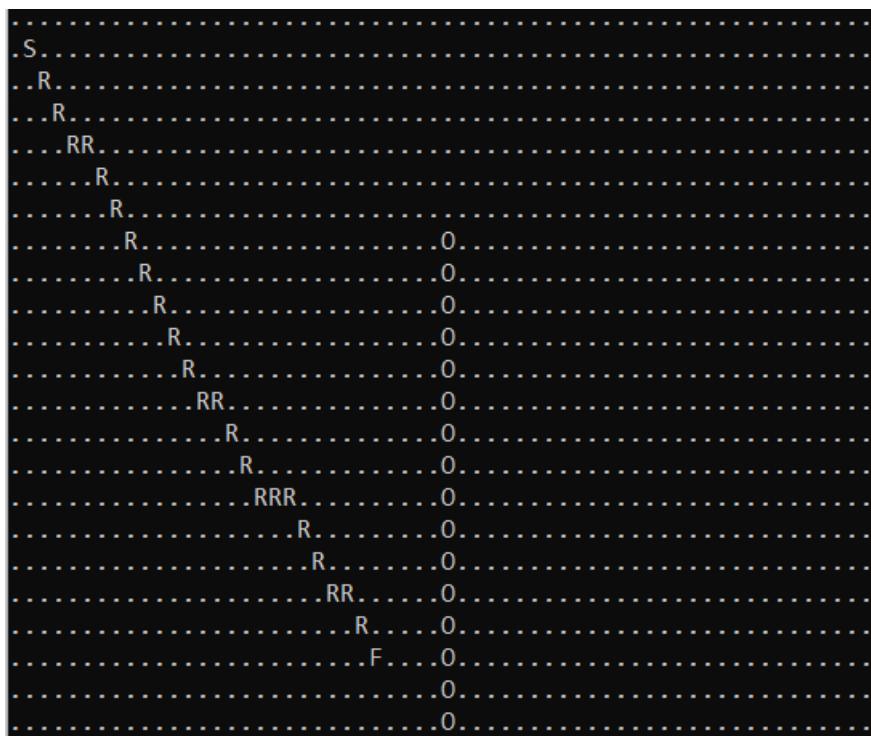
Hình 7-14 Đường đi A* thẳng xuống



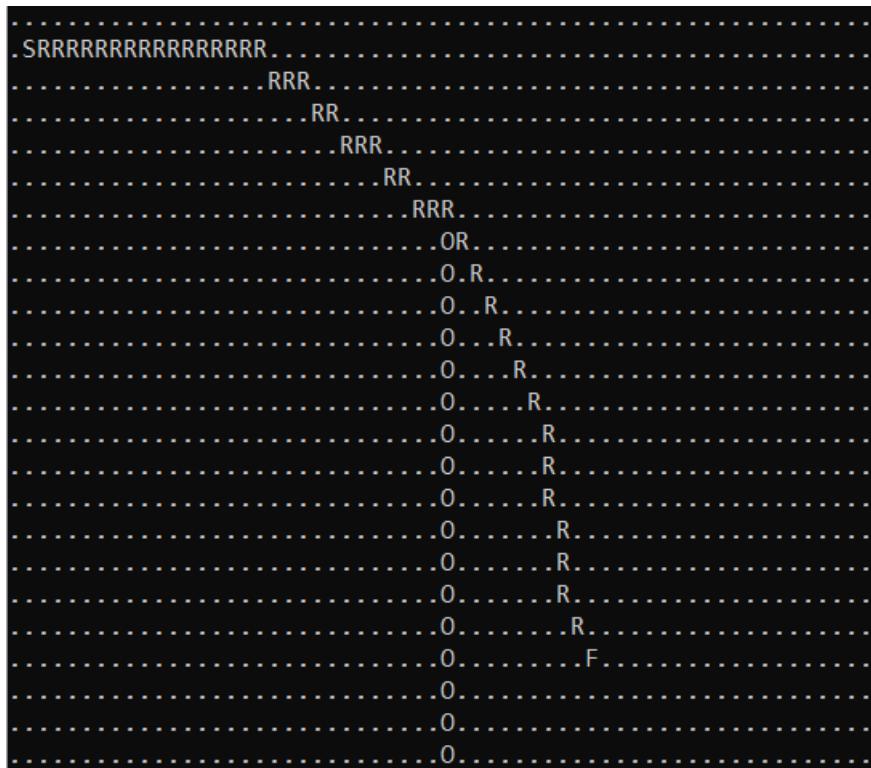
Hình 7-15 Đường đi A * thẳng ngang



Hình 7-16 Đường đi A * đường chéo



Hình 7-17 Đường đi A * đường chéo



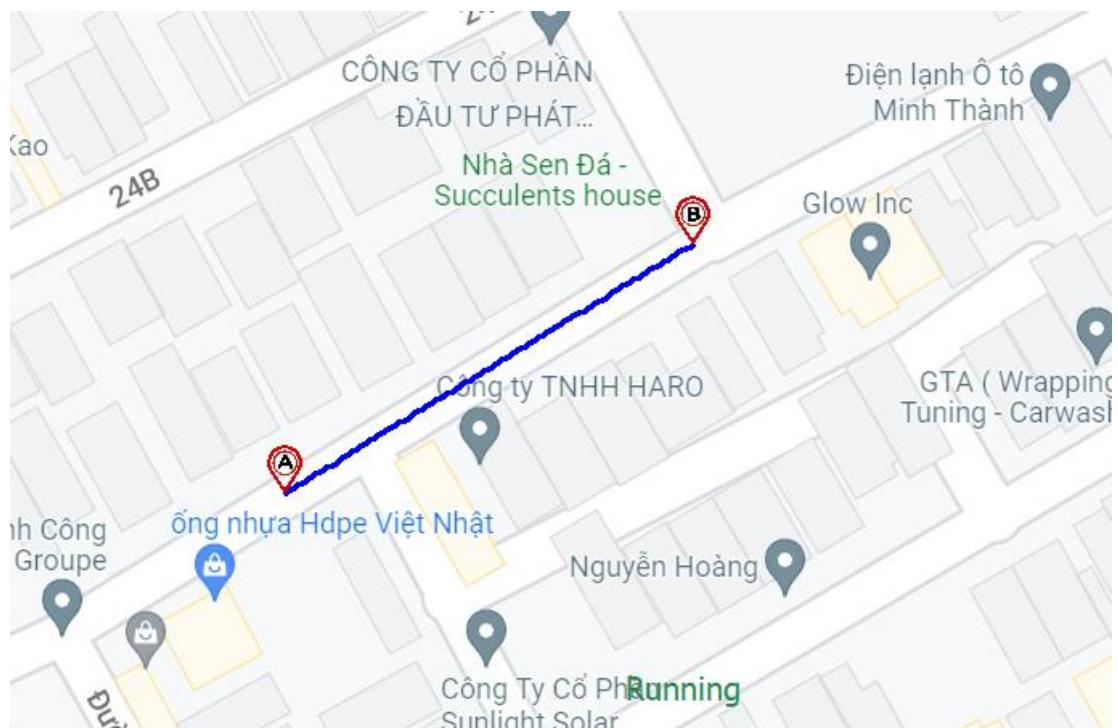
Hình 7-18 Đường đi A * có tránh vật cản

Đánh giá nhận xét:

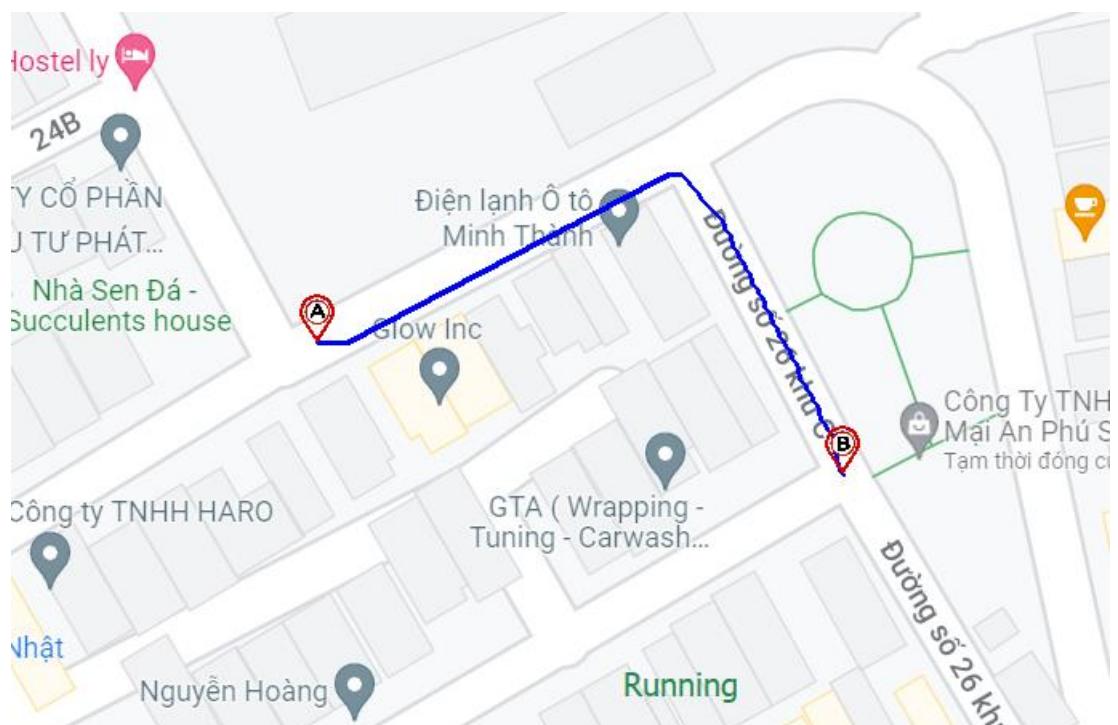
- Với các đường đi thẳng, đường chéo có hệ số góc $|k| = 1$, bằng quan sát thực tế, thuật toán A * cho kết quả rất tốt, tối ưu hóa được đường đi.
- Các đường chéo với hệ số góc khác hay cần di chuyển tránh vật cản thì quỹ đạo có xu hướng bị rãnh cưa, do tại một nút khi quyết định mở rộng tìm kiếm để đến được vị trí đích, A * có xu hướng mở rộng theo các điểm ngang và gần ở phía điểm đích, vì chi phí $h(n)$ tại điểm này nhỏ, $g(n)$ tại vị trí này cũng nhỏ hơn so với với điểm tại điểm đường chéo ($g(n) = 1$ tại vị trí này và $g(n) = \sqrt{2}$ tại điểm ở vị trí đường chéo). Tuy nhiên, đây vẫn là một đường đi tối ưu.

7.3.3.2. Kết quả tìm đường trên bản đồ số

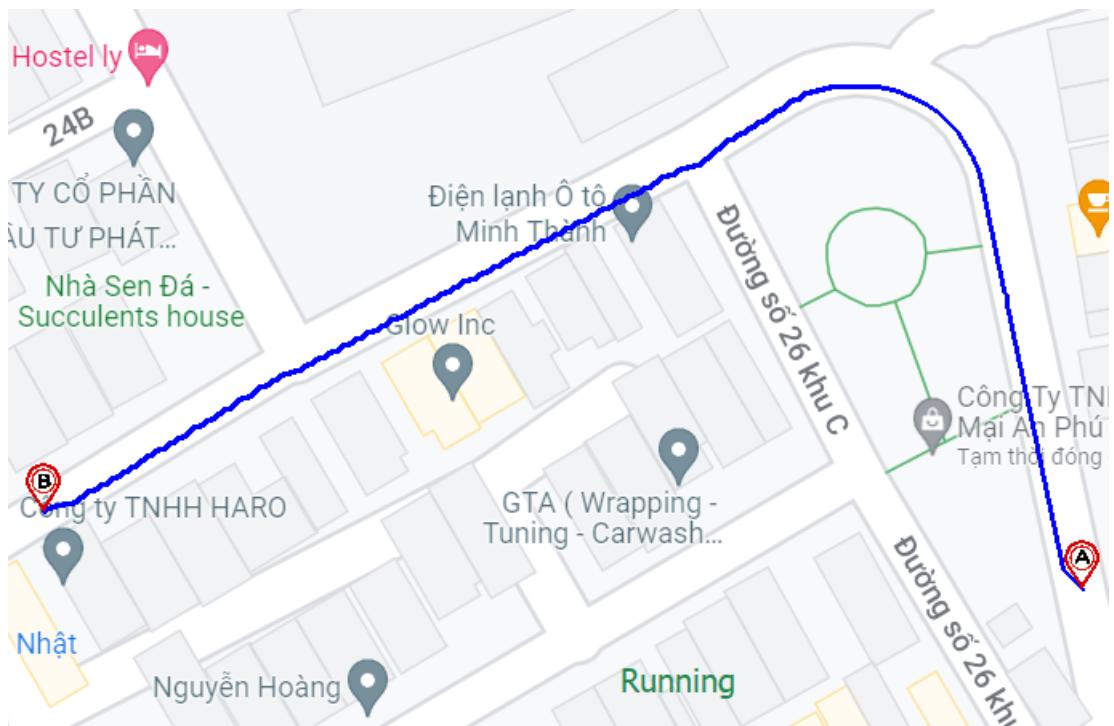
Việc chọn các điểm là đường đi giữa hai điểm, A * sẽ dùng bản đồ số thực hiện phía trên để tính toán nhằm tạo



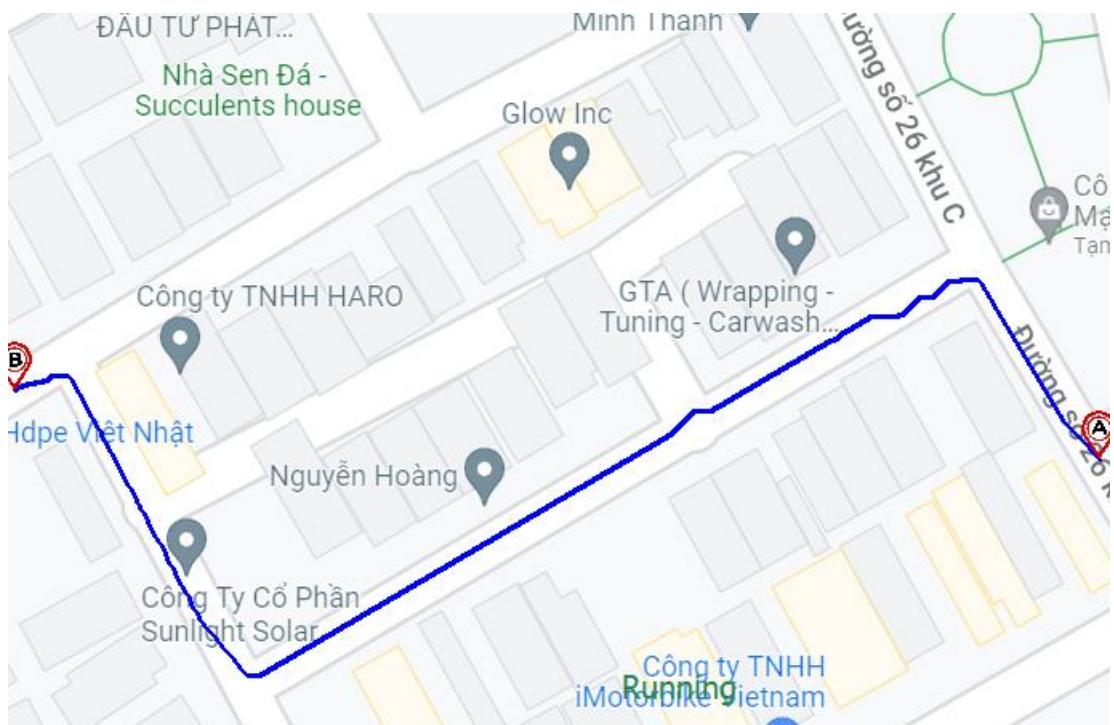
Hình 7-19 Quỹ đạo A * với đường đi thẳng



Hình 7-20 Quỹ đạo A * với đường cong ngắn



Hình 7-21 Quỹ đạo A * với đoạn đường dài



Hình 7-22 Quỹ đạo A * với đoạn đường phức tạp hơn

Đánh giá, nhận xét

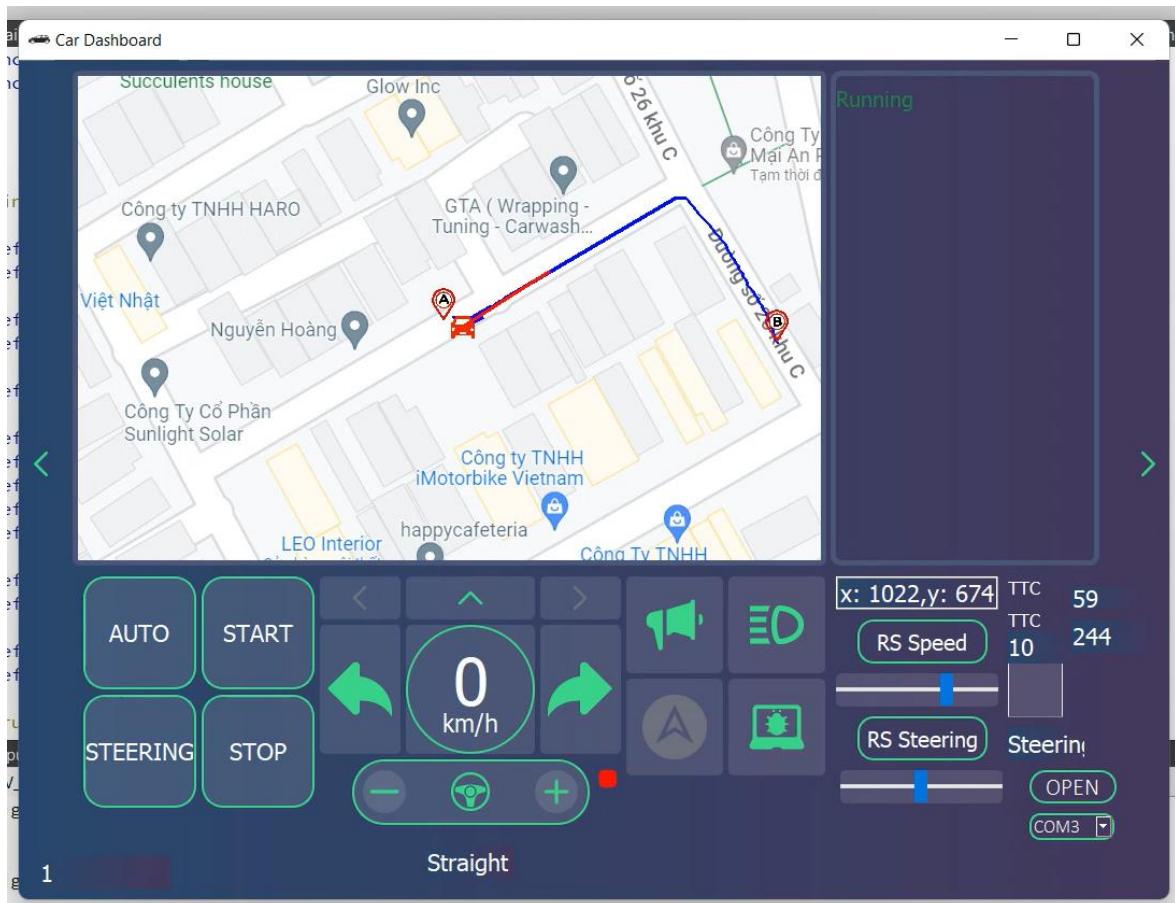
- Với hầu hết các trường hợp thì thuật toán đều tìm được quỹ đạo di chuyển tối ưu và với thời gian tính toán nhanh (chưa đến 0,5 giây), với

những đoạn đường ngắn, gần như đường đi được hiển thị ngay lập tức sau khi chọn được hai điểm. Đây cũng chính là ưu điểm lớn nhất của thuật toán A *, nó mở rộng đường đi một cách có định hướng dựa trên tối ưu hàm mục tiêu $f(n) = g(n) + h(n)$ vì vậy sẽ chọn được quỹ đạo tốt nhất với khối lượng tính toán là thấp nhất.

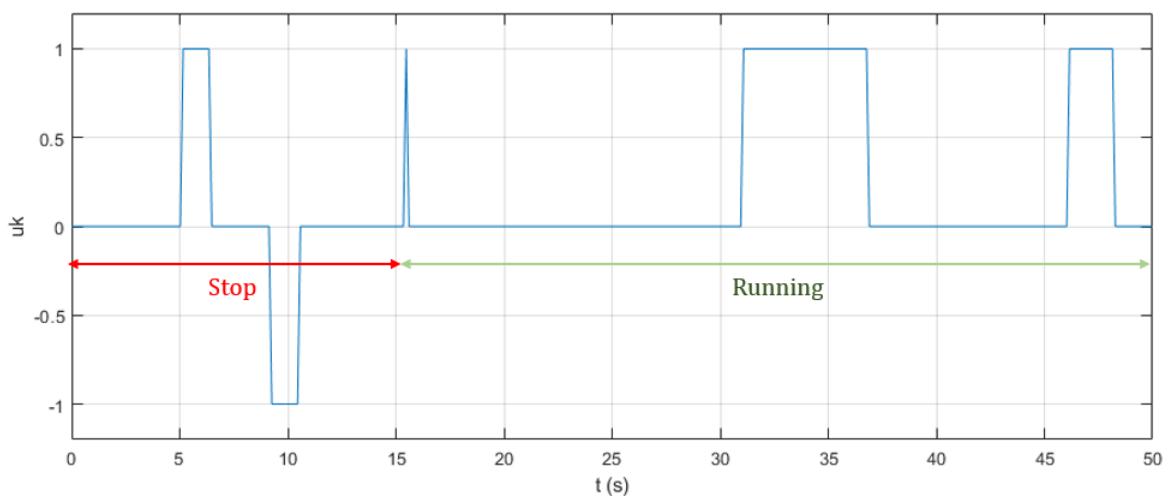
- Có một số khuyết điểm, nhìn chung các quỹ đạo đều có các răng cưa, điều này như lý giải lúc thực hiện kiểm tra sơ bộ thuật toán, A * có xu hướng mở rộng theo các điểm ngang và gần ở phía điểm đích, vì chi phí $h(n)$ tại điểm này nhỏ, $g(n)$ tại vị trí này cũng nhỏ hơn so với so với điểm tại điểm đường chéo ($g(n) = 1$ tại vị trí này và $g(n) = \sqrt{2}$ tại điểm ở vị trí đường chéo). Tuy nhiên, nhờ vào hàm tìm giá trị nhỏ nhất của hàm $f(n)$ nên quỹ đạo tìm được vẫn được xem là tối ưu.

7.3.4. Kết quả điều hướng kết hợp phương pháp hồi quy tuyến tính và quỹ đạo A *

Thực hiện chọn hai điểm trên giao diện điều khiển, nhấn nút chỉ đường, quỹ đạo A * được thiết lập, di chuyển theo quỹ đạo đã thiết lập để thu thập đánh giá kết quả tín hiệu điều hướng.



Hình 7-23 Chọn địa điểm di chuyển trên giao diện điều khiển



Hình 7-24 Tín hiệu điều hướng

Đánh giá, nhận xét:

- Từ 0s đến 15s xe đứng yên, 15s đến 50s xe di chuyển theo quỹ đạo A *.
- Mức tín hiệu 0 là *đi thẳng*, 1 là *rẽ phải*, -1 là *rẽ trái*.

- Đối với các thiết bị GPS có độ chính xác thấp, dữ liệu vị trí của GPS thông thường sẽ được đánh giá trong một khu vực, bởi vì dữ liệu đó sẽ có độ trôi lớn tại cùng một vị trí. Điều này đồng nghĩa với khi đứng yên việc xác định phương hướng sẽ không thể chính xác, dẫn đến tín hiệu điều hướng sai. Vì vậy ta sẽ không xử lý dữ liệu điều hướng khi xe đứng yên, khi xe bắt đầu tăng tốc ta sẽ gửi dữ liệu điều hướng này đến hệ thống điều khiển trung tâm, tức là trong khoảng thời gian từ 15s đến khi dừng lại là 50s. Tín hiệu điều hướng trong khoảng thời gian này ban đầu đi thẳng, để giây thứ 31 tín hiệu sẽ phải và sau đó tiếp tục đi thẳng. Trong khoảng thời gian này hệ thống điều hướng khá chính xác, xử lý được độ trễ của tín hiệu GPS. Tuy nhiên ở những giây cuối do tác động nhiều đến tín hiệu GPS dẫn đến xác định hướng di chuyển hiện tại không chính xác, dẫn đến tín hiệu điều hướng chưa chính xác. Sau đó, hệ thống phát hiện vị trí hiện tại gần điểm đích và kết thúc quá trình di chuyển.

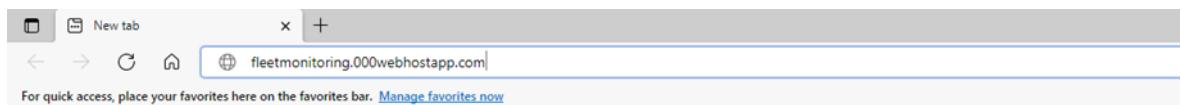
7.4. Kết quả thực hiện server và cơ sở dữ liệu

7.4.1. Kết quả xây dựng server

Xây dựng server thành công trên máy chủ do 000webhost cung cấp với tên miền là “*fleetmonitoring.000webhostapp.com*”. Khi phía client là các trình duyệt hay các vi điều khiển gửi yêu cầu đến server thì sẽ có những phản hồi trả về tương ứng.

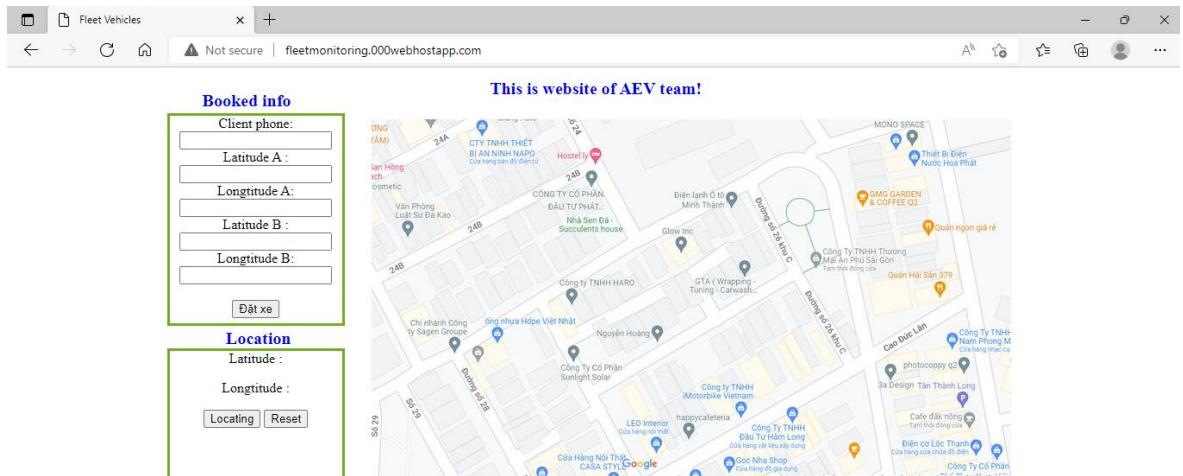
Ví dụ 1:

Trình duyệt Microsoft Edge gửi yêu cầu đến server qua URL: *fleetmonitoring.000webhostapp.com*.



Hình 7-25 Trình duyệt gửi yêu cầu đến server qua URL

Server nhận được và phản hồi lại:



Hình 7-26 Kết quả trả về từ server là trang web có tên Fleet Vehicles

Ví dụ 2:

Thực hiện GET dữ liệu vị trí của các xe tự hành từ server qua mô-đun 4G

```

AT+HTTPINIT

OK

AT+HTTPPARA="URL","https://fleetmonitoring.000webhostapp.com/searchGPSdata.php?frame_ID=1"

OK

AT+HTTPACTION=0

OK

+HTTPACTION: 0,200,99

AT+HTTPREAD=0,99

OK

+HTTPREAD: 99
{"frame_ID":"1","car_gps_ID":"2","latitude":"1047.96435","longitude":"10644.59685","velocity":"0"}
+HTTPREAD: 0

```

Hình 7-27 Thực hiện GET dữ liệu từ server và nhận được phản hồi

Trong ví dụ trên, người dùng thực hiện gửi các yêu cầu để lấy dữ liệu vị trí với \$key là *frame_ID* = 1, yêu cầu này được server xử lý, truy cập vào CSDL và trả về phản hồi là gói tin có \$key *frame_ID*=1, đó là

```
{"frame_ID":"1","car_gps_ID":"2","latitude":"1047.96435","longitude":"10644.59685","velocity":"0"}
```

Đánh giá, nhận xét

Qua các ví dụ trên, cho thấy server đã xây dựng thành công, khi các client gửi các request (yêu cầu) đến, thì server sẽ kiểm tra và trả về phản hồi.

7.4.2. Kết quả xây dựng cơ sở dữ liệu

Các bảng dữ liệu thực hiện lưu các dữ liệu tương ứng về thông tin hoạt động của xe, các dữ liệu của trung tâm giám sát, cũng như các dữ liệu của khách hàng.

Bảng 7-4 Bảng gps_data_list lưu dữ liệu GPS

frame_ID	car_gps_ID	latitude	longitude	velocity	timestamp
1	2	1045.57040	10639.64824	0	2022-05-19 18:25:33
2	2	1045.56949	10639.64343	0	2022-05-19 18:25:37
3	2	1045.57045	10639.64248	0	2022-05-19 18:25:41
4	2	1045.57375	10639.64726	0	2022-05-19 18:25:45
5	2	1045.57585	10639.64956	0	2022-05-19 18:25:49
6	2	1045.57645	10639.64971	0	2022-05-19 18:25:53
7	2	1045.57731	10639.65109	0	2022-05-19 18:25:57
8	2	1045.57864	10639.65360	0	2022-05-19 18:26:01
9	2	1045.57914	10639.65448	0	2022-05-19 18:26:05
10	2	1045.57879	10639.65343	0	2022-05-19 18:26:09

Bảng 7-5 Bảng fleet_info lưu thông tin đầy đủ hoạt động của đội xe

car_ID	car_status	car_speed	car_position	car_task	car_power	timestamp
1	ON	15	10.45124, 106.76541	Test Update	80	2022-05-19 18:34:32
2	ON	13	10.45221, 106.76536	Delivery	65	2022-05-19 18:34:32
3	OFF	0	10.15642, 106.65432	None	60	2022-05-21 15:12:16
5	ON	17	10.32642, 106.75411	Delivery	90	2022-05-21 15:12:16

Bảng 7-6 Bảng booked_info lưu thông tin của khách hàng

guest_num	lat_a	lon_a	lat_b	lon_b	timestamp
0384370746	10.79745	106.74354	10.79758	106.74502	2022-05-19 18:38:27
0384370746	10.79790	106.74331	10.79812	106.74478	2022-05-19 18:38:27
0384370746	10.79749	106.74397	10.79772	106.74434	2022-05-19 18:38:27
0384370746	10.79767	106.74496	10.79720	106.74436	2022-05-19 18:38:27
0327196891	10.79581	106.74456	10.79668	106.74430	2022-05-19 18:38:27
0384370746	10.7600186	24680.02468	9753.95731	86420.08642	2022-05-19 18:38:27
0384370746	10.79602	106.74301	10.79602	106.74323	2022-05-21 13:51:06

Cơ sở dữ liệu được tổ chức có hệ thống giúp người vận hành dễ dàng theo dõi cũng như vận hành hoạt động của xe và đội xe tự hành.

7.5. Kết quả thực hiện server kết nối dữ liệu

7.5.1. Thực nghiệm kiểm tra tốc độ truyền nhận dữ liệu

Tiến hành POST dữ liệu mặc định liên tục và xem thời gian cơ sở dữ liệu cập nhật

Bảng 7-7 Bảng gps_data_list lưu dữ liệu POST gói tin mặc định liên tục

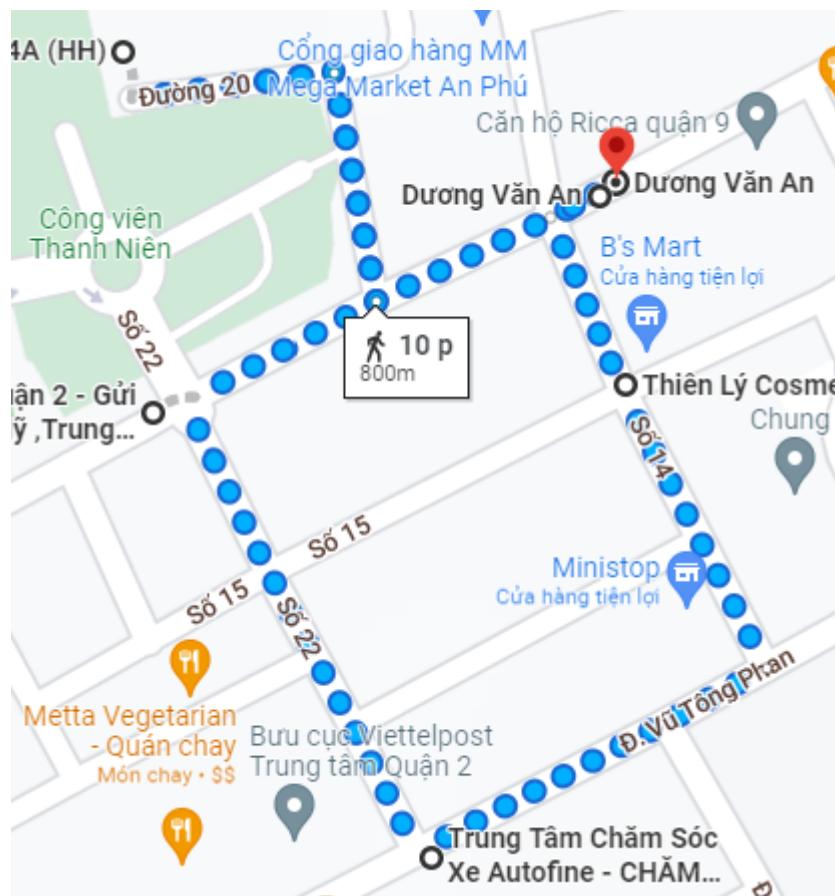
frame_ID	car_gps_ID	latitude	longitude	velocity	timestamp
1	2	1045.57040	10639.64824	0	2022-05-19 18:25:33
2	2	1045.56949	10639.64343	0	2022-05-19 18:25:37
3	2	1045.57045	10639.64248	0	2022-05-19 18:25:41
4	2	1045.57375	10639.64726	0	2022-05-19 18:25:45
5	2	1045.57585	10639.64956	0	2022-05-19 18:25:49
6	2	1045.57645	10639.64971	0	2022-05-19 18:25:53
7	2	1045.57731	10639.65109	0	2022-05-19 18:25:57
8	2	1045.57864	10639.65360	0	2022-05-19 18:26:01
9	2	1045.57914	10639.65448	0	2022-05-19 18:26:05
10	2	1045.57879	10639.65343	0	2022-05-19 18:26:09

Dữ liệu đã được cập nhật chứng tỏ đã thực hiện thành công kết nối giữa server và dữ liệu GPS được gửi lên qua mô-đun 4G.

7.5.2. Thực hiện gửi dữ liệu hoạt động của xe lên cơ sở dữ liệu qua mô-đun 4G

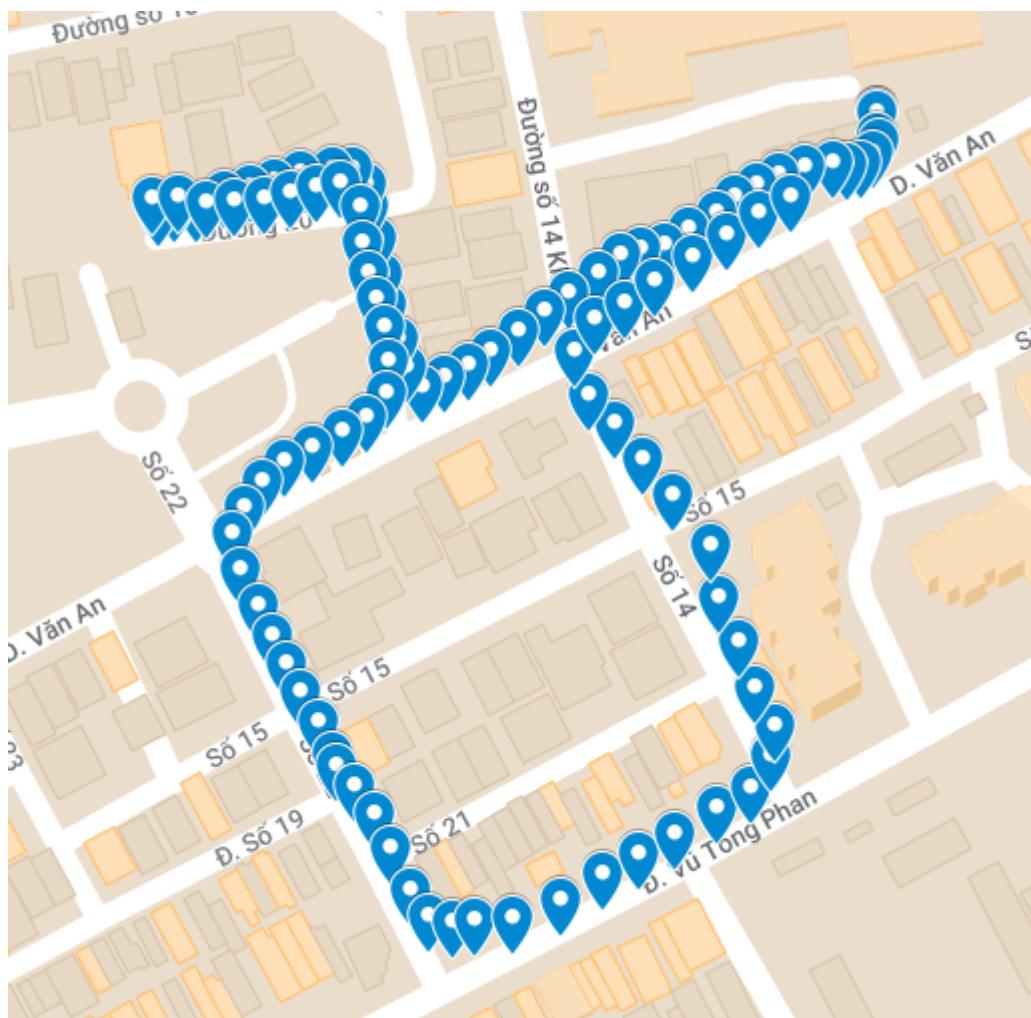
❖ Khảo sát 1

Gắn hộp định vị lên xe và di chuyển theo quỹ đạo được xác định trên Google Maps như hình bên dưới, thiết lập mô-đun 4G ở trạng thái POST dữ liệu để đẩy dữ liệu lên database. Ở đây nhằm mục đích đánh giá khả năng gửi dữ liệu lên server của mô-đun 4G trong một khoảng thời gian xác định.



Hình 7-28 Đoạn đường di chuyển khi khảo sát

Đoạn đường di chuyển với dữ liệu từ database có được từ quá trình POST dữ liệu của mô-đun khi di chuyển theo đoạn đường trên.



Hình 7-29 Kết quả dựng lại lộ trình di chuyển bằng công cụ Google MyMaps với dữ liệu từ database – bảng gps_data_list

Bảng 7-8 Bảng dữ liệu lưu vị trí của xe trong khảo sát 1

Point	Frame	a	b	Lat	Lon
1	1;"2";"1047.96435";"10644.59685";"0"	1047.964	10644.6	10.79941	106.7433
2	2;"2";"1047.96427";"10644.59684";"0"	1047.964	10644.6	10.7994	106.7433
3	3;"2";"1047.96408";"10644.59679";"0"	1047.964	10644.6	10.7994	106.7433

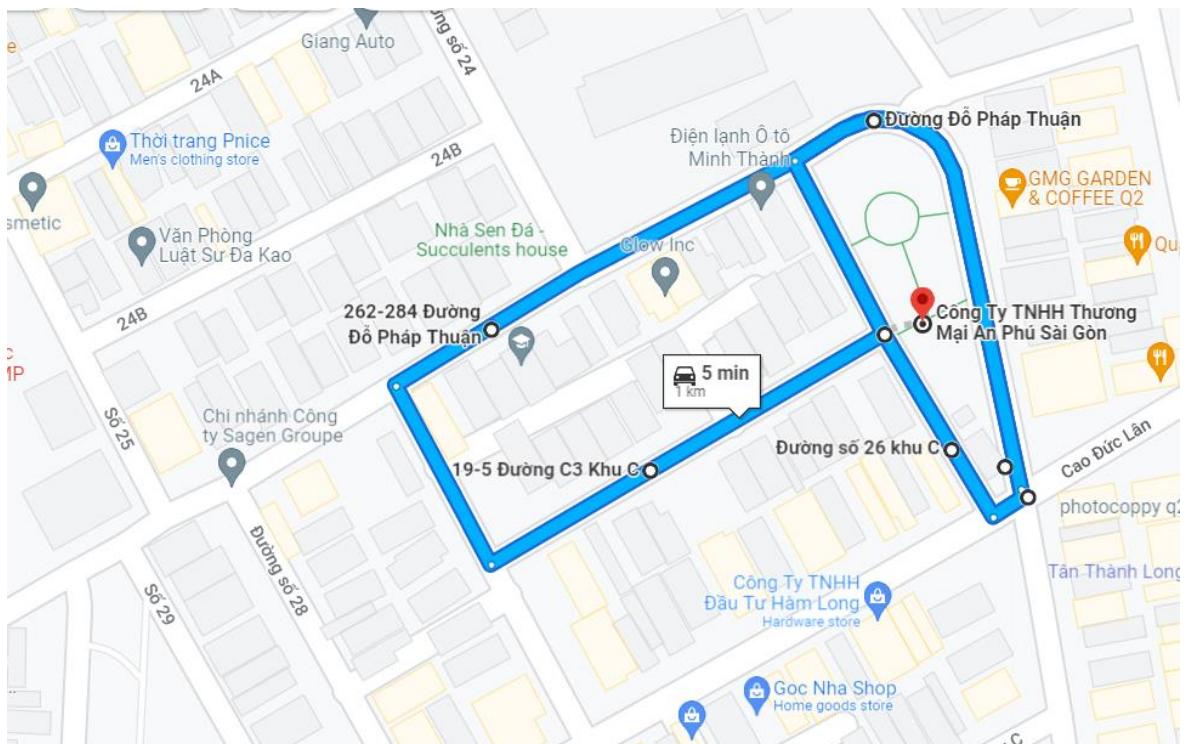
109	109;"2";"1047.95541";"10644.59728";"0"	1047.955	10644.6	10.79926	106.7433
110	110;"2";"1047.95559";"10644.59726";"0"	1047.956	10644.6	10.79926	106.7433
111	111;"2";"1047.95582";"10644.59727";"0"	1047.956	10644.6	10.79926	106.7433

Trong khoảng thời gian 7 phút, môđun gửi được 111 gói tin đến SERVER và thực hiện lưu vào cơ sở dữ liệu.

$$t = \frac{7 \times 60}{111} = 3,784 \text{ (s)}$$

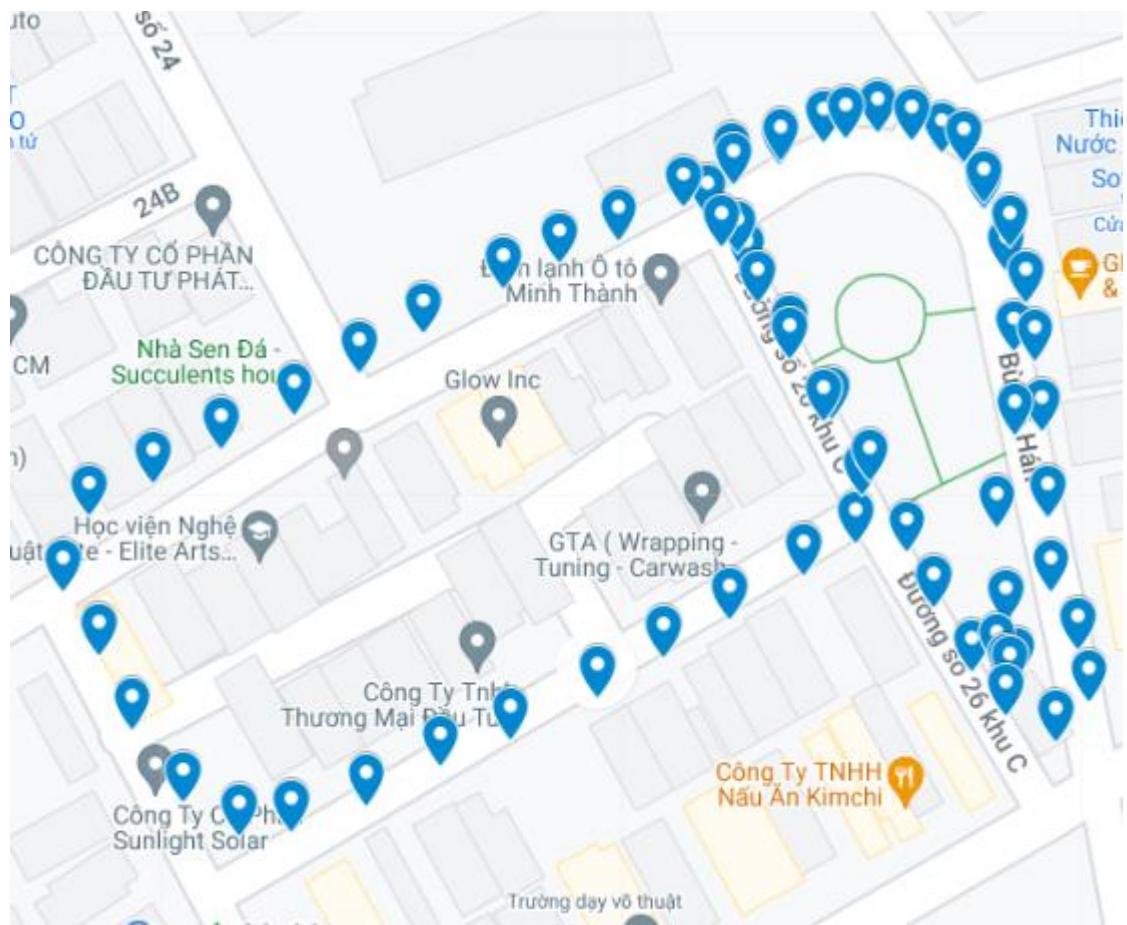
❖ *Khảo sát 2*

Thực hiện tương tự như khảo sát 1 với một cung đường khác.



Hình 7-30 Lộ trình di chuyển trên Google Maps

Thực hiện dựng lại đoạn đường di chuyển bằng công cụ Google MyMaps từ dữ liệu trên database có được do quá trình POST dữ liệu của mô-đun khi di chuyển theo đoạn đường trên.



Hình 7-31 Kết quả dựng lại đoạn đường di chuyển với Google MyMaps

Bảng 7-9 Bảng dữ liệu lưu vị trí của xe trong khảo sát 2

frame_ID	car_gps_ID	latitude	longitude	velocity	timestamp
1	2	1047.75772	10644.66439	0	2022-05-22 02:45:05
2	2	1047.75595	10644.66513	0	2022-05-22 02:45:08
3	2	1047.75577	10644.66555	0	2022-05-22 02:45:12
4	2	1047.75653	10644.66619	0	2022-05-22 02:45:16

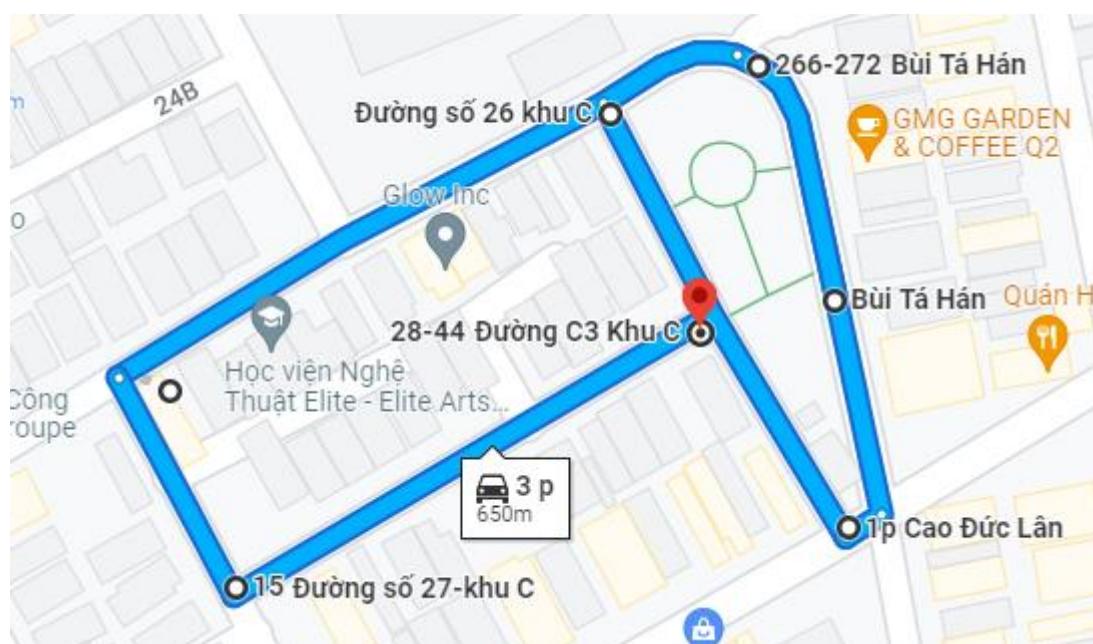
78	2	1047.77280	10644.66808	0	2022-05-22 02:50:05
79	2	1047.76641	10644.66929	0	2022-05-22 02:50:09
80	2	1047.76066	10644.67064	0	2022-05-22 02:50:13
81	2	1047.75650	10644.67090	0	2022-05-22 02:50:16

Kết quả dựng lại lộ trình đã đi từ dữ liệu trên database có được do quá trình POST dữ liệu GPS qua mô-đun 4G khá giống với quỹ đạo thực tế đã di chuyển. Thời gian cập nhật giữa các gói tin trung bình là 4 giây.

7.5.3. Thực hiện giám sát lộ trình di chuyển của xe từ GUI giám sát trung tâm

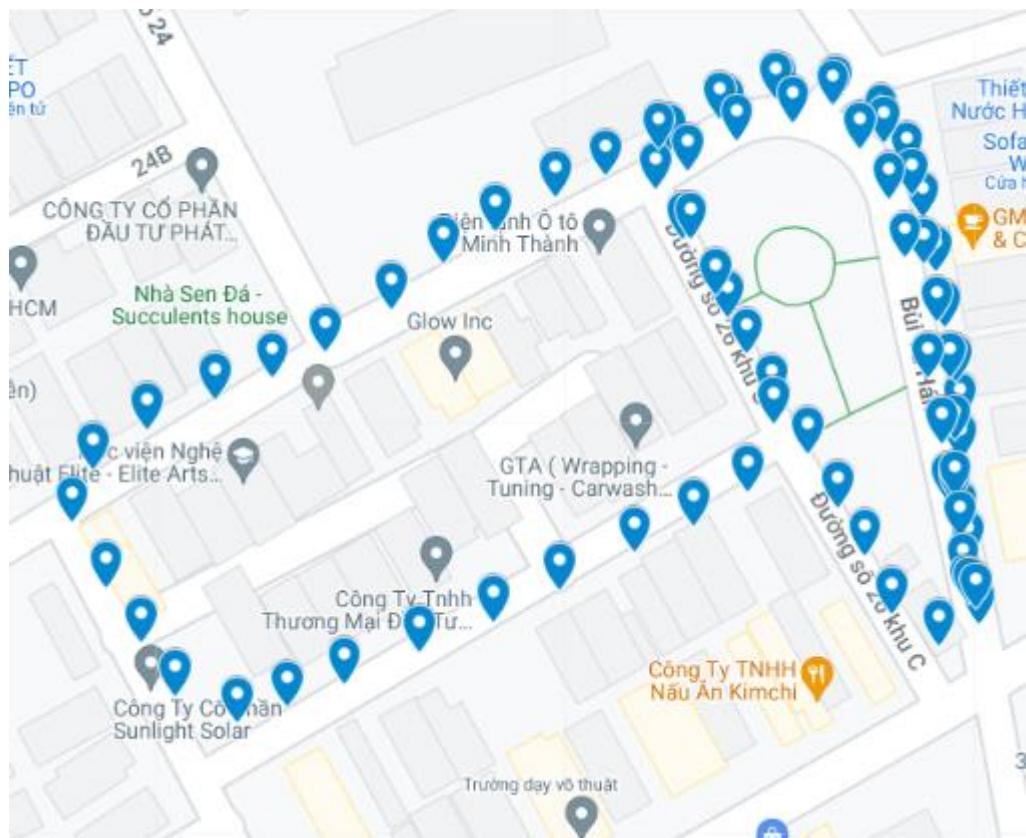
❖ *Khảo sát 3*

Thực hiện tương tự *khảo sát 1,2* với cung đường như ở *khảo sát 2* nhưng có một điểm khác là quá trình POST, GET được diễn ra song song. Mô-đun GPS-4G được gắn trên xe sẽ liên tục POST vị trí hiện tại của xe lên server, đồng thời GUI ở trung tâm giám sát liên tục thực hiện GET dữ liệu từ server về để giám sát.



Hình 7-32 Cung đường thực hiện quá trình Post Get song song

Thực hiện dựng lại đoạn đường di chuyển bằng công cụ Google MyMaps từ dữ liệu trên database có được do quá trình POST dữ liệu của mô-đun khi di chuyển theo đoạn đường trên.

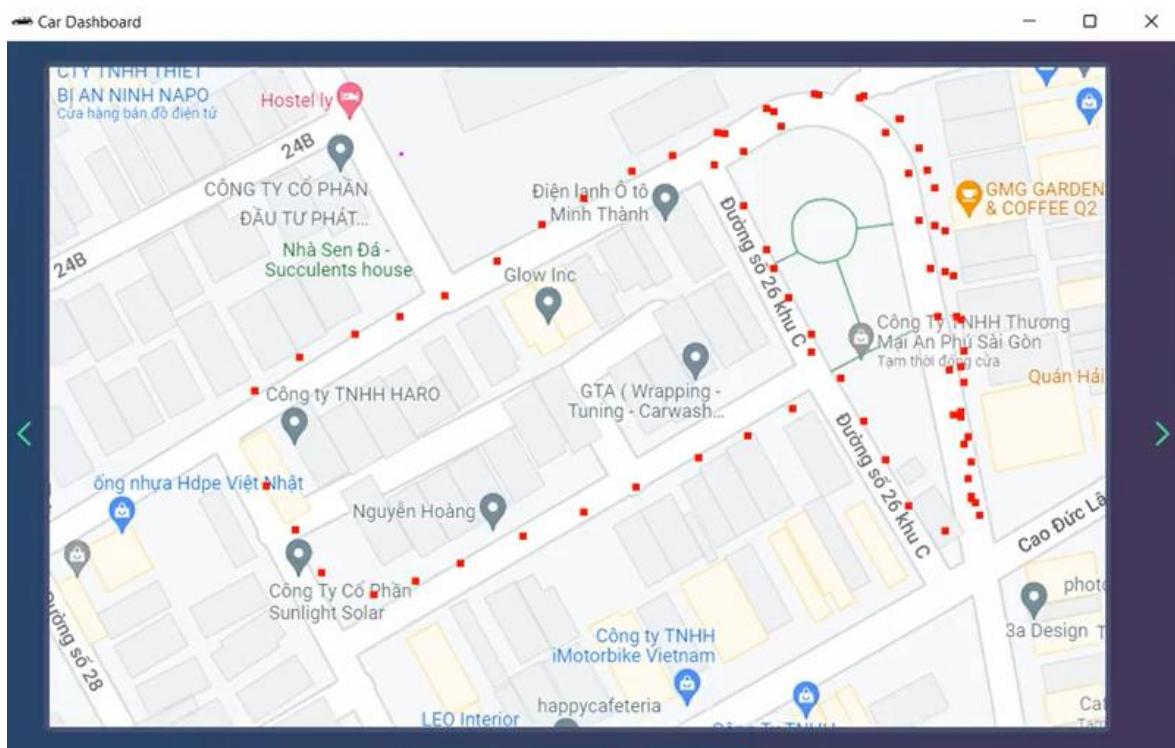


Hình 7-33 Kết quả dựng lại đoạn đường di chuyển với Google MyMaps

Bảng 7-10 Bảng dữ liệu lưu vị trí của xe trong khảo sát

frame_ID	car_gps_ID	latitude	longitude	velocity	timestamp
1	2	1047.76235	10644.67066	0	2022-05-22 03:10:01
2	2	1047.76515	10644.66923	0	2022-05-22 03:10:05
3	2	1047.76900	10644.67057	0	2022-05-22 03:10:09
4	2	1047.77298	10644.67052	0	2022-05-22 03:10:12
.....					
75	2	1047.75714	10644.67076	0	2022-05-22 03:14:52
76	2	1047.75507	10644.67118	0	2022-05-22 03:14:56
77	2	1047.75481	10644.67141	0	2022-05-22 03:15:00
78	2	1047.75430	10644.67201	0	2022-05-22 03:15:03

Thực hiện GET dữ liệu về từ database hiển thị lên giao diện giám sát trung tâm, quá trình này được thực hiện song song với quá trình xe di chuyển và gửi dữ liệu về vị trí và trạng thái hoạt động lên server.



Hình 7-34 Giao diện giám sát trung tâm

Trên giao diện giám sát trung tâm, thể hiện đầy đủ các tọa độ được cập nhật lên cơ sở dữ liệu (các điểm màu đỏ), do dữ liệu từ xe được gửi lên với tốc độ chậm (4 giây cho mỗi lần gửi) nên các điểm thể hiện khá thưa trên giao diện.

Đánh giá nhận xét:

Quá trình cập nhật dữ liệu vị trí cũng như các trạng thái hoạt động của xe điện về lưu trữ trên CSDL qua mô-đun 4G hoạt động ổn định. Các gói tin được truyền nhận đúng, bảo đảm thông tin.

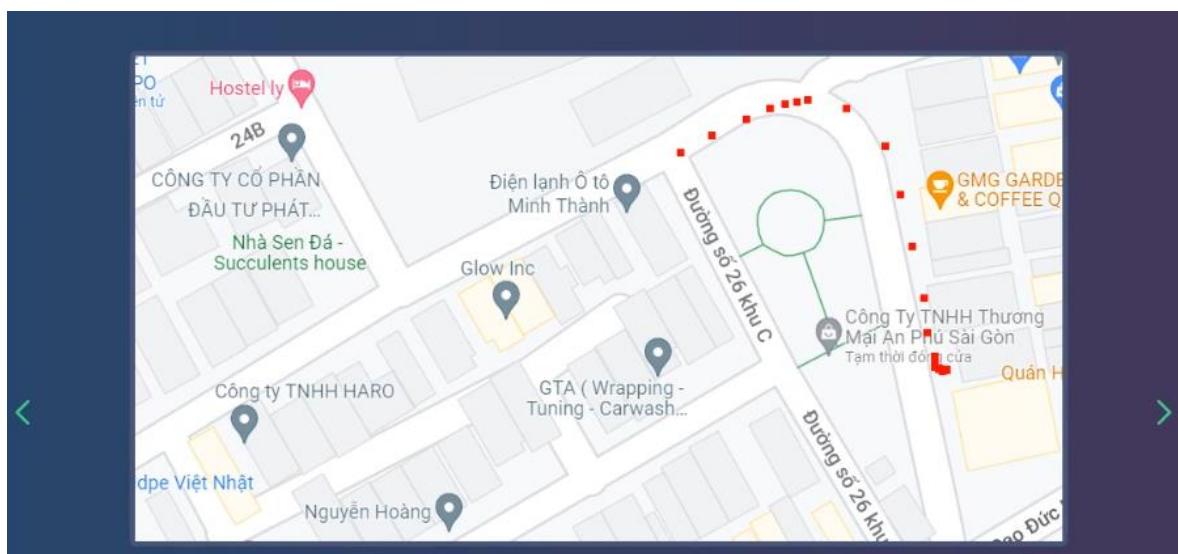
Kết quả dựng lại lộ trình đã đi trên giao diện giám sát với dữ liệu được lưu trên database có được do quá trình POST dữ liệu GPS qua module 4G khá giống với quỹ đạo thực tế đã di chuyển.

Tuy nhiên, khoảng thời gian giữa các lần POST dữ liệu lên server qua mô-đun 4G trung bình là 4 giây, là khá chậm so với tốc độ di chuyển của xe điện cho nên vị trí các điểm thể hiện trên giao diện giám sát hay khi dựng lại lộ trình di chuyển bằng Google MyMaps còn khá thưa. Điều này cho thấy những hạn chế

của giao thức HTTP trong các ứng dụng thời gian thực. Nhược điểm này có thể được chấp nhận trong mục tiêu giám sát hoạt động, tuy nhiên, để có thể phát các lệnh điều khiển từ xa đến xe, cần đòi hỏi sử dụng một giao thức có độ trễ thấp hơn.

7.6. Kết quả thực hiện giao diện web cho người sử dụng

7.6.1. Giao diện giám sát cho người vận hành



Hình 7-35 Giao diện giám sát ở trung tâm vận hành

Trên giao diện giám sát dành cho người vận hành sẽ là vị trí của xe trên bản đồ hiện tại (những điểm màu đỏ), nó cho phép người vận hành giám sát lộ trình của các xe, kiểm tra đội xe có thực hiện đúng nhiệm vụ không.

7.6.2. Giao diện web cho khách hàng đặt xe



Hình 7-36 Giao diện đặt xe cho khách hàng

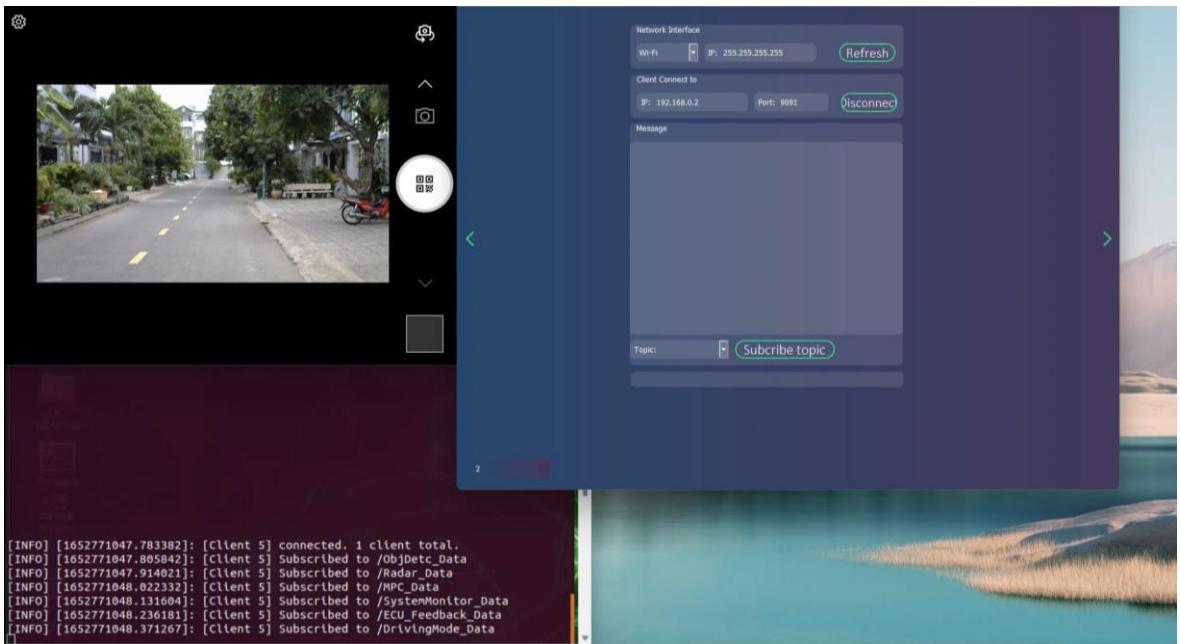
Khách hàng đặt xe sẽ có 2 cách để chọn điểm đón và điểm đến cho chuyến đi của mình:

- Nhập tọa độ điểm đón, điểm đến vào khung cùng với số điện thoại sau đó nhấn đặt xe. Số điện thoại để xác nhận người đặt xe là có thật bằng cách xác nhận một mã xác minh do server gửi về.
- Nhấp chuột trên bản đồ để chọn tọa độ 2 điểm sau đó nhấn đặt xe. Việc xác minh người đặt xe cũng được thực hiện tương tự.

Tại mục Location, người dùng có thể nhấn Locating để xem vị trí hiện tại của mình trên bản đồ.

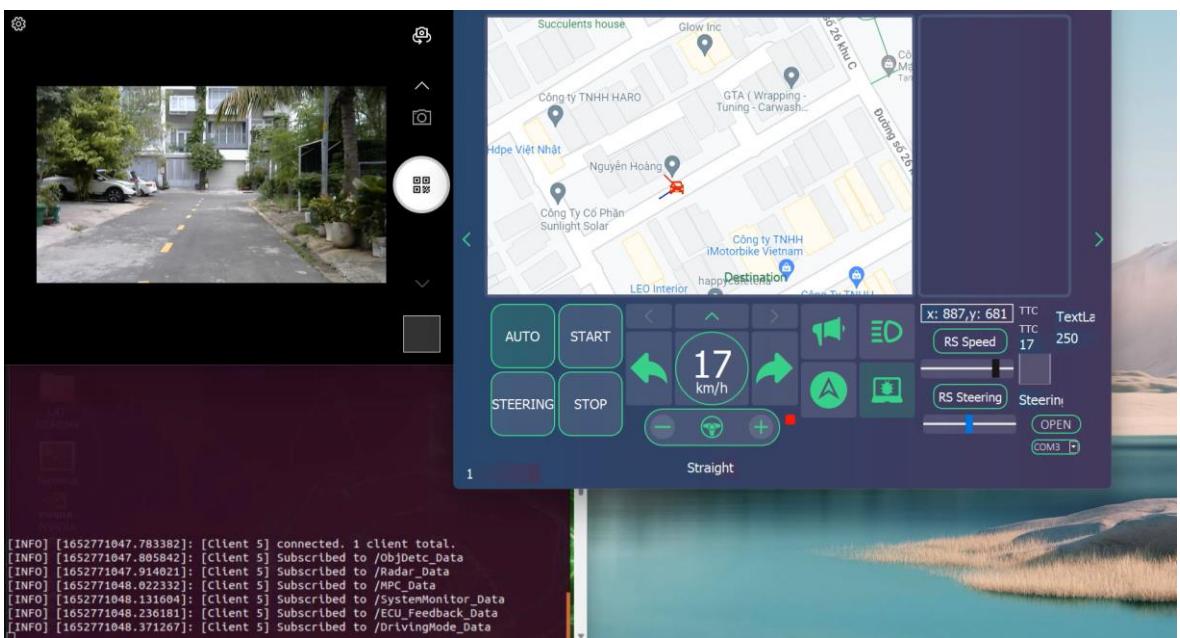
7.7. Kết quả ứng dụng lên xe điện tự hành

- ❖ Thực hiện kết nối TCP thành công với hệ thống điều khiển



Hình 7-37 Kết nối giao diện với hệ thống điều khiển

❖ Kết quả hiển thị vị trí và điều khiển vận tốc cho xe



Hình 7-38 Hiển thị vị trí hiện tại của xe

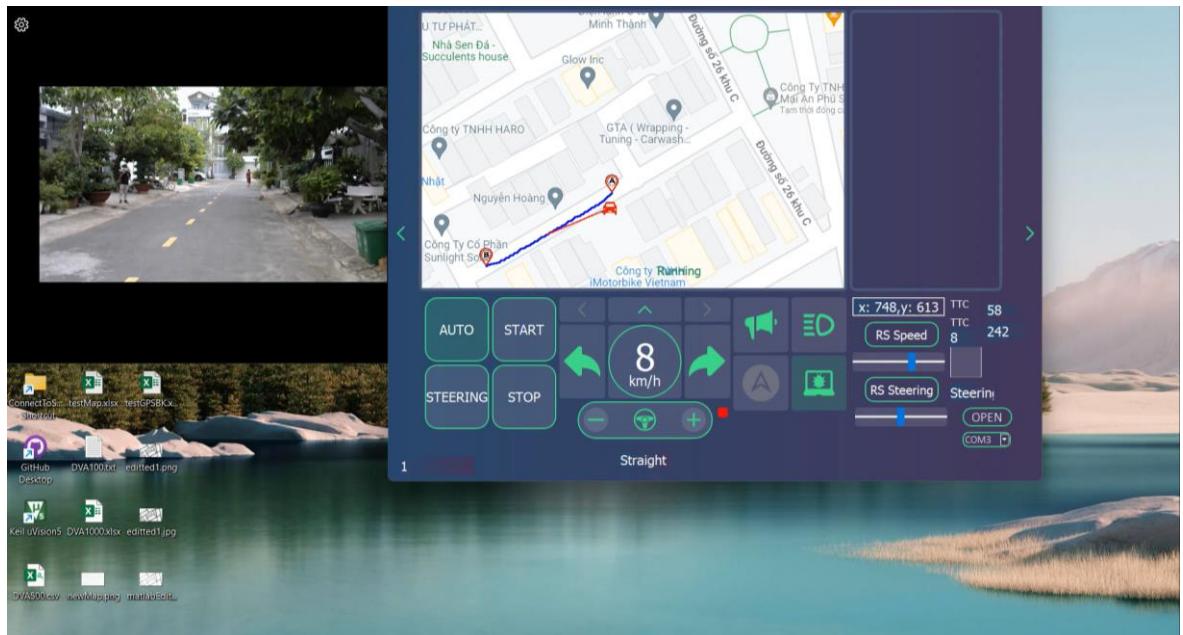
Đánh giá nhận xét

Kết nối thành công với hệ thống điều khiển thông qua giao thức TCP với thời gian lấy mẫu là 100ms.

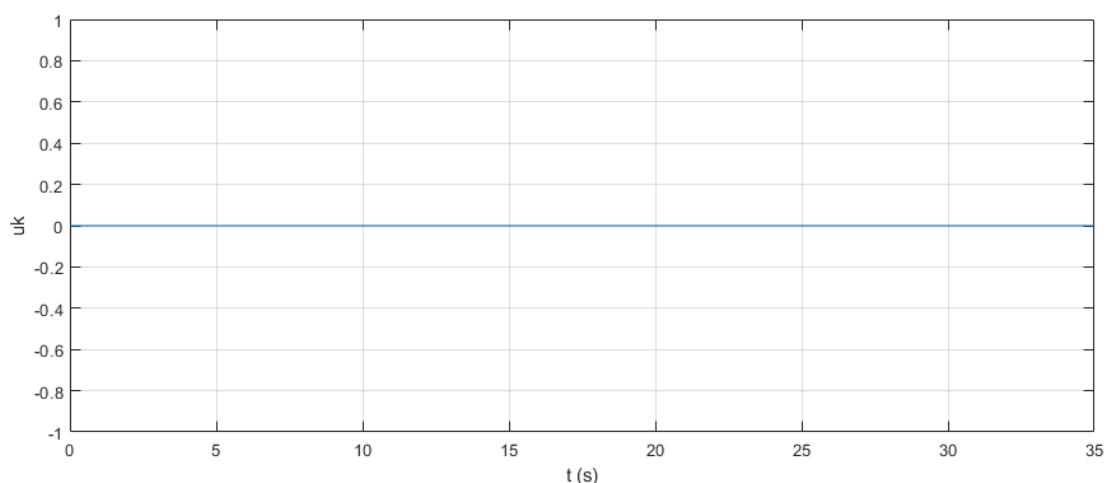
Giao diện hoạt động ổn định, hiển thị được vị trí di chuyển của phương tiện thu thập từ GPS. Tuy nhiên với tần số của GPS là 5Hz thì vị trí hiển thị bị trễ một khoảng so với vị trí thực tế vì xe di chuyển với tốc độ nhanh có thể lên đến 20km/h. Vì vậy khi thực hiện cho xe tự chạy theo quỹ đạo hoạch định bởi A* cần cho xe dừng trước điểm đích khoảng 2 đến 3m để đảm bảo an toàn. Theo dữ liệu của data sheet thì GPS có thể trả về dữ liệu với tần số lên đến 10Hz, tuy nhiên khi khảo sát thực nghiệm với chu kỳ này, dữ liệu trả về bị nhiễu và không được ổn định, gây ảnh hưởng đến việc xác định phương hướng đang chuyển động của xe.

❖ *Hệ thống điều hướng xe đi theo quỹ đạo được hoạch định từ thuật toán A**

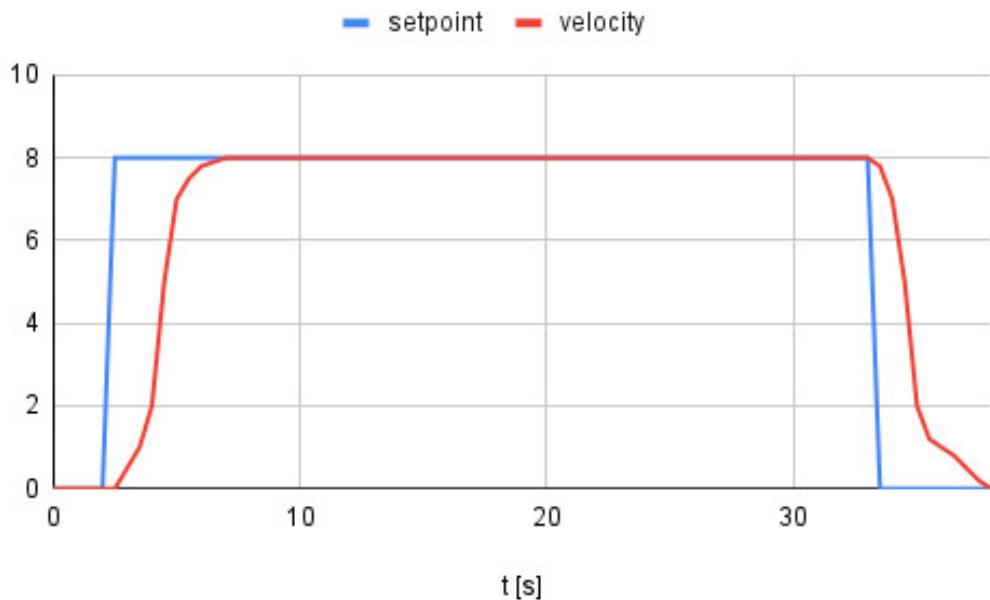
Khi tương tác với giao diện điều khiển trên xe, người dùng thực hiện chọn lộ trình di chuyển cho xe bằng cách chọn điểm bắt đầu và điểm kết thúc. Thuật toán A* hoạch định đường đi tối ưu cho xe di chuyển. Khi đã hoạch định xong lộ trình, xe chuyển sang chế độ *running* và di chuyển theo các tín hiệu điều hướng và vận tốc do GUI điều khiển gửi xuống. Khi tiếp cận điểm kết thúc trong bán kính 2 – 3 mét thì GUI gửi xuống tín hiệu điều khiển vận tốc về 0 và cho xe dừng lại (chuyển sang *stop*).



Hình 7-39 Điều hướng xe chạy theo quỹ đạo A *



Hình 7-40 Tín hiệu điều hướng của quá trình di chuyển



Hình 7-41 Đồ thị thay đổi vận tốc trong quá trình di chuyển

Đánh giá, nhận xét:

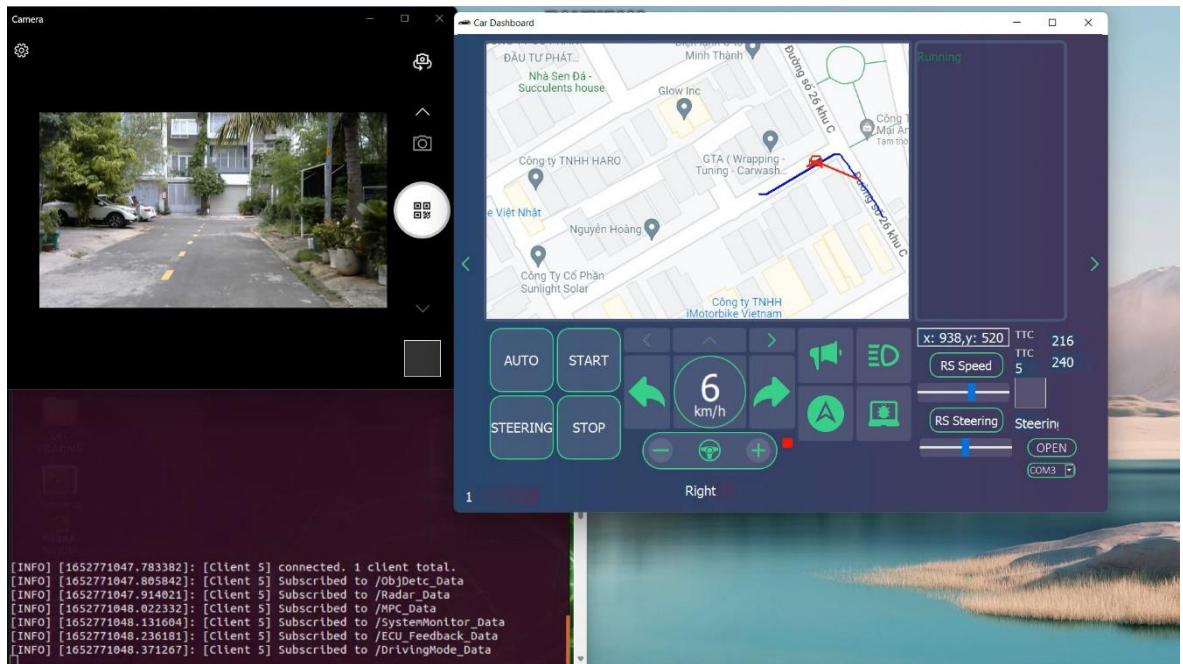
Giải thuật hoạch định quỹ đạo cho kết quả tốt trong quá trình chuyển động. Với quỹ đạo là đường thẳng, hệ thống gửi tín hiệu điều hướng $uk = 0$ (đi thẳng) trong suốt quá trình với vận tốc là 8 km/h và giảm nhanh khi gần tới điểm đích và dừng lại.

- ❖ Kết hợp hệ thống hoạch định quỹ đạo và vận hành từ xa vào quá trình hoạt động chung của xe

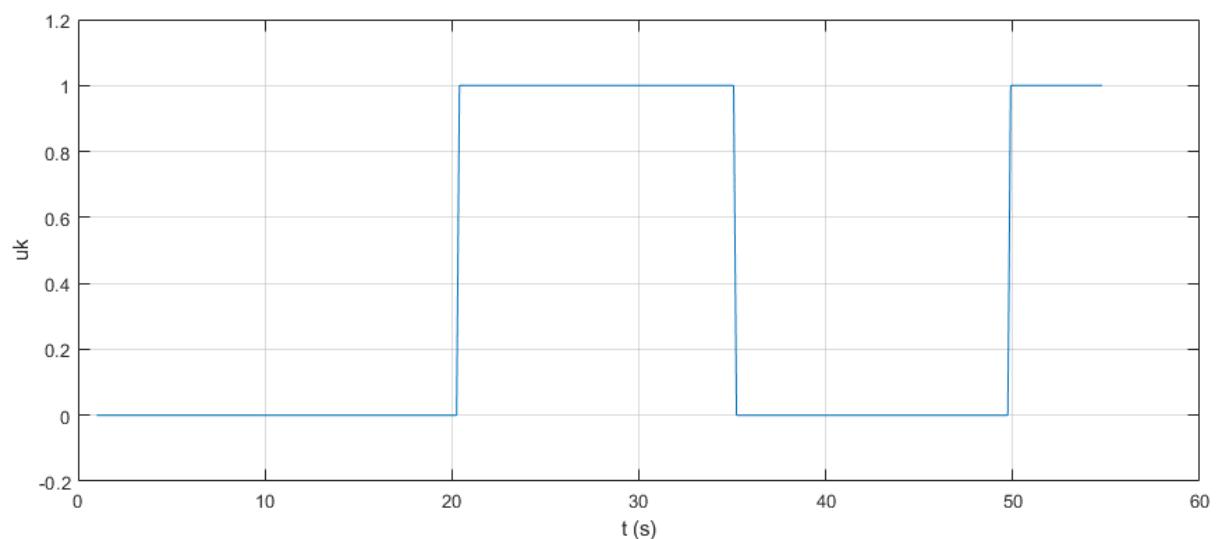


Hình 7-42 Chạy theo quỹ đạo được hoạch định từ thuật toán A* dựa trên dữ liệu từ server

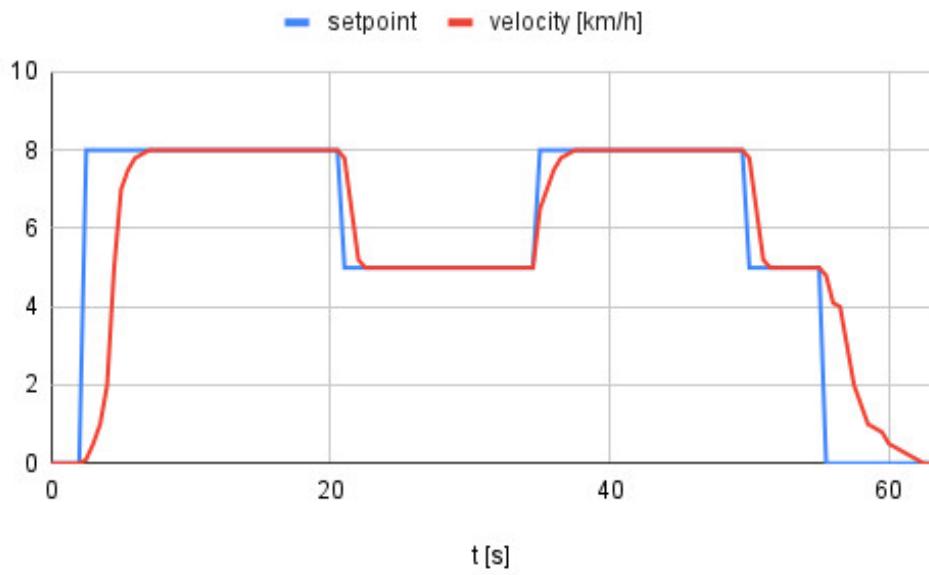
Sau khi khởi động, hệ thống thực hiện GET để kiểm tra dữ liệu đặt xe từ CSDL, khi nhận được thông tin có người dùng đặt xe từ web, hệ thống sẽ hoạch định quỹ đạo dựa trên các thông tin về điểm bắt đầu và điểm kết thúc, sau đó xe bắt đầu di chuyển và thực hiện POST dữ liệu về vị trí và các trạng thái hoạt động lên server. Sau khi di chuyển hết quỹ đạo và dừng lại tại điểm kết thúc, xe chuyển về trạng thái GET để tiếp tục nhận lượt đặt xe tiếp theo.



Hình 7-43 Hệ thống Get dữ liệu đặt xe và di chuyển hết quỹ đạo



Hình 7-44 Tín hiệu điều hướng của quá trình di chuyển



Hình 7-45 Đồ thị vận tốc trong quá trình di chuyển

Đánh giá, nhận xét:

Hệ thống đã nhận được dữ liệu từ server, dựng được quỹ đạo và tự động di chuyển hết quỹ đạo đã hoạch định.

Hệ thống hoạch định quỹ đạo hoạt động tốt, các tín hiệu điều hướng phù hợp với đoạn đường đang đi, từ đó xuất ra các tín hiệu điều khiển vận tốc phù hợp: tăng tốc nếu là đoạn đường thẳng và hệ thống đang điều hướng đi thẳng; giảm tốc khi có tín hiệu điều hướng rẽ trái hay rẽ phải. Tuy nhiên, tại vị trí gần điểm kết thúc, do nhiễu tín hiệu GPS dẫn đến việc xác định hướng di chuyển của xe bị sai lệch, khiến tín hiệu điều hướng không chính xác (rẽ phải thay vì đi thẳng).

Các tín hiệu điều hướng và điều khiển vận tốc từ hệ thống hoạch định quỹ đạo là các thông tin quan trọng trong việc hỗ trợ bộ điều khiển trung tâm (bộ điều khiển MPC) điều khiển chính xác các tín hiệu góc đánh lái, vận tốc để thực hiện đúng lộ trình cần đi và di chuyển đúng làn đường, đặc biệt là tại các vị trí ngã ba, ngã tư.

Chương 8. ĐÁNH GIÁ, KẾT LUẬN, HƯỚNG PHÁT TRIỂN ĐỀ TÀI

8.1. Đánh giá kết quả

❖ Phần cứng

Hộp điện cho hệ thống mô-đun GPS-4G hoạt động ổn định và thực hiện đúng các chức năng như yêu cầu. Kết nối giữa các phần trong hệ thống chắc chắn và dễ dàng theo dõi mạch. Thiết kế bên ngoài nhỏ gọn, dễ lắp đặt, cũng như tháo lắp, dễ tích hợp vào các hệ thống phần cứng khác.

Mô-đun GPS thực hiện tốt vai trò cập nhật vị trí của xe trong các khu vực thực hiện khảo sát quá trình hoạt động của xe điện tự hành; tạo được tập dữ liệu vị trí giúp xác định hướng di chuyển của xe. Tuy nhiên, tốc độ cập nhật dữ liệu từ mô-đun GPS còn khá hạn chế (5 Hz), trong khi tốc độ của xe có thể lên đến 10-15 km/h, điều này dẫn đến độ trễ trong việc điều hướng cho xe.

Mô-đun 4G hoạt động ổn định và dễ sử dụng. Quá trình truyền nhận liên tục dữ liệu tiêu tốn rất ít dung lượng 4G.

❖ Phần mềm

Phát triển được công cụ thực hiện số hóa bản đồ cho một khu vực nhất định, cung cấp các thông tin chi tiết về địa hình, giao thông của khu vực đó, đặc biệt là những chi tiết mà Google Maps không cung cấp.

Thuật toán tìm đường A* giúp hoạch định được quỹ đạo di chuyển tối ưu từ những đường đi đơn giản đến phức tạp, hỗ trợ xe di chuyển trong khu vực cục bộ. Trong một số trường hợp, quỹ đạo được hoạch định ra còn chưa được trơn tru, cần được cải thiện thêm.

Thiết kế được giao diện điều khiển xe, hoạt động ổn định, trên đó thể hiện ví trí của xe, và lộ trình di chuyển mong muốn. Người dùng cũng có thể tác động đến các tín hiệu điều khiển của xe thông qua các nút nhấn, thanh trượt... được tích hợp trên giao diện.

Thiết lập được kết nối giữa với xe và trung tâm giám sát qua mạng di động 4G. Xây dựng được cơ sở dữ liệu để lưu trữ dữ liệu hoạt động của xe và đội xe tự hành. Xây dựng được server hoạt động ổn định, xử lý tốt các truy vấn từ client và trả về phản hồi đúng như được thiết lập. Cơ sở dữ liệu hoạt động đúng với yêu cầu khi giao tiếp tốt với các truy vấn dữ liệu từ server. Tuy nhiên đối với tốc độ truyền nhận dữ liệu từ xe về server còn khá chậm, do đặc điểm mô hình request (yêu cầu)/response (phản hồi) của mô hình HTTP. Mặc dù, điều này là chấp nhận được trong mục tiêu của đề tài là giám sát hoạt động, nhưng về phát triển lâu dài, điều này phải được cải thiện.

Thiết kế được giao diện cho người vận hành đội xe tại trung tâm giám sát và vận hành. Trên đó, người vận hành có thể theo dõi đầy đủ các thông tin về hoạt động của các xe điện tử hành, các thông tin về cơ sở dữ liệu, các yêu cầu đặt xe của khách hàng, từ đó có thể thực hiện các gửi các yêu cầu điều khiển hoạt động xuống cho đội xe. Ngoài ra, một giao diện web dành cho người sử dụng là khách hàng thực hiện đặt xe. Trên đó người dùng sẽ cung cấp các thông tin đặt xe, theo dõi vị trí xe mình đặt, cũng như vị trí hiện tại của khách hàng. Song, các giao diện còn khá đơn giản về mặt hình thức, chưa thực sự thân thiện với người dùng, đây là một số hạn chế mà phần thiết kế giao diện còn vướng phải.

8.2. Kết luận đề tài

Tính đến thời điểm hiện tại, chúng em đã hoàn thành được những mục tiêu đặt ra khi bắt đầu thực hiện đề tài: thiết lập thành công kết nối dữ liệu giữa server và xe tự hành để có thể giám sát và vận hành hoạt động của xe từ xa thông qua mạng 4G. Áp dụng thuật toán A*, công cụ số hóa bản đồ, cũng như xử lý được các thông tin về vị trí của xe từ GPS để hoạch định được quỹ đạo chuyển động tối ưu, điều hướng ổn định cho lộ trình di chuyển của xe, cung cấp các thông tin về hướng di chuyển giúp bộ điều khiển MPC thực hiện bám làn

đường đặc biệt tại các khu vực ngã ba ngã tư. Bên cạnh đó vẫn còn một số hạn chế cần phải được khắc phục và phát triển thêm như đã nêu ở trên.

8.3. Hướng phát triển đề tài

Sau khi hoàn thành đề tài, với kết quả hiện tại, thì những hướng cải thiện, phát triển tiếp theo sẽ là:

Đối với phần cứng, cần thay đổi thiết bị GPS hiện tại thành hệ thống với độ chính xác cao hơn để cung cấp thông tin có độ tin cậy cao hơn về vị trí của đối tượng, phục vụ hiệu quả giải thuật điều hướng cho xe, có thể tích hợp GPS/INS để có thể chạy ở mode GPS single. Nếu vẫn sử dụng những mô-đun GPS như hiện tại cần áp dụng các thuật toán nhằm nâng cao độ chính xác, xử lý được nhiều môi trường, để giảm lỗi cho máy thu GPS giá rẻ, Refan và Palangi đã đề xuất phương pháp Recursive Averaging and Autoregressive-moving Average (ARMA), và mô hình ARMA cho thấy hiệu suất khả quan [17].

Phát triển hệ thống camera giúp xây dựng bản đồ trực tuyến để xe có thể hoạt động ở những khu vực cục bộ khác nhau mà không cần phải thực hiện số hóa bản đồ trước đó. Nếu vẫn sử dụng phương pháp số hóa như hiện tại cần cập nhật thêm nhiều chi tiết về khu vực đó, đặc biệt chiều di chuyển cho phép của các con đường trên bản đồ. Nâng cấp thuật toán A * không chỉ tìm đường đi tối ưu mà phải đúng phần làn chuyển động của xe trên con đường đó, cải thiện quỹ đạo tối ưu, tạo quỹ đạo đường cong mượt khi hoạch định. Khi GPS đã tăng độ chính xác, quỹ đạo di chuyển động hoạch định đúng làn đường thì việc di chuyển của xe chỉ cần bám đúng theo quỹ đạo chuyển động đó và di chuyển đến đích.

Cải thiện tốc độ truyền nhận dữ liệu giữa vi điều khiển và server, giảm thời gian cần thiết cho mỗi gói tin POST hay GET đến server. Để có thể đạt được yêu cầu trên, giải pháp sẽ là sử dụng một giao thức khác tốt hơn cho việc truyền nhận dữ liệu giữa vi điều khiển và server, trong đó MQTT có thể sẽ là

một giải pháp phù hợp bởi những ưu điểm về tốc độ truyền tải, kích thước các gói tin, cũng như đặc điểm tiết kiệm băng thông trong các ứng dụng M2M [18].

Phát triển giao diện web tốt hơn tạo trải nghiệm thuận tiện cho người dùng và xa hơn sẽ là thiết kế một ứng dụng trên điện thoại để người dùng có thể dễ dàng truy cập và thao tác trên các thiết bị cầm tay.

Với kết quả hiện tại của đề tài, nhờ khả năng dễ tích hợp của hệ thống phần cứng, cũng như việc thiết lập phần mềm đơn giản, hướng phát triển nổi bật của đề tài chính là cho dự án quản lý đội xe tự hành (Fleet Management System).

TÀI LIỆU THAM KHẢO

- [1] [Online]. Available: <https://www.nuro.ai/technology>.
- [2] S. M. Kim, M. I. Peña, M. Moll, G. N. Bennett, and L. E. Kavraki, "A review of parameters and heuristics for guiding metabolic pathfinding," *J. Cheminform*, vol. 9, pp. 1-13, 2017.
- [3] I. Zaremba and S. Kodors, "Pathfinding Algorithm Efficiency Analysis in 2D Grid," *Environ. Technol. Resour. Proc. Int. Sci. Pract. Conf.*, vol. 2, p. 46, 2015.
- [4] Mahadevi S. 1 ,K. R. Shylaja 2 ,Ravinandan M. E, "Memory Based A-Star Algorithm for Path Planning of a Mobile Robot," *International Journal of Science and Research (IJSR)*, vol. 3, no. 6, p. 1352, 2014.
- [5] Zhengbing He, Liang Zheng, Peng Chen, Wei Guan, "Computer-Aided Civil and Infrastructure Engineering," *Mapping to cells: a simple method to extract traffic dynamics from probe vehicle data*, p. 1, 2017.
- [6] Luca Venturi, Krishtof Korda, in *Hands-On Vision and Behavior for Self-Driving Cars*, 2020, pp. 66-86.
- [7] Raj Kishen Moloo, Varun Kumar Digumber, "International Conference on Business Computing and Global Informatization," *Low-Cost Mobile GPS Tracking Solution*, pp. 516-519, 2011.
- [8] Debao Yuan, Ximin Cui, Guo Wang, Jingjing Jin, Wei Xue, *Research on Improving the Accuracy of Positioning Data Obtained by GPS Module*, vol. 500, pp. 610 - 6015, 2012.
- [9] Md. Rashedul Islam and Jong-Myon Kim, "Hindawi Publishing Corporation The Scientific World Journal," *An Effective Approach to Improving Low-Cost*

GPS Positioning Accuracy in Real-Time Navigation, 2014.

- [10] www.u-blox.com, u-blox 7 Receiver Description, 2018.
- [11] Junfeng Yao, Chao Lin , Xiaobiao Xie, Andy JuAn Wang, Chih-Cheng Hung, "Seventh International Conference on Information Technology," *Path Planning for Virtual Human Motion Using Improved A* Algorithm*, p. 1156, 2010.
- [12] Junfeng Yao, Chao Lin , Xiaobiao Xie, Andy JuAn Wang, Chih-Cheng Hung, "Seventh International Conference on Information Technology," *Path Planning for Virtual Human Motion Using Improved A* Algorithm*, p. 1155, 2010.
- [13] Andy JuAn Wang, Chih-Cheng Hung, Junfeng Yao, Chao Lin , Xiaobiao Xie, "Seventh International Conference on Information Technology," *Path Planning for Virtual Human Motion Using Improved A* Algorithm*, p. 1157, 2010.
- [14] "WIKIPEDIA," 03 05 2022. [Online]. Available: https://en.wikipedia.org/wiki/Simple_linear_regression. [Accessed 10 05 2022].
- [15] S. W. Solutions, "A7600 Series_AT Command Manual".
- [16] "Movable Type Script," [Online]. Available: <https://www.movable-type.co.uk/scripts/latlong.html>. [Accessed 04 03 2022].
- [17] Md. Rashedul Islam and Jong-Myon Kim, *An Effective Approach to Improving Low-Cost GPS Positioning Accuracy in Real-Time Navigation*, vol. 2014, p. 3, 2014.
- [18] [Online]. Available: <http://iottuonglai.com/giao-thuc-du-lieu-iot.html>.

[Accessed 18 5 2022].

[19] *NEO-7 Data Sheet. [Performance]*. U - Blox, 2014.

[20] Daviteg, [Online]. Available: <https://www.daviteq.com/blog/vi/cac-giao-thuc-iot-pho-bien-nhat/>.