

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

KHOA ĐIỆN – ĐIỆN TỬ

NGÀNH HỆ THỐNG NHÚNG VÀ IOT



**HCMUTE**

**MÔN HỌC: THỰC TẬP GIAO THỨC VÀ KIẾN TRÚC IOT**

**GVHD : ThS. Trương Quang Phúc**

**SVTH : Nguyễn Đặng Mai Thy 20139039**

**Trần Minh Đô 20142623**

Thành phố Hồ Chí Minh, 24 Tháng 12 năm 2022

## LỜI CAM ĐOAN

Chúng tôi xin cam kết đồ án cuối kì này là đề tài do chính chúng tôi nghiên cứu và thực hiện. Chúng tôi chỉ dựa trên mà không sao chép từ bất kì tài liệu hay công trình nào đã được thực hiện trước đó. Nếu có bất kỳ vi phạm nào, chúng tôi xin hoàn toàn chịu trách nhiệm.

Người thực hiện

**Nguyễn Đặng Mai Thy**

**Trần Minh Đô**

# Mục lục

<b>LỜI CẢM ƠN .....</b>	6
<b>PHẦN I: TỔNG QUAN .....</b>	7
1.1 Tình trạng thực tế: .....	7
1.2 Đặt vấn đề: .....	7
1.3 Mục tiêu đề tài:.....	8
1.4 Tóm tắt chức năng sản phẩm: .....	9
1.5 Giới hạn đề tài: .....	9
1.6 Phạm vi đề tài:.....	10
1.7 Bố cục đề tài:.....	10
1.8 Kết quả dự kiến đạt được: .....	11
<b>PHẦN II: CƠ SỞ LÝ THUYẾT.....</b>	13
2.1 IoT .....	13
2.1.1 IoT là gì?.....	13
2.1.2 Lịch sử hình thành và tương lai của IoT .....	14
2.1.3 Ứng dụng IoT trong thực tế: .....	14
2.2 Công nghệ WIFI.....	14
2.2.1 Giới thiệu về Wifi .....	14
2.2.2 Lịch sử hình thành các chuẩn kết nối WiFi đến nay.....	15
2.2.3 Ứng dụng với cuộc sống .....	16
2.3 WISE-PaaS – Advantech .....	16
2.3.1 Công ty Advantech .....	16
2.3.2 WISE-PaaS là gì?.....	17
2.3.3 Ưu điểm của WISE-PaaS? .....	17
2.3.4 Vai trò của WISE-PaaS với đề tài .....	18
2.4 Ngôn ngữ Python .....	18
2.4.1 Giới thiệu về ngôn ngữ Python .....	18
2.4.2 Ưu điểm của ngôn ngữ Python .....	18
2.4.3 Vai trò của ngôn ngữ Python với đề tài .....	18
2.5 Ngôn ngữ C++.....	19
2.5.1 Giới thiệu về ngôn ngữ C++ .....	19
2.5.2 Ưu điểm của ngôn ngữ C++ .....	19
2.5.3 Vai trò của ngôn ngữ C++ với đề tài .....	19
<b>PHẦN III: CÔNG CỤ SỬ DỤNG .....</b>	20
3.1 Công cụ phân cứng.....	20
3.2 Công cụ phân mềm.....	36

3.2.1 DataHub .....	36
3.2.2 DashBoard .....	37
3.2.3 Notification .....	38
3.2.4 Arduino IDE .....	39
3.2.5 Pycharm .....	39
3.2.6 Blynk IoT .....	40
3.2.7 Spyder .....	41
<b>PHẦN IV: THUẬT TOÁN, KỸ THUẬT SỬ DỤNG.....</b>	<b>42</b>
4.1 Semantic Segmentation .....	42
4.2 U-net.....	43
4.3 Obstacle Detection .....	44
4.4 Obstacle Avoiding.....	45
4.5 Planning Algorithms for Outdoor Environment .....	45
4.6 Dijkstra Algorithms.....	46
4.7 Building Maps .....	46
<b>PHẦN V: GÓI DỮ LIỆU .....</b>	<b>47</b>
5.1 Client .....	47
5.2 Robot .....	47
5.3 Host .....	48
<b>PHẦN VI: THIẾT KẾ VÀ MÔ HÌNH HOẠT ĐỘNG .....</b>	<b>49</b>
6.1 Hệ thống .....	49
6.1.1 Sơ đồ hoạt động chung của hệ thống .....	49
6.1.2 Sơ đồ chi tiết: .....	50
6.1.3 Hệ thống Pipeline: .....	51
6.2 Xe tự hành vận chuyển.....	52
6.2.1 Tiêu chí thiết kế: .....	52
6.2.2 Cấu trúc phần cứng:.....	53
6.2.3 Thiết kế phần cứng: .....	56
6.2.4 Mô tả sản phẩm.....	57
6.3 WISE-PaaS .....	58
<b>PHẦN VII: KẾT QUẢ VÀ PHÂN TÍCH.....</b>	<b>61</b>
7.1 Obstacle avoiding Robot.....	61
7.2 Semantic Segmentation of Aerial imagery using U-net .....	65
7.3 Dijkstra Algorithms.....	72
7.4 Obstacle Detection .....	73
7.5 DataHub .....	74
7.6 DashBoard.....	75

7.7 Notification .....	75
<b>PHẦN VIII: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN SẢN PHẨM TRONG TƯƠNG LAI .....</b>	<b>76</b>
8.1 Kết luận .....	76
8.2 Định hướng phát triển sản phẩm trong tương lai .....	76
<b>PHẦN IX: NHẬN XÉT CỦA GIÁO VIÊN BỘ MÔN VÀ TIẾN HÀNH VĂN ĐÁP VỀ ĐỀ TÀI .....</b>	<b>77</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>78</b>

## LỜI CẢM ƠN

Lời đầu tiên chúng em xin cảm ơn Thầy Trương Quang Phúc đã đồng hành và hỗ trợ chúng em trong suốt học kì vừa qua, mặc dù chúng em chỉ mới gặp thầy trong một vài tháng nhưng chúng em đã cảm nhận được sự tận tình của thầy trong việc chỉ bảo, định hướng và hỗ trợ cho chúng em cả về vật chất và tinh thần.

Đề tài của chúng em được hoàn thành trong 8 tuần, nó không phải là một thời gian dài nhưng đối với chúng em thì nó cũng khá gấp rút để hoàn thành đề tài vì khả năng và sự hiểu biết còn hạn hẹp. Chắc chắn đề tài chúng em sẽ không được như mong đợi của em và thầy nhưng đó là sự nỗ lực của chúng em trong quá trình vừa qua.

Chúng em mong rằng sẽ nhận được sự góp ý, bổ sung để đề tài của chúng em được hoàn thiện nhất có thể và rút ra những kinh nghiệm cho những môn học tiếp theo.

Một lần nữa chúng em xin cảm ơn thầy. Chúc thầy thật nhiều sức khỏe và luôn thành công trong công cuộc giảng dạy của mình.

# PHẦN I: TỔNG QUAN

## 1.1 Tình trạng thực tế:

Lĩnh vực robot tự hành đang phát triển nhanh chóng trên toàn thế giới, được ứng dụng đa dạng và nhận được sự quan tâm của các cộng đồng ô tô, xe tải, vận tải công cộng, công nghiệp và quân đội. Hệ thống robot tự hành đã và đang được cải tiến đáng kể về an toàn và hiệu quả vận hành. Do sự tăng trưởng của thương mại điện tử, việc phát triển các phương pháp giao hàng nhanh hơn, tiết kiệm hơn và bền vững hơn trở nên quan trọng hơn.

Nhiều thách thức như giảm khả năng vận chuyển, thiếu nhân lực lái xe, sản phẩm bị hư hại và bị đánh cắp, thất bại trong việc giao hàng, tăng tắc đường giao thông, v.v. có thể được giải quyết bằng robot tự hành.

Một robot tự hành được thiết kế và kỹ thuật để xử lý môi trường của nó một cách độc lập và hoạt động trong thời gian dài mà không cần can thiệp của con người. Nó không chỉ phải thực hiện nhiệm vụ giao hàng đúng cách, mà còn phải xem xét các tình huống khác nhau xảy ra xung quanh nó và hành động phù hợp. Robot phải đưa ra quyết định nhanh chóng ngay cả trong điều kiện bất lợi, xem xét an toàn cho người trên tuyến đường xung quanh nó.

Mục đích của robot tự hành là làm việc cùng với con người và cố gắng làm cho cuộc sống của con người dễ dàng hơn. Hiện nay, nhiều robot đang được sử dụng trong các ngành công nghiệp, gia đình, các ứng dụng quân sự, quản lý thiên tai, v.v., trên toàn thế giới. Sự phát triển của lĩnh vực robot đã làm cuộc sống dễ dàng hơn cho con người trong nhiều khía cạnh và nó cung cấp cho con người một phương án an toàn và hiệu quả hơn để thực hiện các nhiệm vụ khó khăn hoặc tốn thời gian.

Bên cạnh đó hiện nay nhiều doanh nghiệp bán hàng đang tìm các giải pháp mở rộng quy mô giao hàng chặng cuối (đến tay khách hàng), Tuy nhiên, chi phí và nguồn lao động thường là những thách thức mà họ phải đối mặt.

## 1.2 Đặt vấn đề:

Nhu cầu vận chuyển trong khuôn viên trường học, bệnh viện, văn phòng tăng cao. Trong khi đó việc đi lại từ các tòa với nhau để vận chuyển các giấy tờ, thuốc thang, vật dụng, vật phẩm cần thiết tốn khá nhiều thời gian và sức lực.

Từ những điều đó nhóm muốn xây dựng một dịch vụ giao hàng an toàn, hiệu quả, tiết kiệm thông qua giải pháp sử dụng robot để thực hiện giao hàng tự động, sử dụng thuật toán tìm đường ngắn nhất để tối ưu hóa việc vận chuyển hàng hóa.

Khi tạo ra một dịch vụ giao hàng như trên, có một số vấn đề cần được giải quyết như sau:

- Kỹ thuật di chuyển: Xác định các phương pháp, kỹ thuật để robot có thể di chuyển đến đúng địa điểm giao hàng với độ chính xác cao.
- Kỹ thuật giao hàng: Xác định các phương pháp, kỹ thuật mà robot có thể giao hàng một cách an toàn và chính xác, và cách đảm bảo rằng hàng hóa được giao trong tình trạng tốt nhất.
- Tích hợp với hệ thống quản lý: Chúng ta cần tích hợp robot với hệ thống quản lý để có thể gửi và nhận thông tin về giao hàng một cách dễ dàng.
- Tương tác với người dùng: Cần xác định phương pháp mà robot có thể tương tác với người nhận hàng và giải quyết những vấn đề xảy ra trong quá trình giao hàng.
- An toàn và bảo mật: Đảm bảo rằng robot hoạt động một cách an toàn và dữ liệu của khách hàng được bảo mật.
- Đánh giá và phát triển: Cần phải đánh giá hiệu suất của robot giao hàng và cải tiến hàng ngày để đảm bảo rằng robot giao hàng hoạt động một cách hiệu quả và đáp ứng được nhu cầu của người dùng.
- Chi phí: Xác định chi phí cho việc sản xuất và triển khai robot giao hàng, bao gồm chi phí cho công nghệ, nhân lực và các chi phí khác để tối ưu hóa lợi nhuận.

### 1.3 Mục tiêu đề tài:

Dựa vào vấn đề đã được nêu ra, nhóm quyết định chọn thực hiện đề tài với những mục tiêu cụ thể như sau:

- Tăng tốc độ giao hàng: Một trong những mục tiêu chính của robot giao hàng là tăng tốc độ giao hàng, giúp giao hàng đến tay người nhận nhanh hơn so với giao hàng truyền thống.
- Giảm chi phí: Giảm chi phí cho dịch vụ giao hàng bằng cách thay thế nhân viên bằng robots.
- Hoạt động chính xác: Robot sẽ có khả năng theo dõi và giữ gìn chính xác các thông tin giao hàng, giúp tránh các lỗi trong quá trình giao hàng.
- Mở rộng khả năng hoạt động: Mở rộng hoạt động của dịch vụ giao hàng bằng cách cho phép robot hoạt động liên tục 24/7.

- Giảm tải lượng công việc cho các nhân viên: Mục tiêu của robot giao hàng cũng bao gồm giảm tải cho nhân viên bằng cách giao hàng cho họ, cho phép họ tập trung vào các nhiệm vụ khác cần sự tập trung.
- Đạt được những mục tiêu này của robot giao hàng sẽ đảm bảo dịch vụ giao hàng hoạt động một cách hiệu quả và tốt hơn, giúp tối ưu hóa quá trình giao hàng và tăng độ hài lòng của khách hàng.

#### **1.4 Tóm tắt chức năng sản phẩm:**

- Vận chuyển hàng hóa trong khuôn viên trường học
- Kết nối giữa robot và người dùng (người gửi và người nhận) thông qua APP/Website
- Cập nhật dữ liệu liên tục về quá trình giao hàng của robot, giúp người dùng dễ dàng theo dõi và kiểm soát quá trình giao hàng.
- Kết hợp với mô hình thang máy thông minh giúp robot có thể vận chuyển đến những tầng trong tòa nhà và giao hàng đến tận nơi.
- Có khả năng tương tác với người dùng bằng cách hiển thị thông tin trên màn LCD.
- Áp dụng xử lý ảnh để xử lý các tình huống trong quá trình giao hàng.
- Đọc hiểu các bản đồ, có khả năng tự vẽ lại bản đồ đường đi.
- Tìm đường đi ngắn nhất.

#### **1.5 Giới hạn đề tài:**

Bài toán này được lên ý tưởng nhằm giải quyết một số vấn đề nhất định, đó là hạn chế đầu tiên của đề tài. Bên cạnh đó còn nhiều giới hạn như:

- Đề tài vẫn đang sử dụng ứng dụng hỗ trợ thứ ba là Blynk, về bản chất, ta có thể hoàn toàn xây dựng một dự án thuần và không cần sử dụng đến Blynk và áp dụng các kiến thức về API . Đây là hạn chế tiếp theo của đề tài.
- Các linh kiện được sử dụng trong đề tài đã được kiểm thử nhiều lần và sẽ có những sai sót trong quá trình sử dụng tiếp theo.
- Môi trường ngoại trời rất khó dự đoán do các chướng ngại vật động như người đi bộ, động vật, xe cộ,....
- Module GPS có sai số rất lớn so với các khoảng cách nhỏ.
- Cảm biến IR và YDLidar dễ bị nhiễu khi chạy outdoor

- Đề tài còn dựa trên lý thuyết khá nhiều nên để có thể triển khai thành một sản phẩm cụ thể cần phải dành nhiều thời gian hơn để nghiên cứu và bổ sung cũng như tìm ra các giải pháp tối ưu triệt để các vấn đề của sản phẩm.

## **1.6 Phạm vi đề tài:**

Phạm vi hoạt động nhắm đến chủ yếu là tại các khuôn viên của các cơ sở như Trường học, Bệnh viện, Cơ quan,..

Đối tượng nhắm đến là các giáo viên, học sinh, y bác sĩ, bệnh nhân, nhân viên cũng như các chủ nhà hàng, chủ cửa hàng đang hoạt động tại cái khu vực này

## **1.7 Bố cục đề tài:**

### **Chương 1: Tổng quan đề tài**

Nêu những tổng quan về đề tài để giáo viên có cái nhìn chung nhất về dự án trước khi đến với chi tiết. Bên cạnh đó nêu ra mục tiêu, giới hạn của đề tài để hiểu sâu hơn về dự án

### **Chương 2: Cơ sở lý thuyết**

Với mỗi bài toán được đặt ra đều dựa trên những kiến thức nền tảng. Chương 2 sẽ nêu ra những lý thuyết cơ sở để hoàn thành được đề tài này. Chương này sẽ nêu ra những kiến thức về IoT , về giao thức, về realtime database .

### **Chương 3: Công cụ sử dụng**

Nêu ra những công cụ được sử dụng cho đề tài. Chương này chia ra 2 phần gồm công cụ vật lý và công cụ phần mềm. Với công cụ vật lý, trình bày các linh kiện, phần cứng ( về mặt vật lý) đã được sử dụng . Tương tự, công cụ phần mềm trình bày các phần mềm đã được sử dụng để lập trình, lưu trữ và điều khiển các thiết bị vật lý đã được trình bày trước đó.

### **Chương 4: Thuật toán, kỹ thuật sử dụng**

Nêu ra những thuật toán, kỹ thuật được sử dụng cho đề tài. Chương này chia ra 2 phần gồm thuật toán và kỹ thuật. Cả 2 đều được cung cấp các phần lý thuyết cơ bản, hình ảnh minh họa của thuật toán và kỹ thuật đó, cũng như các ứng dụng của nó cho đề tài này

## **Chương 5: Gói dữ liệu**

Nêu ra những gói dữ liệu được sử dụng trong đề tài. Gồm 3 phần lớn bao gồm Client, Robot, Host. Cung cấp các dữ liệu về thông tin của các bên, chương này sẽ giải thích kỹ hơn về các nội dung, mục đích của những gói dữ liệu này.

## **Chương 6: Thiết kế và mô hình hoạt động**

Dựa vào những kiến thức đề cập ở trên, tiến hành thiết kế hệ thống. Trình bày tổng quan hệ thống, sơ đồ khái hoạt động chung. Bên cạnh đó chia hệ thống thành 2 phần nhỏ để xây dựng gồm: Xe điều khiển đa năng và WISE-PaaS đi kèm. Qua đó trình bày chi tiết về nguyên lý hoạt động, sơ đồ chức năng của 2 phần này.

## **Chương 7: Báo cáo kết quả thực nghiệm khi vận hành và phân tích**

Cung cấp hình ảnh, mã code, mô hình thực tế liên quan và phân tích về kết quả thực nghiệm khi vận hành của đề tài. Nêu ra những khó khăn gặp phải khi thực nghiệm cũng như báo cáo về những phần chưa làm được.

## **Chương 8: Kết luận và định hướng phát triển sản phẩm trong tương lai**

Đưa ra nhận xét về tiến trình, tiến độ hoàn thành đề tài cũng như đánh giá các ưu, nhược điểm của hệ thống. Xem xét độ khả thi của đề tài, đảm bảo rằng đề tài có khả năng hoàn thiện trong tương lai. Qua đó nêu ra các ý tưởng để cải thiện và phát triển đề tài sau này.

## **Chương 9: Lời nhận xét và giáo viên tiến hành vấn đáp chi tiết về đề tài.**

### **1.8 Kết quả dự kiến đạt được:**

- Mô hình xe tự hành
- App/Website thân thiện với người dùng: App hoặc website sẽ được thiết kế để dễ dàng sử dụng, giúp cho người dùng dễ dàng quản lý và theo dõi việc giao hàng của mình.
- Có khả năng đọc và xử lý: Mô hình xe tự hành sẽ có khả năng đọc và xử lý các thông tin liên quan đến giao hàng, giúp tăng tốc độ và chính xác của việc giao hàng.
- Truyền dữ liệu liên tục và đồng bộ: Mô hình sẽ truyền dữ liệu liên tục lên App/Website, giúp cho người dùng theo dõi việc giao hàng của mình một cách dễ dàng.

- Sử dụng Wise-PaaS kết hợp với tính năng google maps để cập nhận dữ liệu trong quá trình giao hàng.
- Kết hợp với mô hình thang máy quét/đọc hiểu mã từ: Mô hình sẽ kết hợp với mô hình thang máy quét/đọc hiểu mã từ để tăng tốc độ và chính xác của việc giao hàng.
- Mô hình xử lý ảnh: Mô hình sẽ sử dụng các kỹ thuật xử lý ảnh để nhận diện và giải quyết các vấn đề liên quan đến việc giao hàng (nhận diện khách hàng, nhận diện địa hình, môi trường), giúp tăng tốc độ và hiệu quả của việc giao hàng.
- Mô hình tối giản, hiện đại, tối ưu hóa chức năng,... thân thiện với người dùng và có những đặc điểm để nhận diện từ xa, mô hình có hộp đựng hàng hóa hiện đại cho nhiều mục đích khác nhau từ vận chuyển tài liệu đến vận chuyển thức ăn.
- Xây dựng các bản đồ chuyên dụng cho mục đích điều hướng tự động.
- Ước lượng trạng thái và định vị của robot
- Vẽ lại bản đồ, hình ảnh xung quanh trên đường vận chuyển của robot
- Sử dụng Wise-PaaS để tính toán thống kê, số lượng, mô phỏng lại bản đồ, đường đi, khuôn viên nơi vận chuyển

## PHẦN II: CƠ SỞ LÝ THUYẾT

STT	Nội dung lý thuyết	Phân loại
1	IoT	Công nghệ
2	Wifi	Công nghệ
3	WISE-PaaS	Ứng dụng
4	Python	Ngôn ngữ lập trình
5	C++	Ngôn ngữ lập trình

### 2.1 IoT.

#### 2.1.1 IoT là gì?

Thuật ngữ IoT hay Internet vạn vật đề cập đến mạng lưới tập hợp các thiết bị thông minh và công nghệ tạo điều kiện thuận lợi cho hoạt động giao tiếp giữa thiết bị và đám mây cũng như giữa các thiết bị với nhau. Nhờ sự ra đời của chip máy tính giá rẻ và công nghệ viễn thông băng thông cao, ngày nay, chúng ta có hàng tỷ thiết bị được kết nối với internet. Điều này nghĩa là các thiết bị hàng ngày như bàn chải đánh răng, máy hút bụi, ô tô và máy móc có thể sử dụng cảm biến để thu thập dữ liệu và phản hồi lại người dùng một cách thông minh.

Internet vạn vật tích hợp “vạn vật” với Internet mỗi ngày. Các kỹ sư máy tính đã và đang thêm các cảm biến và bộ xử lý vào các vật dụng hàng ngày kể từ những năm 90. Tuy nhiên, tiến độ ban đầu rất chậm vì các con chip còn to và cồng kềnh. Loại chip máy tính công suất thấp gọi là thẻ tag RFID, lần đầu tiên được sử dụng để theo dõi các thiết bị đắt đỏ. Khi kích cỡ của các thiết bị điện toán dần nhỏ lại, những con chip này cũng trở nên nhỏ hơn, nhanh hơn và thông minh hơn theo thời gian.

Chi phí tích hợp công suất điện toán vào trong các vật dụng nhỏ bé hiện nay đã giảm đáng kể. Ví dụ: bạn có thể thêm khả năng kết nối với các tính năng của dịch vụ giọng thoại Alexa vào các MCU tích hợp sẵn RAM chưa đến 1 MB, chẳng hạn như cho công tắc đèn. Nguyên cả một ngành công nghiệp đã bắt đầu xuất hiện với trọng tâm xoay quanh việc trang bị các thiết bị IoT khắp mọi ngóc ngách căn nhà, doanh nghiệp và văn phòng của chúng ta.

### **2.1.2 Lịch sử hình thành và tương lai của IoT**

Mặc dù thuật ngữ Internet of things chỉ thật sự bùng nổ và thu hút sự quan tâm của thế giới công nghệ trong những năm gần đây. Tuy nhiên thực tế IoT đã có từ rất lâu đời, khoảng từ nhiều thập kỷ trước đó. Bởi vì phải đợi mãi đến năm 1999, cụm từ thuật ngữ Internet of things này mới được đưa ra thị trường bởi nhà khoa học Kevin Ashton.

IoT phát triển qua nhiều giai đoạn và đang ngày càng lớn mạnh trên thị trường toàn cầu. Theo như phân tích của một số chuyên gia từ IDC thì dự đoán đến năm 2025 sẽ có xấp xỉ 41,6 tỷ thiết bị thông minh được kết nối.

Đồng thời, các mảng dịch vụ tiện ích khác cũng sẽ ưu tiên sử dụng IoT ngày càng nhiều hơn. Đi song song với đó là các thiết bị camera an ninh, công nghệ ô tô, nhà thông minh hay các vấn đề liên quan đến chăm sóc sức khỏe.

### **2.1.3 Ứng dụng IoT trong thực tế:**

- Tự động hóa trong giao thông.
- Quản lý hạ tầng và các vấn đề liên quan đến chất thải.
- Lĩnh vực xây dựng, môi trường, y tế (gồm các thiết bị, giám sát bệnh án, dụng cụ theo dõi sức khỏe,...) hay giao thông.
- Những ứng dụng thông minh khác như tự động hóa các thiết bị trong nhà, văn phòng.
- Ngoài ra còn có mua sắm thông minh, đồng hồ đo thông minh, quản lý các vật dụng hay thiết bị trong nhà, tìm kiếm thông tin qua GPS...

## **2.2 Công nghệ WIFI**

### **2.2.1 Giới thiệu về Wifi**

Wifi là viết tắt của Wireless Fidelity là hệ thống truy cập internet không dây.

Loại sóng vô tuyến này tương tự như sóng điện thoại, truyền hình và radio.

Wifi là một mạng thay thế cho mạng có dây thông thường, thường được sử dụng để kết nối các thiết bị ở chế độ không dây bằng việc sử dụng công nghệ sóng vô tuyến. Dữ liệu được truyền qua sóng vô tuyến cho phép cá thiết bị truyền nhận dữ liệu ở tốc độ cao trong phạm vi của mạng WIFI. Kết nối các máy tính với nhau, với Internet và với mạng có dây.

Vì nhu cầu sử dụng ngày càng cao của con người và chi phí lắp đặt thấp đã kiến cho WIFI ngày càng được phổ biến rộng rãi ở nhiều nơi như công viên, quán cafe, sân bay,...

## 2.2.2 Lịch sử hình thành các chuẩn kết nối WiFi đến nay

### Chuẩn 802.11

Năm 1997, IEEE giới thiệu chuẩn mạng không dây đầu tiên và đặt tên nó là 802.11. Khi đó, tốc độ hỗ trợ tối đa của mạng này chỉ là 2 Mbps với băng tầng 2.4GHz.

### Chuẩn 802.11b

Vào tháng 7/1999, chuẩn 802.11b ra đời và hỗ trợ tốc độ lên 11Mbps. Chuẩn này cũng hoạt động tại băng tần 2.4GHz nên cũng rất dễ bị nhiễu từ các thiết bị điện tử khác.

### Chuẩn 802.11a

Song song với quá trình hình thành chuẩn b, chuẩn 802.11a phát ở tần số cao hơn là 5GHz nhằm tránh bị nhiễu từ các thiết bị khác. Tốc độ xử lý của chuẩn đạt 54Mbps tuy nhiên chuẩn này khó xuyên qua các vách tường và giá cả của nó hơi cao.

### Chuẩn 802.11g

Chuẩn 802.11g có phần hơn so với chuẩn b, tuy nhiên nó cũng hoạt động ở tần số 2.4GHz nên vẫn dễ nhiễu. Chuẩn này có thể xử lý tốc độ lên tới 54 Mbps.

### Chuẩn 802.11n

Ra mắt năm 2009 và là chuẩn phổ biến nhất hiện nay nhờ sự vượt trội hơn so với chuẩn b và g. Chuẩn kết nối 802.11n hỗ trợ tốc độ tối đa lên đến 300Mbps, có thể hoạt động trên cả băng tần 2,4 GHz và 5 GHz.

Chuẩn kết nối này đã và đang dần thay thế chuẩn 802.11g với, phạm vi phát sóng lớn hơn, tốc độ cao hơn và giá hợp lý.

### Chuẩn 802.11ac

Là chuẩn được IEEE giới thiệu vào đầu năm 2013, hoạt động ở băng tần 5 GHz. Chuẩn ac có thể mang đến cho người dùng trải nghiệm tốc độ cao nhất lên đến 1730 Mbps.

Do vấn đề giá thành cao nên các thiết bị phát tín hiệu cho chuẩn này chưa phổ biến dẫn đến các thiết bị này sẽ bị hạn chế sự tối ưu do thiết bị phát.

## **Chuẩn 802.11ad**

Được giới thiệu năm 2014, chuẩn wifi 802.11ad được hỗ trợ băng thông lên đến 70 Gbps và hoạt động ở dải tần 60GHz.

Nhược điểm của chuẩn này là sóng tín hiệu khó có thể xuyên qua các bức tường, đồng nghĩa với việc chỉ cần Router khuất khỏi tầm mắt, thiết bị sẽ không còn kết nối tới Wifi được nữa.

## **Chuẩn 802.11ax**

Wi-Fi 6 là bản cập nhật mới nhất cho chuẩn mạng không dây. Wi-Fi 6 dựa trên tiêu chuẩn IEEE 802.11ax, với tốc độ nhanh hơn, dung lượng lớn hơn và hiệu suất năng lượng được cải thiện tốt hơn so với các kết nối không dây trước đây. Tên gọi mới Wifi 6 này sẽ chính thức được áp dụng từ năm 2019.

### **Wifi Hotspot**

#### **2.2.3 Úng dụng với cuộc sống**

- Kết nối các thiết bị điện tử trong nhà và thông qua đó điều khiển chúng từ xa.  
Vd: iTune controlling remove của Apple.
- Điều khiển máy in từ xa, gửi văn bản đến máy in trong cùng một khu vực.
- Dùng mạng wifi để chia sẻ dữ liệu máy tính xung quanh mà không cần qua các thiết bị có dây nào.
- Chuyển các thông báo từ Smartphone đến máy tính.
- Chia sẻ kết nối dữ liệu của Smartphone cho máy tính: Dùng Smartphone có kết nối mạng để phát wifi cho máy tính.
- Với những Tivi đời cao có kết nối Wifi bạn có thể tha hồ xem phim mà không cần đến đầu thu. Không chỉ ti vi mà bạn có thể truyền đến các loa trong nhà thông qua Smartphone.

## **2.3 WISE-PaaS – Advantech**

### **2.3.1 Công ty Advantech**

Advantech là công ty hàng đầu thế giới trong việc cung cấp các sản phẩm và giải pháp nhúng và tự động hóa. Công ty được thành lập vào năm 1983 và có trụ sở chính tại Đài Loan. Advantech cung cấp nhiều loại sản phẩm, bao gồm máy tính công nghiệp, máy tính bảng đơn, thiết bị liên lạc công nghiệp, giao diện người-máy (HMI), hệ thống điều khiển công nghiệp và hệ thống giao thông thông minh.

Các sản phẩm và giải pháp của Advantech được sử dụng trong nhiều ngành công nghiệp khác nhau, chẳng hạn như sản xuất, vận chuyển, chăm sóc sức khỏe, bán lẻ và hậu cần. Họ tập trung vào việc kích hoạt các hệ thống thông minh và giúp khách hàng đạt được chuyển đổi kỹ thuật số bằng cách tích hợp công nghệ phần cứng và phần mềm.

### 2.3.2 WISE-PaaS là gì?

WISE-PaaS (Platform-as-a-Service) là một nền tảng ứng dụng Internet of Things (IoT) do Advantech cung cấp. Nó được thiết kế để cung cấp một bộ giải pháp phát triển và cơ sở hạ tầng phần mềm IoT hoàn chỉnh cho các nhà tích hợp hệ thống và nhà phát triển phần mềm để dễ dàng phát triển và triển khai các ứng dụng và dịch vụ IoT.

WISE-PaaS cung cấp nhiều loại dịch vụ phần mềm IoT, bao gồm thu thập và quản lý dữ liệu, phân tích đám mây, quản lý thiết bị và hỗ trợ ứng dụng. Nó giúp các doanh nghiệp nhanh chóng xây dựng và triển khai các ứng dụng và dịch vụ IoT, giảm chi phí phát triển và bảo trì, đồng thời nâng cao hiệu quả hoạt động.

### 2.3.3 Ưu điểm của WISE-PaaS?

- Tích hợp đồng bộ:** Wise-PaaS kết hợp các công cụ và dịch vụ cần thiết để phát triển, triển khai và quản lý các ứng dụng IoT công nghiệp. Nó cung cấp một môi trường tích hợp và đồng bộ để giảm thiểu sự phức tạp trong việc tích hợp các thành phần khác nhau.
- Quản lý tập trung:** Wise-PaaS cung cấp một trung tâm điều khiển tập trung để quản lý các ứng dụng và thiết bị IoT. Người dùng có thể theo dõi và điều khiển từ xa các thiết bị, thu thập dữ liệu, và quản lý cấu hình và cập nhật.
- Phân tích dữ liệu:** Wise-PaaS cung cấp các công cụ và khả năng phân tích dữ liệu mạnh mẽ để trích xuất thông tin giá trị từ dữ liệu IoT.
- Linh hoạt và mở rộng:** Wise-PaaS hỗ trợ nhiều giao thức và tiêu chuẩn kết nối, cho phép tích hợp dễ dàng với các thiết bị và hệ thống khác nhau. Nó cũng hỗ trợ việc mở rộng và tùy chỉnh để đáp ứng các yêu cầu cụ thể của từng doanh nghiệp.
- Bảo mật và tin cậy:** Wise-PaaS đảm bảo tính bảo mật và tin cậy trong việc lưu trữ và truyền tải dữ liệu IoT. Nó áp dụng các biện pháp bảo mật nâng cao và các chứng chỉ bảo mật để bảo vệ dữ liệu quan trọng khỏi các mối đe dọa tiềm tàng.

### **2.3.4 Vai trò của WISE-PaaS với đề tài**

Với những tiện ích WISE-PaaS đem lại, đề tài đã sử dụng 3 tiện ích mạnh mẽ là DataHub, DashBoard và Notification. Giúp dữ liệu được cập nhật theo thời gian thực và chính xác.

WISE-PaaS trong đề tài còn là cầu nối giữa xe điều khiển và khách hàng cũng như máy chủ. Các dữ liệu được khách hàng, máy chủ trả về khi tiến hành truy vấn đều được lấy từ WISE-PaaS.

Qua đó, WISE-PaaS là nền tảng hỗ trợ rất nhiều cho sinh viên nói riêng và IoT ngày càng đang phát triển nói chung.

## **2.4 Ngôn ngữ Python**

### **2.4.1 Giới thiệu về ngôn ngữ Python**

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Python có cú pháp đơn giản và dễ sử dụng, do đó nó trở thành một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người mới bắt đầu.

### **2.4.2 Ưu điểm của ngôn ngữ Python**

Python là một ngôn ngữ đơn giản dễ đọc dễ học. Nó không có cú pháp phức tạp như các ngôn ngữ cao cấp khác như C hoặc C++. Nhờ ít phức tạp hơn, Python cho phép bạn suy nghĩ rõ ràng và tập trung vào việc xây dựng logic.

Nhờ có tính năng đơn giản, nên việc bảo trì các chương trình được tạo bởi Python cũng khá đơn giản và nhanh chóng.

Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.

Python có các thư viện như Cython và Jython, cho phép tích hợp các ngôn ngữ khác như C, C++ và Java để phát triển đa nền tảng. Đây là một trong những đặc quyền chính của Python vì không có ngôn ngữ nào hoàn hảo và đôi khi sự phát triển đòi hỏi các chức năng ngôn ngữ đa dạng.

### **2.4.3 Vai trò của ngôn ngữ Python với đề tài**

Python là ngôn ngữ lập trình chính cho các thuật toán, kỹ thuật trong đề tài. Với việc các thư viện hỗ trợ mạnh mẽ nhiều chức năng. Trong đó có cv2, train\_model giúp nhận diện và xử lý ảnh, video,.....

Là ngôn ngữ chạy được trên cả Spyder lẫn Pycharm

## 2.5 Ngôn ngữ C++

### 2.5.1 Giới thiệu về ngôn ngữ C++

C++ là một ngôn ngữ lập trình phổ biến, phân biệt chữ hoa và chữ thường, và ngôn ngữ lập trình dạng tự do hỗ trợ lập trình hướng đối tượng, thủ tục và generic.

**C++ là ngôn ngữ lập trình bậc trung** vì nó bao gồm sự kết hợp giữa các tính năng ngôn ngữ cấp cao và cấp thấp.

### 2.5.2 Ưu điểm của ngôn ngữ C++

- **Tốc độ:** C++ được biết đến với tốc độ nhanh hơn bất kỳ ngôn ngữ lập trình nào khác. Chạy song song nhiều mã cùng một lúc là một trong những ưu điểm tốt nhất mà C++ đem lại, trực tiếp giúp thực thi chương trình nhanh hơn. Cung cấp hiệu suất tốt nhất ngay cả khi máy chủ đang trong tình trạng quá tải.
- **Dựa trên phần cứng:** Khi phần mềm kết hợp chặt chẽ với phần cứng sẽ phải cần một số hỗ trợ cấp thấp ở cấp phần mềm. Và C++ cung cấp sự hỗ trợ đó vì nó gần với phần cứng hơn so với bất kỳ ngôn ngữ lập trình nào khác.
- **Chức năng:** Các tính năng của C++ như Inheritance, Encapsulation, Abstraction rất hữu ích cho lập trình viên trong quá trình phát triển phần mềm. Những ưu điểm này kết hợp lại sẽ cho ra một sản phẩm hiệu quả và vượt trội. Tính hiệu quả sẽ luôn cao hơn trong C++. C++ còn hỗ trợ nạp chồng toán tử, cho phép các toán tử do người dùng định nghĩa cùng với nạp chồng hàm.
- **Mô hình:** Chương trình C++ sử dụng lập trình đa mô hình (mô hình có nghĩa là phong cách lập trình, mô hình liên quan đến logic, cấu trúc và thủ tục của chương trình). Chương trình C++ sử dụng lập trình đa mô hình nghĩa là nó tuân theo ba mô hình Generic, Imperative, Object-Oriented.

### 2.5.3 Vai trò của ngôn ngữ C++ với đề tài

Ngôn ngữ C++ trong lập trình nhúng bởi tốc độ thời gian chạy nhanh, tối ưu mã hóa. Bên cạnh đó, ứng dụng Arduino IDE sử dụng ngôn ngữ C++ để thiết kế chương trình cho các board mạch.

Vì vậy, chúng em sử dụng C++ làm ngôn ngữ chính cho hầu hết chương trình các board mạch của mình.

## PHẦN III: CÔNG CỤ SỬ DỤNG

### 3.1 Công cụ phần cứng

Giới thiệu các công cụ phần cứng được sử dụng trong đề tài đi kèm với thông số kỹ thuật, ứng dụng của chúng.

Bảng liệt kê các thiết bị:

STT	Tên thiết bị	Phân loại
1	Arduino Mega/ Uno	Board
2	Orbecc Astra Depth Camera	Camera
3	ESP32-CAM	Camera
4	USB UART TTL FT232	Camera
5	SparkFun IMU Breakout MPU9250	Góc trực
6	Neo-M8N GPS Module	GPS
7	YDLIDAR X4	Khoảng cách
8	SHARP IR Sensor	Khoảng cách
9	HC-SR04	Khoảng cách
10	Nvidia Jetson TX1	Module
11	L298N	Motor
12	Geared DC Motors & Wheels	Motor
13	Micro Servos	Servo
14	RTC DS3231	Thời gian thực

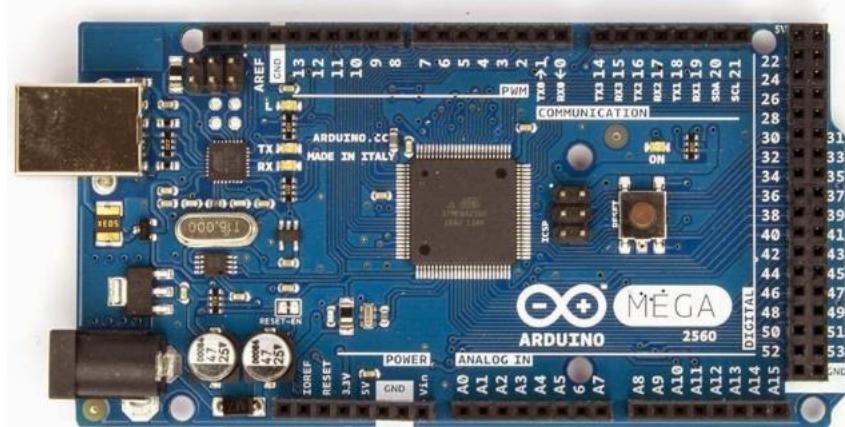
## ❖ Arduino Mega/ Uno

### • Đặc điểm

Arduino Mega là một bảng vi điều khiển dựa trên bộ xử lý ATmega2560. Đây là phiên bản nâng cao của bo mạch Arduino Uno gốc với nhiều bộ nhớ hơn, nhiều chân cảm biến và nhiều tính năng hơn.

Bo mạch Mega có 54 chân đầu vào/đầu ra kỹ thuật số, 16 đầu vào tương tự và 4 UART (cổng nối tiếp phần cứng). Nó cũng có kết nối USB, giắc cắm nguồn, tiêu đề ICSP và nút đặt lại.

Bo mạch phù hợp với nhiều loại ứng dụng yêu cầu nhiều chân cảm biến đầu vào/đầu ra, chẳng hạn như chế tạo người máy, nhà thông minh và thu thập dữ liệu. Nó có thể được lập trình bằng phần mềm Arduino và hỗ trợ nhiều ngôn ngữ lập trình khác nhau, bao gồm C và C++.



### • Thông số kỹ thuật

- MCU (Microcontroller Unit): ATmega2560
- Tần số hoạt động: 16 MHz
- Số lượng chân Digital I/O: 54 (trong đó 15 chân hỗ trợ PWM)
- Số lượng chân Analog Input: 16
- Dung lượng bộ nhớ Flash: 256 KB  
(trong đó 8 KB được dành riêng cho bootloader)
- Dung lượng SRAM: 8 KB
- Dung lượng EEPROM: 4 KB
- Điện áp hoạt động: 5V
- Dòng ra tối đa trên mỗi chân Digital I/O: 20 mA
- Dòng ra tối đa trên chân 3.3V: 50 mA
- Kích thước: 101.52 mm x 53.3 mm

## ❖ Orbecc Astra Depth Camera

### • Đặc điểm

Orbecc Astra Depth Camera là một máy ảnh hiện đại được thiết kế để chụp ảnh chất lượng cao với thông tin về độ sâu. Nó sử dụng công nghệ hồng ngoại tiên tiến để nắm bắt thông tin độ sâu, cho phép nó tạo ra hình ảnh và video 3D.

Máy ảnh này có khả năng theo dõi lên đến 30 khung hình mỗi giây, làm cho nó trở thành một công cụ lý tưởng để chụp các vật thể và người chuyển động nhanh. Nó cũng được thiết kế để sử dụng trong điều kiện ánh sáng yếu, khiến nó trở nên hoàn hảo cho các hoạt động ngoài trời và chụp ảnh ban đêm.

Máy ảnh có thể được sử dụng trong nhiều ứng dụng khác nhau, bao gồm thực tế ảo, người máy và quét 3D. Nhìn chung, Orbecc Astra Depth Camera là một công cụ mạnh mẽ có thể giúp người dùng ghi lại những hình ảnh và video tuyệt đẹp một cách dễ dàng.



### • Thông số kỹ thuật

#### ○ Cảm biến:

Cảm biến chiều sâu (Depth Sensor): IR (infrared) structured light

Độ phân giải chiều sâu: 640 x 480 pixels

Tốc độ khung hình (Frame rate): Lên đến 30 fps

#### ○ Camera màu (RGB Camera):

Độ phân giải camera màu: 640 x 480 pixels

Tốc độ khung hình (Frame rate): Lên đến 30 fps

#### ○ Góc nhìn (Field of View):

Góc nhìn chiều sâu (Depth Field of View): 60° x 49.5° (+/- 3°)

Góc nhìn camera màu (RGB Field of View): 60° x 49.5° (+/- 3°)

#### ○ Hỗ trợ hệ điều hành:

Windows: Windows 7, 8, 10 (32-bit và 64-bit)

Linux: Ubuntu 16.04 (64-bit)

## ❖ ESP32-CAM

### • Đặc điểm

ESP32-CAM có một camera kích thước nhỏ, rất cạnh tranh trong ngành, giống như mô-đun chính, mô-đun này có thể được xử lý công việc độc lập, module có kích thước nhỏ gọn chỉ 40 x 27 x 12 mm, dòng nghỉ chỉ 6mA.

ESP-32CAM có thể được sử dụng rộng rãi trong các ứng dụng IoT khác nhau, thích hợp cho thiết bị thông minh gia đình, điều khiển không dây công nghiệp, giám sát không dây kiểm soát, nhận dạng không dây QR, tín hiệu hệ thống định vị không dây...Nó là một giải pháp lý tưởng cho các ứng dụng IoT

Mạch thu phát Wifi BLE ESP32 này là mạch chính hãng AI – Thinker có chất lượng độ ổn định và độ bền rất cao, sử dụng camera OV2640 chất lượng cao hình ảnh sắc nét, không nhiễu sọc, không xảy ra tình trạng treo khi hoạt động do sử dụng ic cấp nguồn chất lượng cao.

Mạch thu phát Wifi BLE ESP32-CAM Ai-Thinker này có thể sử dụng Arduino IDE để biên dịch và viết code, được hỗ trợ mạnh mẽ từ cộng đồng.

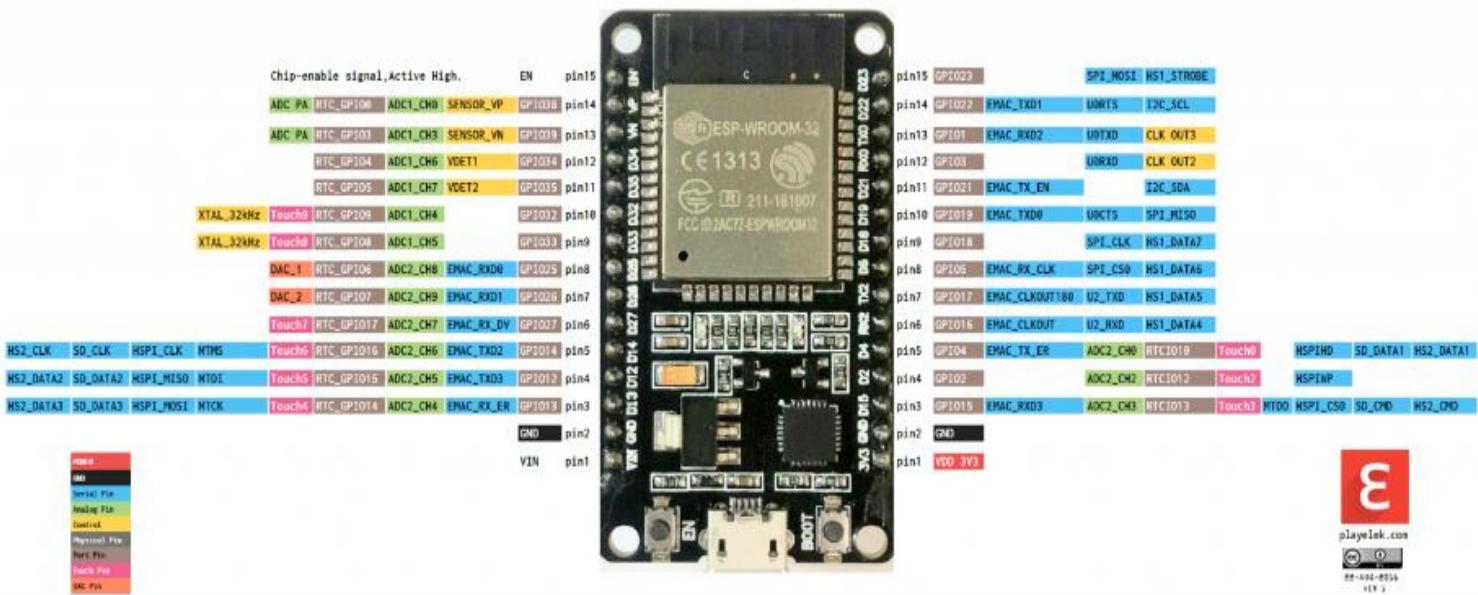


### • Thông số kỹ thuật

- IC chính: ESP32-S (AI-Thinker)
- Mô-đun Wi-Fi BT SoC 802.11 b/g/n/e/i
- CPU 32-bit công suất thấp, cũng có thể phục vụ bộ xử lý ứng dụng
- Tốc độ đồng hồ lên đến 160MHz, sức mạnh tính toán lên đến 600 DMIPS
- Tích hợp 520 KB SRAM, 4MB SRAM bên ngoài
- Dải tần số: 1421 ~ 2484 Mhz
- Bluetooth: 4.2 BR/EDR BLE
- Hỗ trợ UART / SPI / I2C / PWM / ADC / DAC
- Hỗ trợ máy ảnh OV2640 và OV7670, đèn flash tích hợp

- Hỗ trợ tải lên WiFi hình ảnh
  - Hỗ trợ thẻ TF
  - Hỗ trợ nhiều chế độ ngủ
  - Nhúng Lwip và FreeRTOS
  - Hỗ trợ chế độ hoạt động STA / AP / STA + AP
  - Hỗ trợ cấu hình thông minh / công nghệ AirKiss
  - Hỗ trợ nâng cấp cục bộ và từ xa cho cổng nối tiếp (FOTA)

#### • Sơ đồ chân



#### ❖ Mạch chuyển USB UART TTL FT232

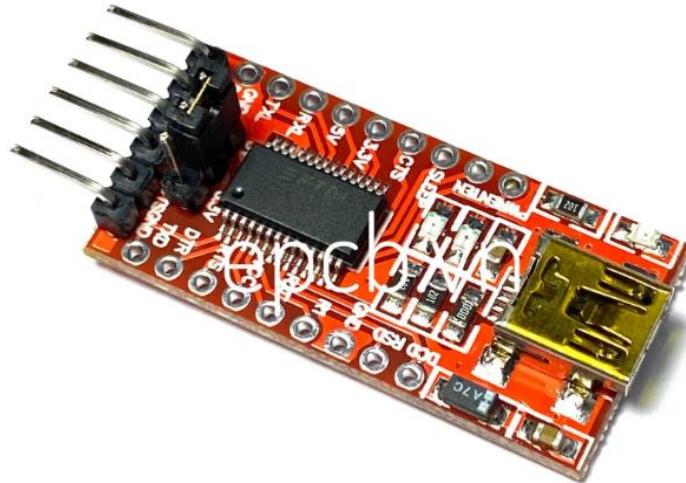
### • Đặc điểm

Mạch chuyển USB UART TTL FT232RL sử dụng IC FT232RL từ chính hãng FTDI, mạch được thiết kế nhỏ gọn nhưng vẫn ra chân đầy đủ, rất dễ sử dụng với mọi hệ điều hành Windows, Mac, Linux

- Chip có sẵn ổn áp và dao động tích hợp bên trong, hoạt động rất ổn định so với các dòng chip USB to serial khác
  - Mạch có thể hoạt động ở 2 chế độ 5v hoặc 3v3, bằng cách thiết lập trên jumper trên mạch
  - Chân cắm ra gồm 2 loại theo chuẩn FTDI (phù hợp với Arduino) và chuẩn UART thường, được ký hiệu rõ ràng trên mạch. Đầu vào sử dụng loại USB B mini.
  - Ngoài ra, trên mạch có sẵn 2 led cho tín hiệu TX và RX, giúp theo dõi trực tiếp trạng thái tín hiệu.

USB to Serial – FTDI có kích thước nhỏ gọn dễ dàng tích hợp vào ứng dụng:

- Làm mạch nạp cho các bản Arduino không tích hợp mạch nạp onboard như: Chipi PRO-LITE, Arduino Pro, Lilypad...
- Làm trung gian giao tiếp bo mạch với máy tính. rất hữu ích khi cần truyền dữ liệu từ bo mạch lên máy tính để kiểm tra, phân tích.
- Làm mạch nạp cho một số dòng vi điều khiển ARM, AVR, 89, PIC,... có hỗ trợ nạp bằng UART.



- **Thông số kỹ thuật**

- IC chính: FT232RL chính hãng FTDI
- Nguồn cấp: 5VDC từ cổng USB (cổng mini USB)
- Có ngõ ra nguồn có thể điều chỉnh 3V3 hoặc 5VDC
- Chuyển giao tiếp từ USB sang UART TTL
- Drive hỗ trợ Windows Mac, Linux
- Có cầu chì tự phục hồi: 500mA
- Tốc độ Baudrate: tùy chỉnh
- Kích thước PCB: 36 x 18.5mm
- Trọng lượng: 3g

- **Sơ đồ chân**

- DTR: Data terminal ready control output / Handshake signal. (có thể reset arduino khi nạp chương trình)
- RXD: Receive asynchronous data Input – nhận tín hiệu
- TXD: Transmit asynchronous data output – truyền tín hiệu
- VCC: Chân nguồn cấp, có thể chọn 5V hoặc 3.3VDC thông qua Jumper
- CTS: Clear to send control input / handshake signal (không sử dụng)

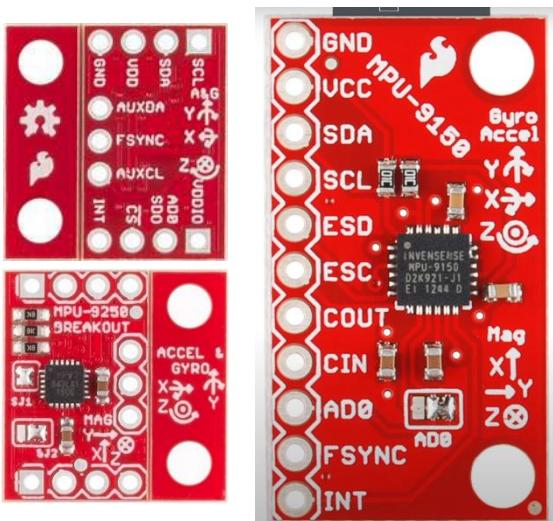
## ❖ SparkFun IMU Breakout MPU9250

### • Đặc điểm

SparkFun MPU-9250 IMU Breakout có cảm biến MEMS 9 trục mới nhất của InvenSense. Mỗi đột phá 9DoF này đều có MPU-9250 với Hệ thống trong Gói (SiP) kết hợp hai chip: MPU-6500, chứa con quay hồi chuyển 3 trục cũng như gia tốc ké 3 trục và AK8963, có tính năng từ ké 3 trục.

Để đạt được kích thước nhỏ hơn, MPU-9250 Breakout có các chân PTH được quấn quanh viền của PCB thành ba hàng ba hoặc bốn hàng. Hàng trên cùng (J1) là tất cả những gì người ta cần để tận dụng tối đa chức năng của IMU.

MPU-9250 sử dụng Bộ chuyển đổi Tương tự sang Kỹ thuật số (ADC) 16-bit để số hóa tất cả chín trục, làm cho nó trở thành bảng 9 Bậc Tự do rất ổn định



## ❖ **Module Định Vị Toàn Cầu GPS - GY-NEO6MV2**

### • **Đặc điểm:**

Bộ thu GPS chất lượng cao với giá thành thấp này được phát triển trên module u-Blox Neo-6M GPS nổi tiếng và cao cấp. Nó đi kèm với pin nhỏ để khởi động nóng (hot startup) và có EEPROM tích hợp.

Để cung cấp khả năng thu tín hiệu GPS tốt hơn, module có một ăng ten bên ngoài kết nối với bo mạch chủ thông qua đầu nối U.FL

Để sử dụng module GY-NEO6MV2, bạn cần kết nối nó với vi điều khiển hoặc hệ thống nhúng thông qua giao tiếp UART và lập trình để đọc và xử lý dữ liệu GPS được truyền từ module.

Lưu ý rằng để module GPS hoạt động chính xác, bạn cần đặt anten ở nơi có tầm nhìn rõ ràng và không bị che khuất bởi các vật cản như tòa nhà cao, cây cối dày, hay nằm trong các khu vực có tòa nhà chọc trời.



### • **Thông số kỹ thuật**

- Mạch định vị GPS GY-NEO 6M, nguồn cung cấp 3V-5V
- Mẫu: GY-GPS6MV2
- Mô-đun với ăng-ten băng sứ, tín hiệu mạnh
- EEPROM power-down lưu dữ liệu tham số cấu hình
- Với pin dự phòng dữ liệu
- Chỉ báo tín hiệu LED
- Kích thước anten 12\*12mm
- Kích thước module 23mm\*30mm
- Lắp đặt khẩu độ 3mm
- Tốc độ baud mặc định: 9600
- Tương thích với nhiều module điều khiển chuyến bay, cung cấp kiểm tra máy tính GPS

## ❖ YDLIDAR X4

### • Đặc điểm

YDLIDAR X4 là một cảm biến đo khoảng cách bằng laser nhỏ gọn, hiệu suất cao, thường được sử dụng trong các ứng dụng về rô-bốt, lập bản đồ và khảo sát.

Nó được thiết kế để cung cấp các phép đo khoảng cách chính xác và đáng tin cậy trên nhiều môi trường trong nhà và ngoài trời.

Cảm biến sử dụng cơ chế quét 360 độ cho phép nó thu được một cái nhìn toàn diện về môi trường xung quanh, khiến nó trở thành lựa chọn tuyệt vời để điều hướng tự động và phát hiện chướng ngại vật.

YDLIDAR X4 cũng dễ dàng tích hợp với các hệ thống phần cứng và phần mềm khác nhờ các giao thức truyền thông linh hoạt và quy trình cài đặt đơn giản.

Nhìn chung, YDLIDAR X4 là một công cụ mạnh mẽ và linh hoạt có thể giúp các cá nhân và tổ chức hoàn thành mục tiêu cảm biến và lập bản đồ của họ một cách dễ dàng.



### • Thông số kỹ thuật

- Số lượng lần quét: 1
- Góc quét: 360°
- Độ phân giải góc quét: 0.36°
- Tốc độ quét: 6000 lần quét mỗi giây
- Phạm vi hoạt động: 0.1 - 12 mét
- Sai số khoảng cách: ±2 cm
- Điện áp hoạt động: 4.8 - 5.2V DC
- Công suất tiêu thụ: 2W
- Giao tiếp: USB (dùng để cung cấp nguồn và truyền dữ liệu)
- Kích thước: Đường kính 70 mm, cao 63 mm
- Trọng lượng: 150g

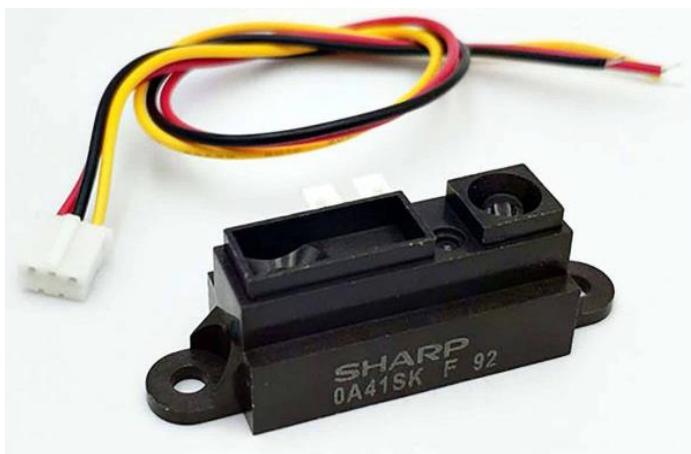
## ❖ SHARP IR Sensor

### • Đặc điểm

Các cảm biến khoảng cách hồng ngoại là một lựa chọn phổ biến cho nhiều dự án yêu cầu đo khoảng cách chính xác. Cảm biến hồng ngoại sharp này kinh tế hơn so với máy đo khoảng cách sonar, nhưng nó cung cấp hiệu suất tốt hơn nhiều so với các lựa chọn thay thế IR khác.

Giao diện với hầu hết các bộ vi điều khiển rất đơn giản: đầu ra tương tự đơn có thể được kết nối với bộ chuyển đổi tương tự sang số để thực hiện các phép đo khoảng cách hoặc đầu ra có thể được kết nối với bộ so sánh để phát hiện ngưỡng. Phạm vi phát hiện của phiên bản này là khoảng 4 cm đến 30 cm (1,5 " đến 12 ").

Cảm biến hồng ngoại GP2Y0A41 sử dụng đầu nối JST PH 3 chân hoạt động với cáp JST PH 3 chân của chúng tôi cho các cảm biến khoảng cách Sharp (không bao gồm), như thể hiện trong hình trên bên phải. Các dây cáp này có các đầu nối JST 3 chân ở một đầu và có sẵn với các chân nam được uốn trước, các chân cái được uốn trước và với các dây bị đứt ở đầu kia.



### • Thông số kỹ thuật

- Điện áp hoạt động: 4,5 V đến 5,5 V
- Mức tiêu thụ điện tại trung bình: 12 mA (lưu ý: cảm biến này rút ra dòng điện lớn, ngắn và nhả sản xuất khuyên nên đặt một tụ 10 10FF hoặc lớn hơn trên nguồn và tiếp đất gần cảm biến để ổn định đường dây cung cấp điện)
- Phạm vi đo khoảng cách: 4cm đến 30cm (1,5 " đến 12 ")
- Loại đầu ra: điện áp tương tự
- Chênh lệch điện áp đầu ra trên phạm vi khoảng cách: 2,3 V (diễn hình)
- Thời gian cập nhật:  $16,5 \pm 4$  ms
- Kích thước:  $44,5 \text{ mm} \times 18,9 \text{ mm} \times 13,5 \text{ mm}$  ( $1,75 \times 0,75 " \times 0,53$ )
- Trọng lượng: 3,5 g (0,12 oz)

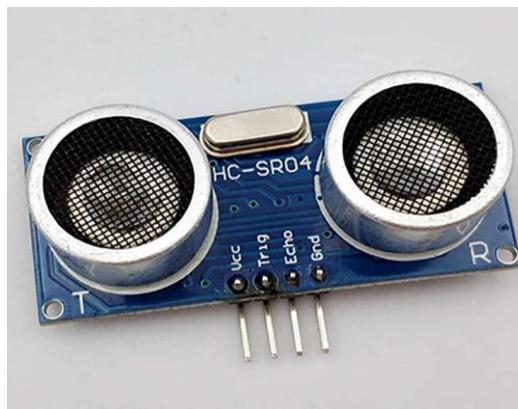
## ❖ HC-SR04

### • Đặc điểm

Cảm biến siêu âm HC-SR04 (Ultrasonic Sensor) được sử dụng rất phổ biến để xác định khoảng cách vì RẺ và CHÍNH XÁC. Cảm biến HC-SR04 sử dụng sóng siêu âm và có thể đo khoảng cách trong khoảng từ 2 -> 300cm, với độ chính xác gần như chỉ phụ thuộc vào cách lập trình.

Cảm biến siêu âm SR04 sử dụng nguyên lý phản xạ sóng siêu âm. Cảm biến gồm 2 module. 1 module phát ra sóng siêu âm và 1 module thu sóng siêu âm phản xạ về. Đầu tiên cảm biến sẽ phát ra 1 sóng siêu âm với tần số 40khz.

Nếu có chướng ngại vật trên đường đi, sóng siêu âm sẽ phản xạ lại và tác động lên module nhận sóng. Bằng cách đo thời gian từ lúc phát đến lúc nhận sóng ta sẽ tính được khoảng cách từ cảm biến đến chướng ngại vật.



### • Thông số kỹ thuật

- Điện áp hoạt động: + 5V
- Khoảng cách đo lý thuyết: 2cm đến 450cm
- Khoảng cách đo thực tế: 2cm đến 80cm
- Độ chính xác: 3mm
- Góc đo được bao phủ: <15 °
- Dòng điện hoạt động: <15mA
- Tần số hoạt động: 40Hz

### • Ứng dụng:

Được sử dụng để tránh và phát hiện các chướng ngại vật với các loại rô bốt như rô bốt hai chân, robot tránh chướng ngại vật, rô bốt dò đường, ...

Dùng để đo khoảng cách trong phạm vi rộng từ 2cm đến 400cm, đo độ sâu của một số nơi nhất định như giếng, hồ, ... vì sóng siêu âm có thể xuyên qua nước.

Có thể được sử dụng để lập bản đồ các đối tượng xung quanh cảm biến bằng cách xoay nó

## ❖ Nvidia Jetson TX1

### • Đặc điểm

Module Jetson TX1 là một module tích hợp cao cấp được phát triển bởi NVIDIA, được thiết kế đặc biệt cho ứng dụng trí tuệ nhân tạo và học sâu. Với một bộ xử lý NVIDIA Tegra X1, Jetson TX1 mang lại sức mạnh tính toán vượt trội và khả năng xử lý đồ họa đỉnh cao.

Nó cung cấp khả năng xử lý đa lõi, hỗ trợ trình diễn video 4K, học máy trên thiết bị và tích hợp các công nghệ như CUDA và cuDNN. Jetson TX1 cung cấp một môi trường phát triển linh hoạt với hỗ trợ đầy đủ cho các framework phổ biến như TensorFlow và PyTorch.

Với hiệu suất ấn tượng và khả năng tích hợp mạnh mẽ, module Jetson TX1 là lựa chọn lý tưởng cho việc phát triển các ứng dụng trí tuệ nhân tạo trong lĩnh vực nhúng.



### • Thông số kỹ thuật

- CPU: NVIDIA Tegra X1 64-bit quad-core ARM Cortex-A57
- GPU: NVIDIA Maxwell CUDA 256-core
- RAM: 4GB LPDDR4
- Bộ nhớ trong: 16GB eMMC
- Kết nối mạng: Gigabit Ethernet
- Kết nối không dây: Wi-Fi 802.11ac và Bluetooth 4.1
- Cổng giao tiếp: HDMI 2.0, USB 3.0, USB 2.0 Micro-B, USB 2.0 Type-A, UART, SPI, I2C, I2S, GPIOs
- Hệ điều hành: NVIDIA Linux for Tegra (L4T)
- Tiêu thụ năng lượng: Tối đa 10W

## ❖ Mạch Điều Khiển Động Cơ DC L298N

### • Đặc điểm

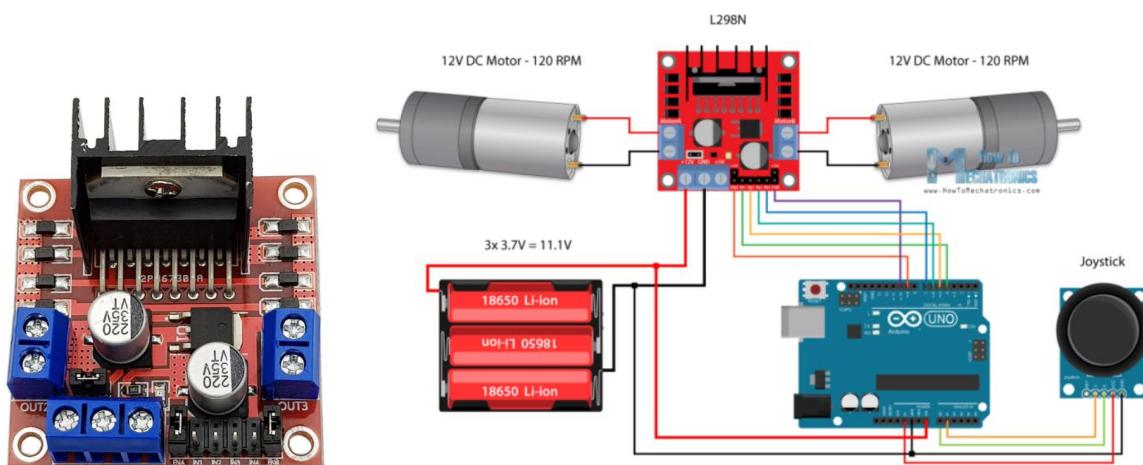
Mạch điều khiển động cơ DC L298N là một mạch điện tử được sử dụng để điều khiển động cơ DC hai chiều. Nó cung cấp khả năng điều khiển động cơ DC với dòng điện lớn và hỗ trợ các chức năng như thay đổi chiều quay và kiểm soát tốc độ.

Hỗ trợ điều khiển động cơ DC hai chiều với dòng điện lên đến 2A mỗi kênh. Nó có thể điều khiển hai động cơ DC độc lập hoặc một động cơ đa cấp, như động cơ bước hoặc động cơ servo.

Mạch L298N được điều khiển bằng cách gửi tín hiệu từ vi điều khiển như Arduino hoặc Raspberry Pi. Nó sử dụng giao tiếp chuẩn logic TTL với các chân IN1, IN2, IN3 và IN4 để điều khiển chiều quay và tốc độ của động cơ.

Để sử dụng mạch L298N, bạn cần kết nối đúng chân của nó với động cơ DC và vi điều khiển. Bạn cần cấu hình đúng các chân điều khiển để điều khiển động cơ theo yêu cầu của bạn, bao gồm chiều quay và tốc độ.

Mạch điều khiển động cơ DC L298N dễ sử dụng, chi phí thấp, dễ lắp đặt, là sự lựa chọn tối ưu trong tầm giá.



### • Thông số kỹ thuật:

- IC chính: L298 – Dual Full Bridge Driver.
- Điện áp đầu vào: 5~30VDC.
- Công suất tối đa: 25W 1 cầu (lưu ý công suất = dòng điện x điện áp nên áp cấp vào càng cao, dòng càng nhỏ, công suất có định 25W).
- Dòng tối đa cho mỗi cầu H là: 2A.
- Mức điện áp logic: Low -0.3V~1.5V, High: 2.3V~Vss.
- Kích thước: 43x43x27mm.

## ❖ Geared DC Motors & Wheels

### • Đặc điểm

Bộ khung xe 4 bánh Mecanum 25.5 x 15cm, 4 động cơ vàng được thiết kế phù hợp với các yêu cầu chế tạo robot dò đường, robot do thám, robot tránh vật cản, có thể lắp ghép với cánh tay robot để điều khiển... Là sản phẩm để làm robot mô hình, nghiên cứu khoa học, đặc biệt phù hợp với Arduino.

Bộ khung xe 4 bánh Mecanum 25.5 x 15cm, 4 động cơ vàng giúp di chuyển đa hướng theo nhiều phương: đi thẳng, đi ngang, đi theo phương chéo và xoay tròn... Xe được thiết kế đẹp mắt, khung kim loại chắc chắn, dễ dàng sử dụng với từng nhu cầu của mỗi người.

Bánh xe Mecanum là giải pháp hoàn hảo cho các sản phẩm hoạt động trong không gian nhỏ hẹp, bị giới hạn diện tích hoạt động và cần tiết kiệm thời gian thực hiện hành động của mình.

Bánh Xe Mecanum là một dạng của Bánh xe đa hướng (Mecanum Wheels), Bánh xe này được tạo thành bởi những con lăn nhỏ với trực được lắp nghiêng một góc 45 so với trực quay chính của bánh xe, nhờ kết cấu đặc biệt này, Bánh Xe Mecanum có thể tạo ra rất nhiều chuyển động phức hợp theo phương pháp cộng vector.



### • Thông số kỹ thuật

- 4 động cơ vàng: điện áp 6-9V
- 4 bánh xe mecanum: đường kính 60mm
- 1 khung xe: kim loại, dài x rộng: 25.5 x 15cm
- Ốc vít lắp đặt
- Trọng lượng: 300g

## ❖ Micro Servos

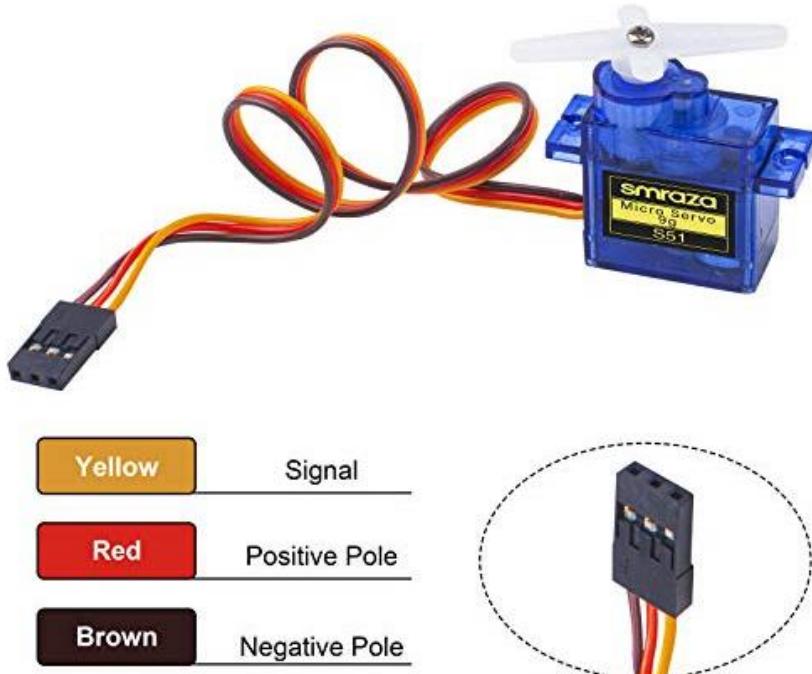
### • Đặc điểm

Động cơ servo SG90 có kích thước nhỏ, là loại được sử dụng nhiều nhất để làm các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng.

Động cơ servo SG90 180 độ có tốc độ phản ứng nhanh, các bánh răng được làm bằng nhựa nên cần lưu ý khi nâng tải nặng vì có thể làm hư bánh răng, động cơ RC Servo 9G có tích hợp sẵn Driver điều khiển động cơ bên trong nên có thể dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

Động cơ servo được thiết kế những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này.

Trong thực tế, bản thân động cơ servo không phải được điều khiển bằng vô tuyến, nó chỉ nối với máy thu vô tuyến trên máy bay hay xe hơi. Động cơ servo nhận tín hiệu từ máy thu này.



### • Thông số kỹ thuật

- Loại: SG90 – 360
- Kích thước: 23mmX12.5mmX29.5mm
- Trọng lượng: 9 gram
- Tốc độ không tải: 0,12 giây / 60 độ (4,8V)
- Mô-men xoắn chặn: 1,2-1,4 kg / cm (4,8V)
- Nhiệt độ hoạt động: -30 độ C ~ + 60 độ C
- Cài đặt vùng chốt: 7 micro giây
- Điện áp làm việc: 4,8V-6V

## ❖ Mạch thời gian thực RTC DS3231

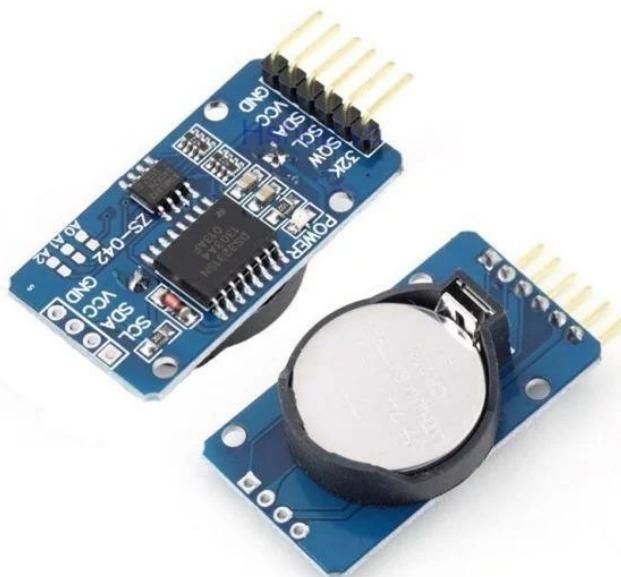
### • Đặc điểm

Là IC thời gian thực giá rẻ, rất chính xác với thạch anh tích hợp sẵn có khả năng điều chỉnh nhiệt. IC có đầu vào cho pin riêng, tách biệt khỏi nguồn chính đảm bảo cho việc giữ thời gian chính xác.

Thạch anh tích hợp sẵn giúp tăng độ chính xác trong thời gian dài hoạt động và giảm số lượng linh kiện cần thiết khi làm board.

Thời gian trong IC được giữ ở dạng: giờ, phút, giây, ngày, thứ, tháng, năm. Các tháng có ít hơn 31 ngày sẽ tự động được điều chỉnh, các năm nhuận cũng được chỉnh đúng số ngày.

Thời gian có thể hoạt động ở chế độ 24h hoặc 12h AM/PM. IC còn có chức năng báo động, có thể cài đặt 2 thời gian báo và lịch, có tín hiệu ra là xung vuông. Giao tiếp với IC được thực hiện thông qua I2C bus.



### • Thông số kỹ thuật:

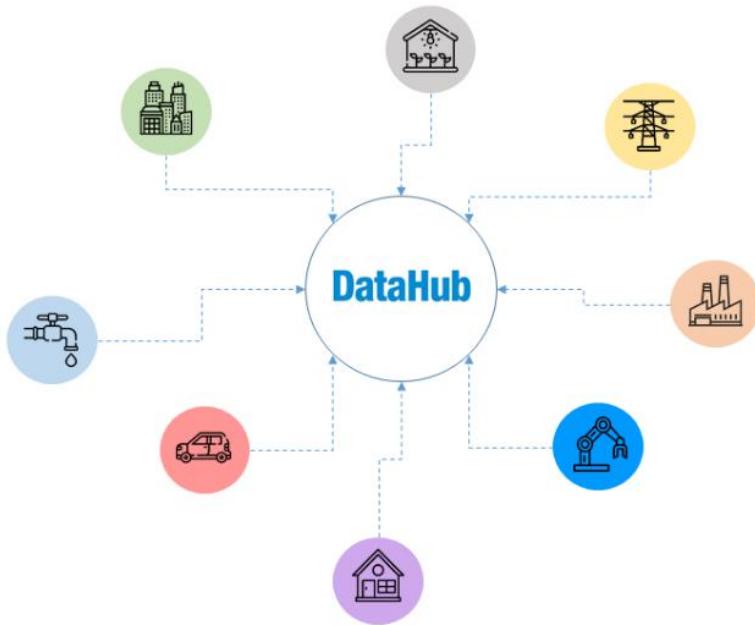
- IC chính: RTC DS3231 + EEPROM AT24C32
- Nguồn cung cấp: 3.3~5VDC.
- Giao tiếp: I2C
- Lưu trữ và cung cấp các thông tin thời gian thực: ngày, tháng, năm, giờ, phút, giây,...
- Có pin backup duy trì thời gian trong trường hợp không cấp nguồn.
- Kích thước: 38 x 22 x 14mm

### 3.2 Công cụ phần mềm

Giới thiệu về các công cụ phần mềm được sử dụng trong đề tài.

STT	Tên Phần Mềm	Phân loại
1	DataHub	WISE-PaaS
2	DashBoard	WISE-PaaS
3	Notification	WISE-PaaS
4	Arduino IDE	Ứng dụng lập trình
5	Pycharm	Ứng dụng lập trình
6	Spyder	Ứng dụng lập trình
7	Blynk IoT	Ứng dụng điều khiển

#### 3.2.1 DataHub



Datahub của Wise-PaaS là một nền tảng dựa trên đám mây cung cấp các dịch vụ phân tích và quản lý dữ liệu toàn diện cho IoT và các ứng dụng công nghiệp. Nó cung cấp nhiều chức năng như thu thập, lưu trữ, xử lý, phân tích và trực quan hóa dữ liệu.

Datahub cũng hỗ trợ nhiều nguồn dữ liệu khác nhau như cảm biến, thiết bị, cơ sở dữ liệu và dịch vụ đám mây, cho phép người dùng tích hợp và quản lý dữ liệu từ các nguồn khác nhau một cách thống nhất.

Với giao diện thân thiện với người dùng, người dùng có thể dễ dàng định cấu hình và giám sát luồng dữ liệu, thiết lập mô hình phân tích dữ liệu và tạo báo cáo dữ liệu lịch sử và thời gian thực.

Nhìn chung, Datahub là một công cụ mạnh mẽ để quản lý và phân tích dữ liệu trên đám mây, cho phép các doanh nghiệp thu được những hiểu biết có giá trị từ IoT và các ứng dụng công nghiệp của họ.

### 3.2.2 DashBoard



Bảng điều khiển của Wise-PaaS đóng vai trò là trung tâm quản lý toàn diện cho các ứng dụng IoT công nghiệp (IIoT).

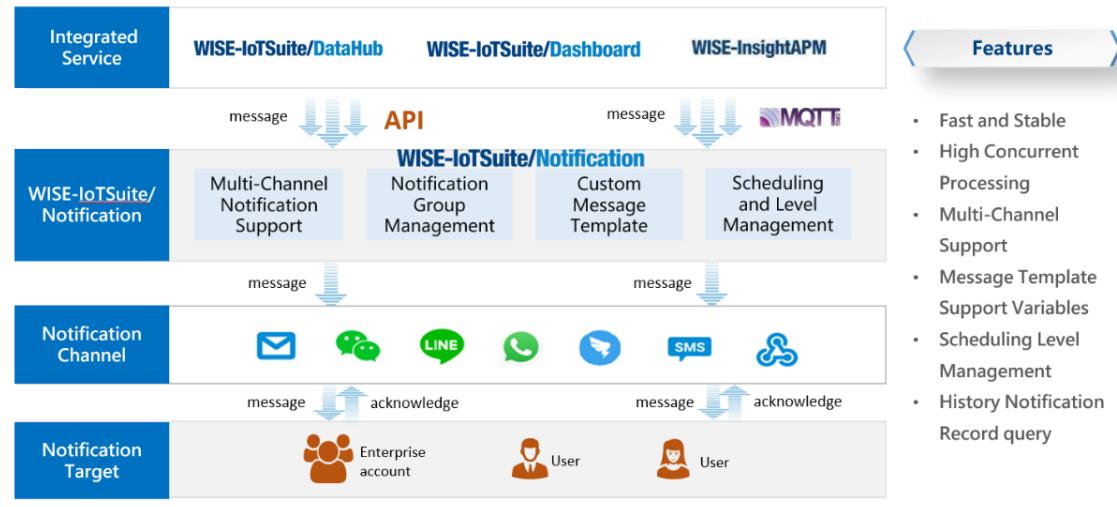
Nó cung cấp giao diện thân thiện với người dùng để giám sát, phân tích và điều khiển các thiết bị và hệ thống kết nối. Bảng điều khiển cung cấp khả năng xem thời gian thực vào các chỉ số quan trọng, chỉ số hiệu suất và phân tích dữ liệu.

Người dùng có thể dễ dàng theo dõi trạng thái, sức khỏe và hiệu suất hoạt động của thiết bị, giúp duy trì và khắc phục sự cố một cách chủ động.

Nó cung cấp các công cụ hình ảnh hóa dữ liệu tương tác như biểu đồ, đồ thị và bảng điều khiển tùy chỉnh để trình bày thông tin một cách hấp dẫn và dễ hiểu.

Với bảng điều khiển Wise-PaaS, người dùng có thể tối ưu hóa hoạt động IIoT, đưa ra quyết định có căn cứ và tăng năng suất trong môi trường công nghiệp của họ.

### 3.2.3 Notification



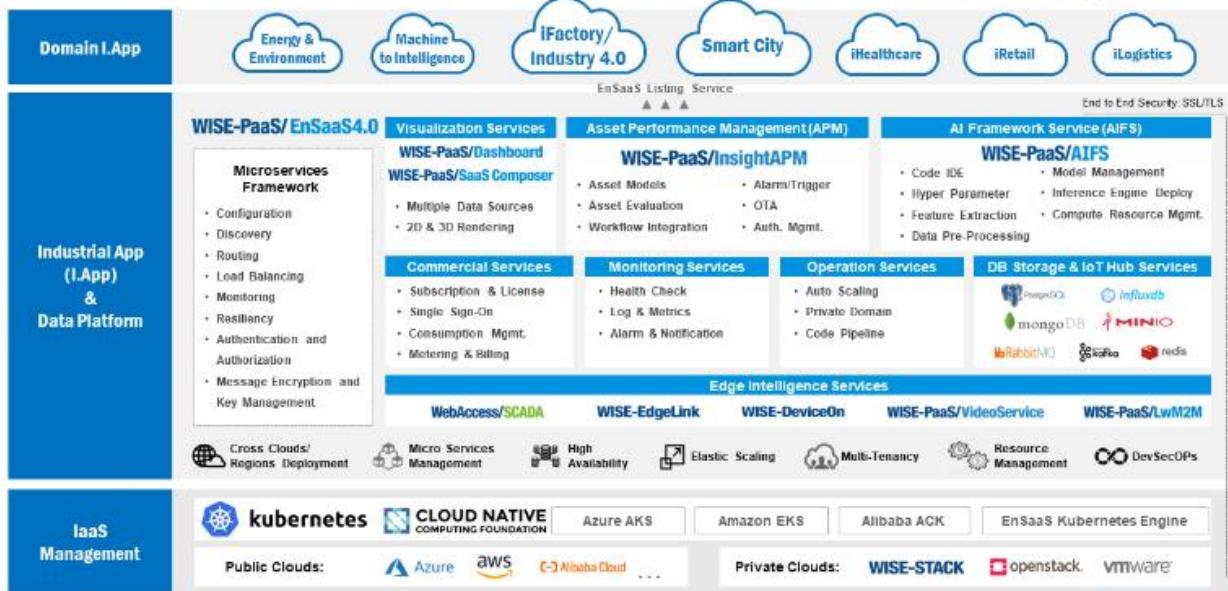
Thông báo trong Wise-PaaS là một tính năng quan trọng cho phép người dùng nhận được thông báo và cập nhật quan trọng liên quan đến các ứng dụng và thiết bị IoT công nghiệp (IIoT) của họ.

Những thông báo này cung cấp thông tin thời gian thực về sự kiện hệ thống, các tình huống bất thường và các điều kiện quan trọng đòi hỏi sự chú ý ngay lập tức.

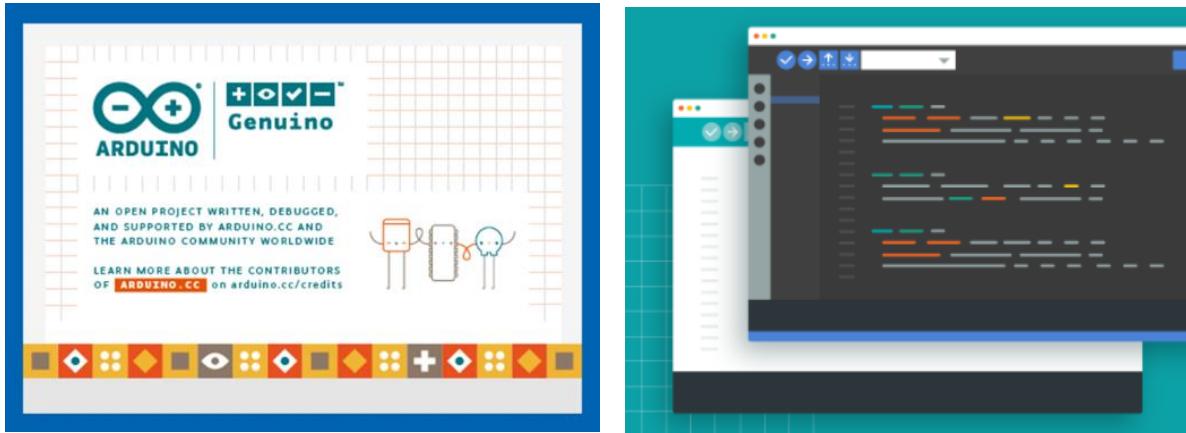
Người dùng có thể cấu hình thông báo để nhận các cảnh báo qua các kênh khác nhau như email, tin nhắn SMS hoặc thông báo đẩy trên điện thoại di động. Thông báo có thể được kích hoạt dựa trên ngưỡng được xác định trước, các chỉ số hiệu suất hoặc các sự kiện cụ thể.

Bằng cách tận dụng thông báo trong Wise-PaaS, người dùng có thể theo dõi IIoT một cách chủ động, được thông báo về các vấn đề hoạt động và thực hiện các biện pháp kịp thời để đảm bảo hoạt động suôn sẻ của quy trình công nghiệp của họ.

### WISE-PaaS 4.0 Architecture with Microservices & I.App



### 3.2.4 Arduino IDE



IDE trong Arduino IDE là phần có nghĩa là mã nguồn mở, nghĩa là phần mềm này miễn phí cả về phần tải về lẫn phần bản quyền: Người dùng có quyền sửa đổi, cải tiến, phát triển, nâng cấp theo một số nguyên tắc chung được nhà phát hành cho phép mà không cần xin phép ai, điều mà họ không được phép làm đối với các phần mềm nguồn đóng.

Tuy là phần mềm mã nguồn mở nhưng khả năng bảo mật thông tin của Arduino IDE là vô cùng tuyệt vời, khi phát hiện lỗi nhà phát hành sẽ vá nó và cập nhật rất nhanh khiến thông tin của người dùng không bị mất hoặc rò rỉ ra bên ngoài.

Có thể chạy trên nhiều hệ điều hành như Windows, macOS và Linux, giúp người dùng trên các nền tảng khác nhau có thể sử dụng và phát triển dự án Arduino.

Đi kèm với một số thư viện tiêu chuẩn và ví dụ sẵn có giúp người dùng dễ dàng thực hiện các chức năng và kỹ thuật khác nhau. Bạn cũng có thể tải thêm các thư viện mở rộng từ cộng đồng Arduino.

### 3.2.5 Pycharm



PyCharm là một môi trường phát triển tích hợp (IDE) phổ biến được thiết kế đặc biệt cho lập trình Python. Được phát triển bởi JetBrains, PyCharm cung cấp một bộ công cụ và tính năng toàn diện để tối ưu quy trình phát triển Python.

Nó cung cấp các tính năng chỉnh sửa mã, gỡ lỗi, kiểm thử và quản lý dự án, là một công cụ mạnh mẽ cho cả những lập trình viên Python mới bắt đầu và có kinh nghiệm.

PyCharm bao gồm tính năng hoàn thành mã thông minh, làm nổi bật cú pháp, điều hướng mã và các công cụ tái cấu trúc để tăng năng suất. Nó cũng hỗ trợ các framework và thư viện phổ biến được sử dụng trong phát triển Python.

Với giao diện thân thiện với người dùng và tính năng mạnh mẽ, PyCharm giúp lập trình viên viết mã Python sạch và hiệu quả, xử lý sự cố và xây dựng ứng dụng Python chất lượng cao.



### 3.2.6 Blynk IoT

Blynk là một nền tảng với các ứng dụng iOS và Android để điều khiển Arduino, Raspberry Pi và các ứng dụng tương tự qua Internet.

Nó là một bảng điều khiển kỹ thuật số nhờ đó bạn có thể xây dựng giao diện đồ họa cho dự án của mình bằng cách kéo và thả các widget. Việc thiết lập mọi thứ rất đơn giản dễ dàng với người mới sử dụng.

Blynk không bị ràng buộc với một số board hoặc shield cụ thể. Blynk hỗ trợ rất nhiều từ Raspberry, Esp 32, Esp8266 và rất nhiều vi xử lý khác.

Blynk hỗ trợ nhiều giao thức kết nối, bao gồm Wi-Fi, Ethernet, Bluetooth và 2G/3G thông qua các mạch phụ kiện như Arduino, Raspberry Pi và ESP8266. Điều này cho phép bạn kết nối và điều khiển các thiết bị từ bất kỳ đâu, qua mạng Internet.

### 3.2.7 Spyder



Spyder là Môi trường phát triển tích hợp (IDE) được thiết kế chủ yếu cho lập trình khoa học bằng Python.

Đây là một phần mềm nguồn mở cho phép các nhà phát triển truy cập nhiều công cụ và tính năng để viết, kiểm tra và gỡ lỗi mã một cách hiệu quả và nhanh chóng.

Spyder IDE bao gồm một trình chỉnh sửa mã, một bảng điều khiển, trình khám phá biến và trình gỡ lỗi, làm cho nó trở thành một lựa chọn ưu tiên cho các lập trình viên khoa học.

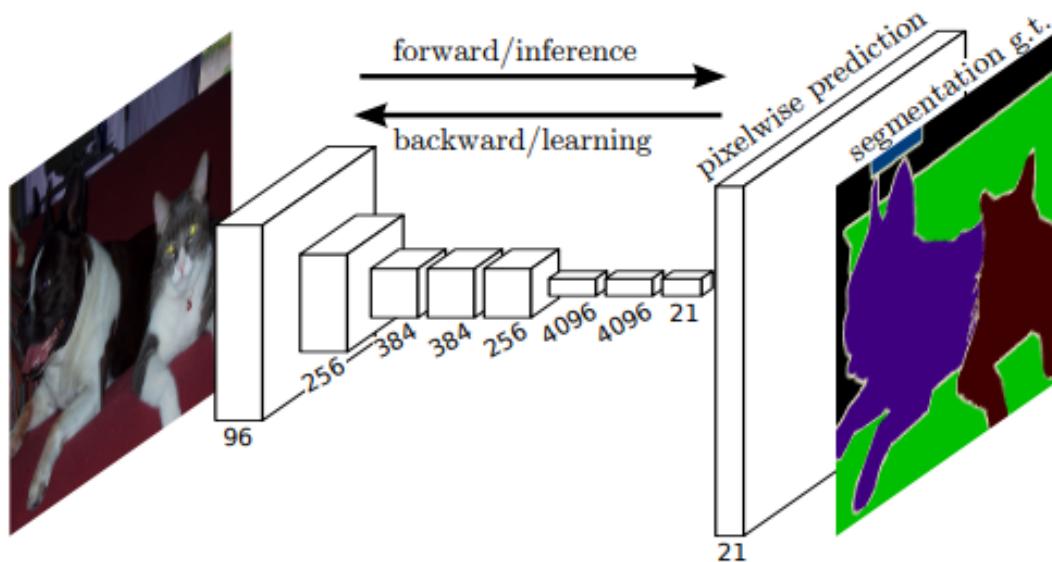
Nó cung cấp các tính năng như tự động hoàn thành, gấp mã và công cụ đánh dấu cú pháp giúp nâng cao trải nghiệm viết mã. Ngoài ra, Spyder IDE tương thích với nhiều thư viện khoa học khác nhau như NumPy, SciPy và Matplotlib.

Nó là một IDE đáng tin cậy cho các nhà phát triển muốn phát triển các ứng dụng khoa học và phân tích dữ liệu.

## PHẦN IV: THUẬT TOÁN, KỸ THUẬT SỬ DỤNG

STT	Nội dung
1	Semantic Segmentation
2	U-net
3	Obstacle Detection
4	Obstacle Avoiding
5	Planning Algorithms for Outdoor Environment
6	Dijkstra Algorithms
7	Building Maps

### 4.1 Semantic Segmentation



Semantic segmentation là quá trình phân loại từng pixel trong một hình ảnh thành các lớp tương ứng với các đối tượng khác nhau.

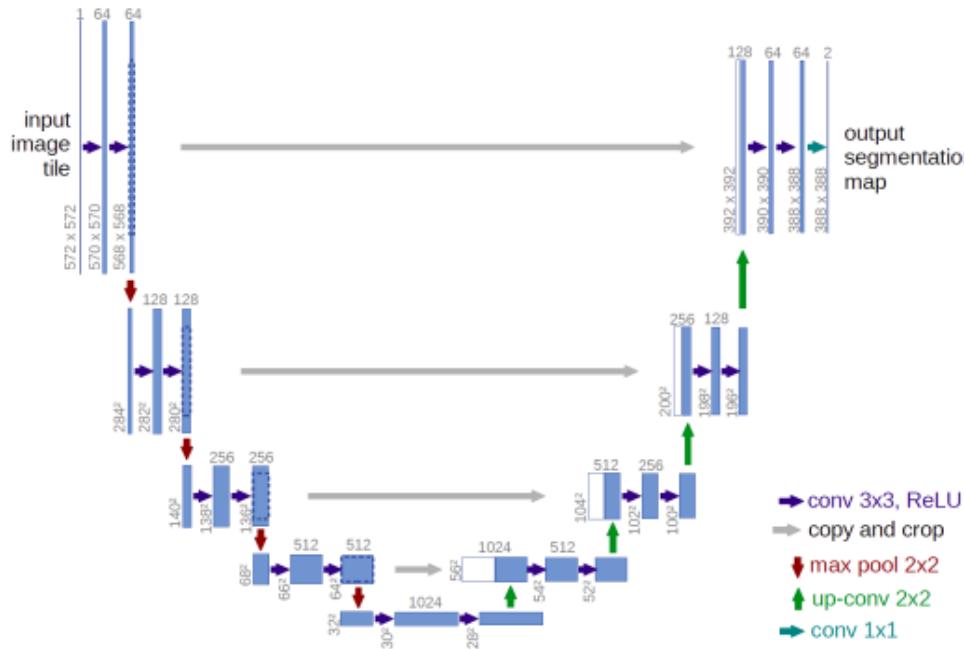
Nó giúp cho máy tính có khả năng hiểu được vật thể, đối tượng có trong hình ảnh bằng cách sử dụng các kỹ thuật Deep Learning.

Mục đích của semantic segmentation là giúp cho máy tính có thể hiểu được cấu trúc và nội dung của hình ảnh để thực hiện các tác vụ như phát hiện đối tượng, theo dõi và phân tích hành vi của chúng.

Ví dụ như áp dụng semantic segmentation vào nhận diện xe cộ, chúng ta có thể xác định được vị trí và loại xe cộ trong hình ảnh.

Các ứng dụng của semantic segmentation rất đa dạng và có thể được sử dụng trong nhiều lĩnh vực như y tế, an ninh, tự động hóa sản xuất, xe tự hành và nhiều lĩnh vực khác nữa.

## 4.2 U-net



Kiến trúc Unet được đề xuất bởi Ronneberger et al. vào năm 2015 cho bài toán semantic segmentation. Unet là một kiến trúc mạng lưới sâu học sử dụng kỹ thuật skip-connection, cho phép nó giữ lại thông tin từ các lớp ở các cấp độ khác nhau, từ đó cải thiện hiệu suất.

Unet có hình dạng như chữ U, gồm hai phần chính: chiều rộng ở đầu vào của mạng và chiều cao cuối cùng của mạng. Phần đầu tiên gồm các lớp tích chập (convolutional layers) và lớp gộp (pooling layers) để nén ảnh thành biểu diễn có số chiều thấp hơn nhưng vẫn giữ lại thông tin quan trọng. Phần thứ hai gồm các lớp tích chập và lớp kích hoạt (activation layers) được sử dụng để tăng kích thước của đầu ra, cho đến khi đạt đến kích thước của ảnh ban đầu.

Kết hợp với kỹ thuật skip-connection, Unet cho phép truy cập thông tin từ các lớp đầu vào của mạng để cải thiện độ chính xác và loại bỏ hiện tượng lõi biên (border-effect) trong quá trình dự đoán nhãn của các đối tượng nhỏ hơn so với kích thước của vùng ảnh mà mạng được huấn luyện trên đó.

Unet đã trở thành một trong những kiến trúc rất phổ biến được sử dụng để giải quyết bài toán semantic segmentation và một số bài toán khác như detection.

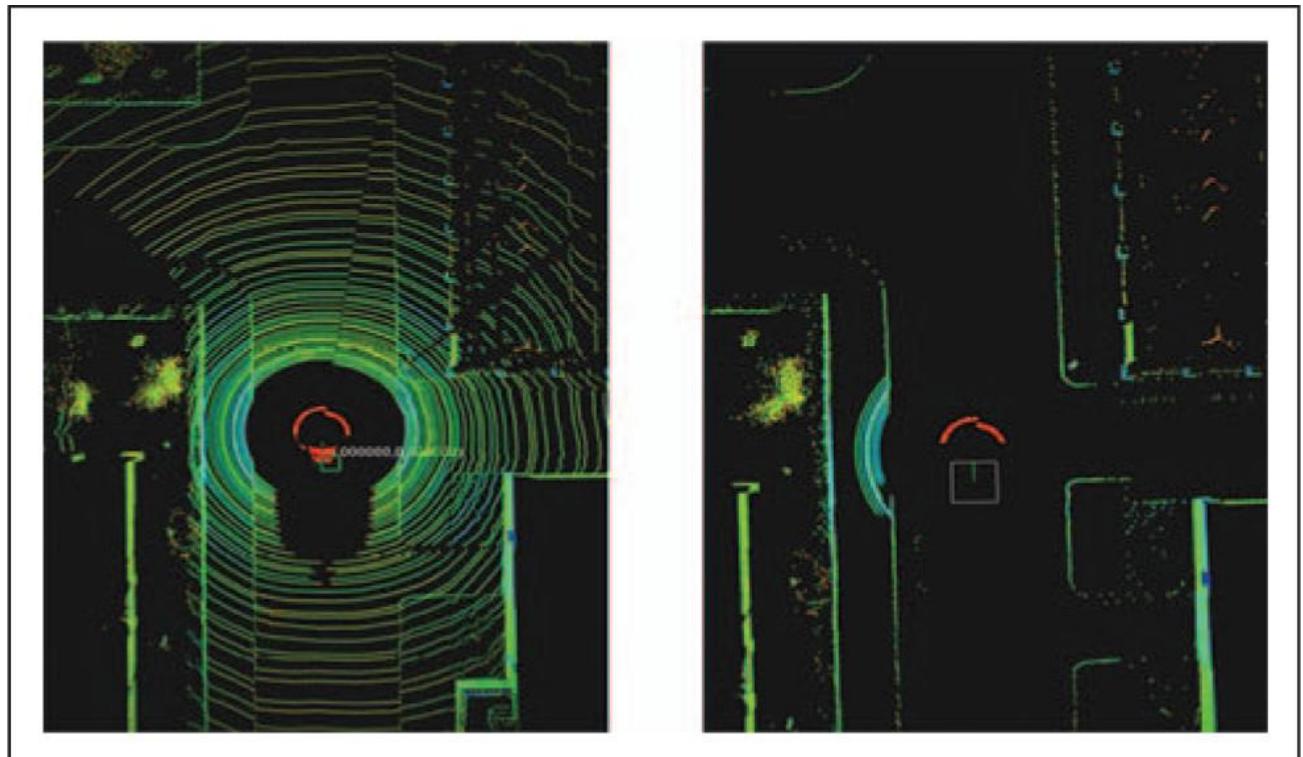
### 4.3 Obstacle Detection



Obstacle Detection là một công nghệ hoặc hệ thống giúp nhận diện và phát hiện các vật cản trên đường, trong một không gian hoặc một khu vực nhất định.

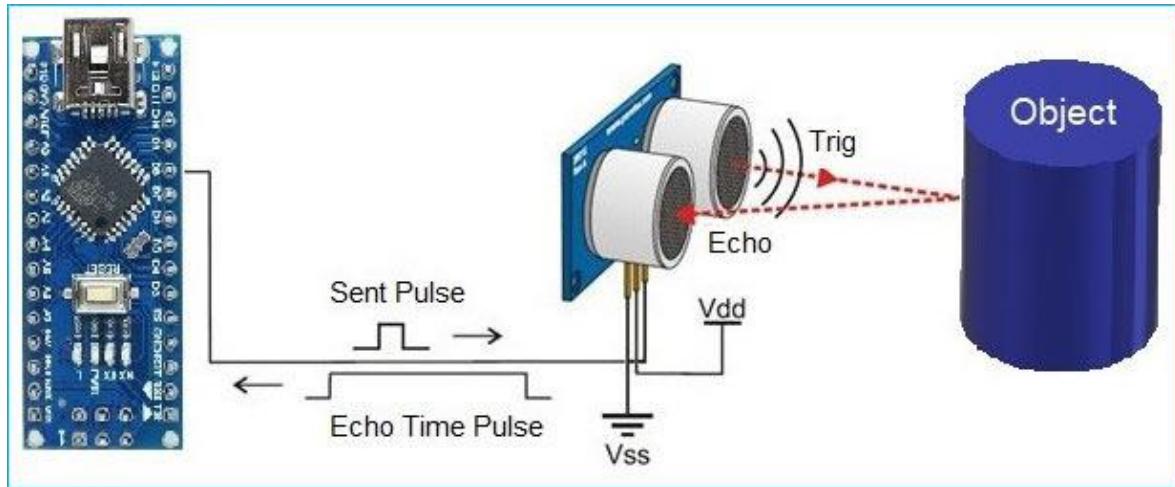
Nó được sử dụng để đảm bảo an toàn cho các phương tiện di chuyển, như xe hơi, máy bay, tàu hỏa, robot và các thiết bị tự động khác.

Công nghệ Obstacle Detection sử dụng các công cụ như cảm biến, radar, camera, lidar và ultrasonic để nhận diện và phát hiện các vật cản trong khoảng cách gần hoặc xa.



Hình bên trái là dữ liệu đám mây điểm ban đầu hình bên phải là kết quả phân đoạn đường.

## 4.4 Obstacle Avoiding

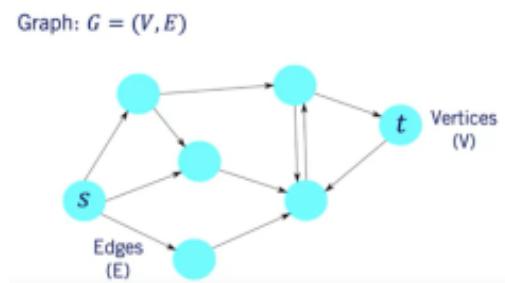
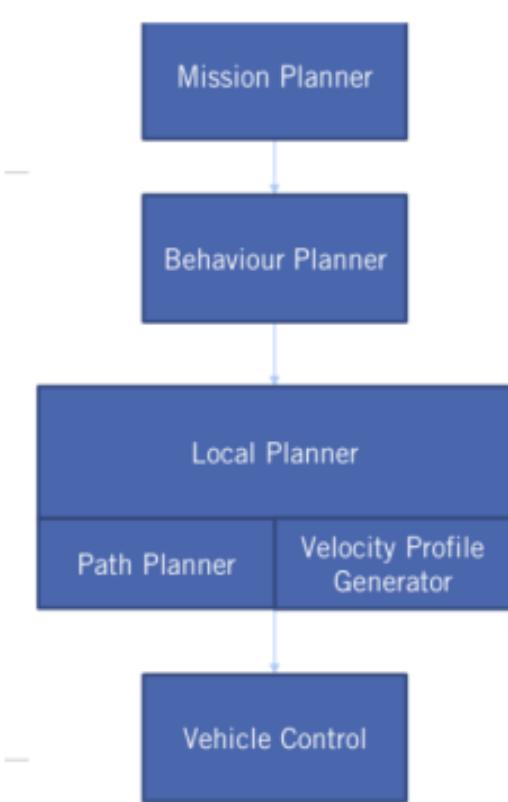


Obstacle Avoiding là một chức năng được sử dụng trong các thiết bị tự động hóa, robot hoặc các phương tiện di chuyển tự động.

Chức năng này giúp thiết bị có khả năng phát hiện và tránh né các vật cản, trở ngại trong quá trình thực hiện tác vụ di chuyển.

Các phương tiện sử dụng chức năng Obstacle Avoiding sẽ được trang bị cảm biến và mạch điều khiển đặc biệt để phát hiện và phản ứng kịp thời khi gặp các vật cản trong quá trình di chuyển.

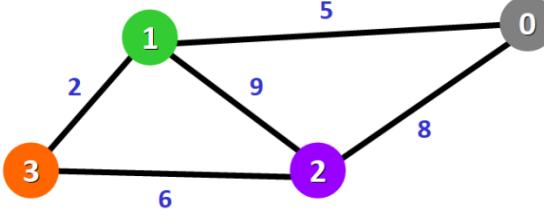
## 4.5 Planning Algorithms for Outdoor Environment



Planning Algorithms for Outdoor Environment là các thuật toán được sử dụng để lập kế hoạch và điều chỉnh hoạt động của các robot hoặc phương tiện tự hành trong môi trường ngoài trời.

Các thuật toán này được thiết kế để xử lý các thách thức đặc biệt của môi trường bên ngoài như địa hình khó khăn, sự thay đổi của thời tiết và tình trạng của địa hình.

## 4.6 Dijkstra Algorithms

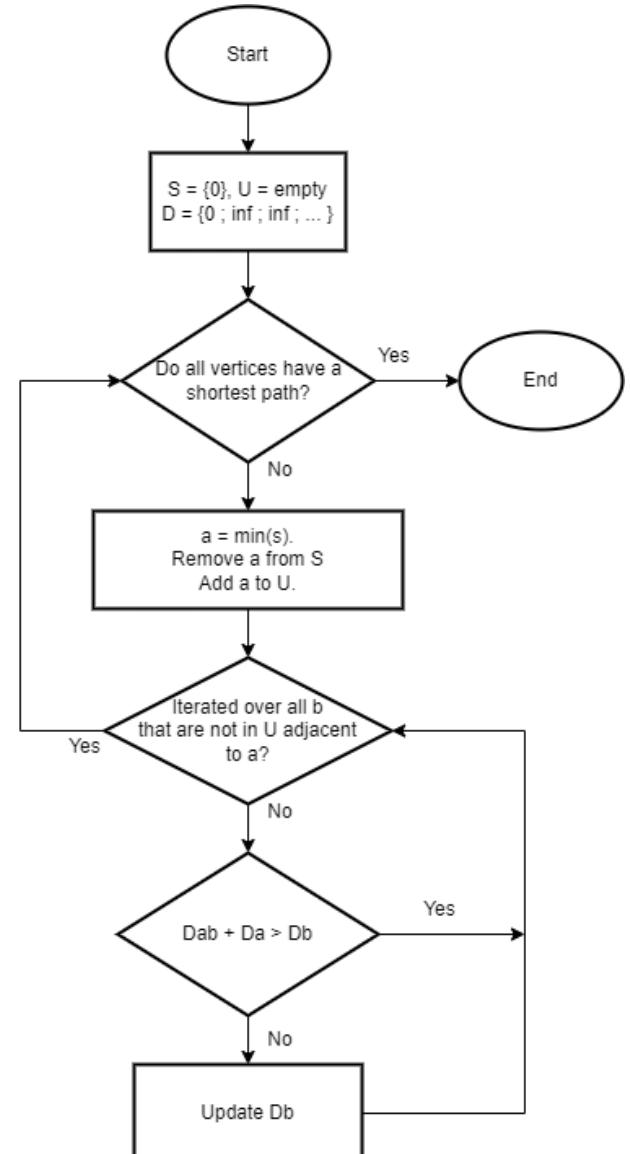


Step	0	1	2	3
0	0	inf	inf	inf
1	-	5	8	inf
3	-	-	8	7
4	-	-	13	-

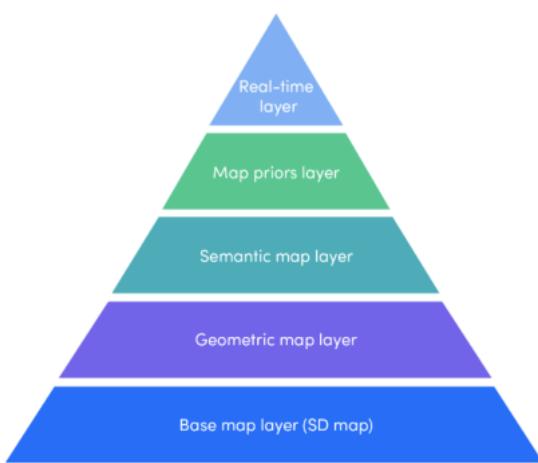
Dijkstra Algorithms là một thuật toán tìm đường đi ngắn nhất từ một đỉnh đến tất cả các đỉnh còn lại trong đồ thị có trọng số dương.

Thuật toán này sử dụng cơ chế liên quan đến các đỉnh đã được xử lý và các đỉnh chưa được xử lý để tìm ra đường đi ngắn nhất từ đỉnh ban đầu đến các đỉnh còn lại.

Thuật toán được đặt theo tên của nhà toán học Edsger W. Dijkstra vào năm 1956.



## 4.7 Building Maps



Building Maps là quá trình xây dựng bản đồ chi tiết của một khu vực nhất định để xe tự hành có thể dựa trên đó để quyết định hành động của mình.

Quá trình này bao gồm sự liên tục thu thập dữ liệu từ các cảm biến và hình ảnh camera của xe tự hành, sau đó phân tích và xử lý các dữ liệu này để tạo ra bản đồ chính xác nhất có thể.

Một bản đồ chi tiết giúp cho xe tự hành có thể định vị chính xác và chính xác đường đi của nó, giảm thiểu nguy cơ tai nạn và tăng khả năng đến đích an toàn và hiệu quả. Diễn hình cho quá trình này là Google Maps hoặc OpenStreetMap.

## PHẦN V: GÓI DỮ LIỆU

	Data Field
Client	- Information ( Name, Phone Number, Location, Time )
Robot	- QR Code - Video Stream - GPS - Notification ( Problem, Begin, Finish, Available )
Host	- Command - Confirmation

### 5.1 Client

Về dữ liệu của khách hàng, chúng ta có các dữ liệu về thông tin cơ bản của khách hàng cần cung cấp để quản lý hệ thống, quản lý đơn hàng.

Các thông tin đó ( gọi chung là Information ) gồm các gói dữ liệu con như:

- Họ và tên khách hàng ( Name ): dùng để xác định cũng như thiết lập các mã QR code.
- Số điện thoại khách hàng ( Phone Number ): dùng để liên hệ khách hàng để xác nhận đơn hàng cũng như thông báo về những vấn đề cần thiết về đơn hàng vận chuyển.
- Địa chỉ ( Location ): dùng để cập nhập địa chỉ giao hàng, nhận hàng.
- Thời gian ( Time ): cung cấp thời gian mà khách hàng muốn giao hàng, nhận hàng.

### 5.2 Robot

Dữ liệu của Robot được dùng để cung cấp cho Khách hàng và Máy chủ về thông tin hành trình vận chuyển. Bao gồm các gói dữ liệu như:

- Mã QR ( QR Code ): được dùng để định danh đơn hàng vận chuyển. Khách hàng quét mã QR này để xác nhận đúng đơn hàng và nhận hàng.
- Camera hành trình (Video Stream ): dùng để phát trực tiếp hành trình vận chuyển của Robot để Khách hàng và Máy chủ có thể cập nhập các thông tin kịp lúc nhất là khi Robot xảy ra vấn đề có thể kịp thời giải quyết, đảm bảo chất lượng đơn hàng.
- Định vị ( GPS ): dùng để cập nhập liên tục vị trí của Robot nhằm đảm bảo Robot di chuyển đúng phần đường định sẵn cũng như quá trình vận chuyển.

- Thông báo ( Notification ): được dùng để gửi thông báo ( tin nhắn ) hoặc gọi điện cho Khách hàng và Máy chủ khi:
  - + Robot gặp vấn đề ( Problem ).
  - + Robot bắt đầu quá trình vận chuyển ( Begin ).
  - + Robot kết thúc quá trình vận chuyển ( Finish ).
  - + Món hàng vận chuyển có hiện hữu trong Robot ( Available ).  
( Phòng trường hợp có kẻ gian cố tình lấy cắp món hàng )

### 5.3 Host

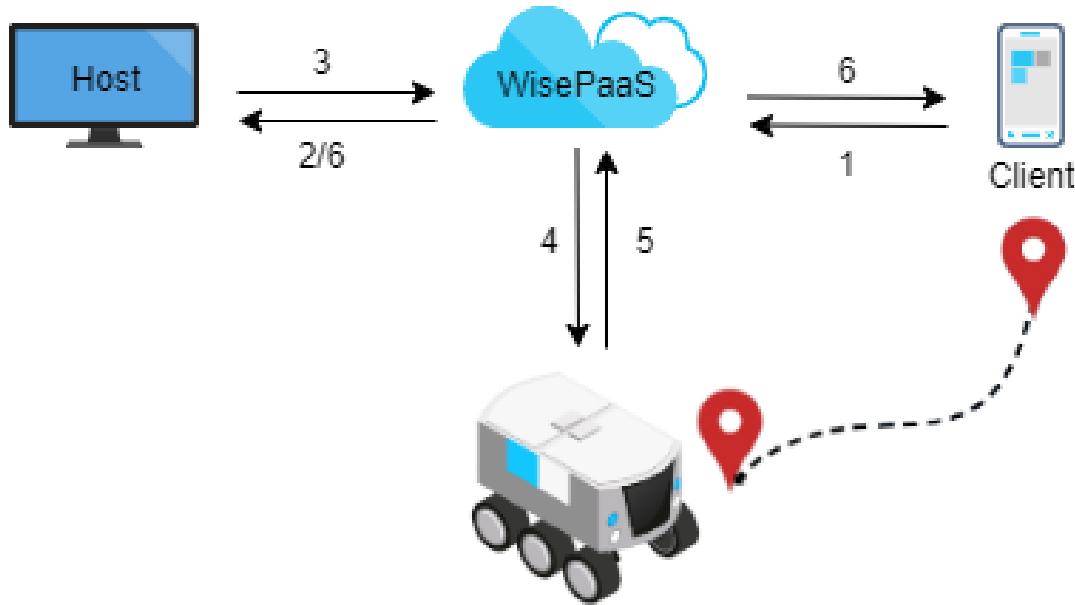
Dữ liệu của Máy chủ sẽ bao gồm 2 phần chính đó là câu lệnh ( Command ) và xác nhận ( Confirmation ) để quản lý dữ liệu của Khách hàng và Robot. Đảm bảo việc vận chuyển thành công.

## PHẦN VI: THIẾT KẾ VÀ MÔ HÌNH HOẠT ĐỘNG

### 6.1 Hệ thống

“Đề tài này được chia thành 2 thành phần có tác dụng hỗ trợ nhau gồm: Xe tự hành vận chuyển và WISE-PaaS”

#### 6.1.1 Sơ đồ hoạt động chung của hệ thống



Hệ thống sử dụng WisePaaS làm nền tảng lưu trữ dữ liệu trung gian giữa Máy chủ, Khách hàng và Robot.

Các thông tin về khách hàng, về giao dịch sẽ được gửi từ Khách hàng ( Ứng dụng người dùng ) đến WisePaaS ( Cloud ).

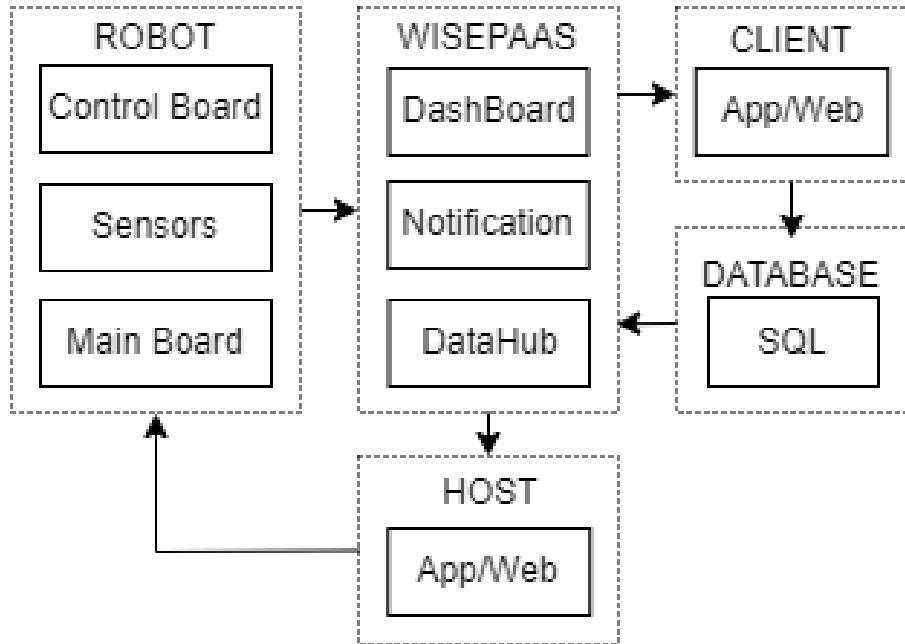
Sau đó Máy chủ sẽ tiếp nhận các thông tin đó ( Trang web quản lý của Máy chủ ) xử lý dữ liệu, xác nhận đơn hàng và gửi trả về WisePaaS sau khi đã hoàn tất các thủ tục để có thể thực hiện giao dịch.

Và Robot sẽ lấy các dữ liệu mà nó cần như: Thông tin khách hàng, Thông tin về giao dịch,... từ WisePaaS để bắt đầu quá trình vận chuyển và gửi ngược lại những thông tin hành trình như GPS, Camera và các thông số khác.

Khách hàng và Máy chủ sẽ cập nhật các thông tin hành trình của Robot thông qua WisePaaS.

Quá trình truyền, nhận dữ liệu giữa WisePaaS và Khách hàng sẽ kết thúc khi Khách hàng xác nhận nhận hàng thành công.

### 6.1.2 Sơ đồ chi tiết:



#### Thành phần hệ thống:

- Robot sẽ bao gồm các thanh phần như Control Board, Sensors, Main Board.
- WisePaaS sẽ sử dụng các ứng dụng như DataHub, DashBoard, Notification.
- Khách hàng, Máy chủ sẽ được hỗ trợ App, Web để cập nhập thông tin cũng như quản lí, nhập lệnh,...
- SQL có nhiều hệ quản trị cơ sở dữ liệu phổ biến như MySQL, PostgreSQL, Oracle, SQL Server và SQLite. Mỗi hệ quản trị cơ sở dữ liệu có cú pháp và tính năng riêng, nhưng ngôn ngữ SQL chung cho phép thực hiện các hoạt động cơ bản như truy vấn dữ liệu, thêm, sửa đổi và xóa bản ghi.

#### Quá trình truyền dữ liệu:

- Robot gửi và nhận các dữ liệu đến WisePaaS.
- Máy chủ và Khách hàng sẽ lấy những dữ liệu này về Web/App để hiển thị cho người dùng.
- Những dữ liệu, câu lệnh được bổ sung, thực hiện bởi người dùng trên app/web sẽ được truyền qua database SQL trung gian để có thể đưa dữ liệu về lại WisePaaS để cập nhập cho Robot.
- Tất cả đều được truyền qua các giao thức truyền thông không dây phổ biến đã được chuẩn hóa

( Vì WisePaaS không thể cập nhập dữ liệu lên DataHub từ App/Web thông qua Wifi như Firebase nên chúng ta phải sử dụng một database trung gian khác có thể tương tác qua lại với WisePaaS đó là SQL. )

### 6.1.3 Hệ thống Pipeline:



Để làm cho rô-bốt hoàn toàn tự động, rô-bốt phải hoàn toàn nhận thức được môi trường xung quanh và phải có khả năng thực hiện các hành động dựa trên thông tin đầu vào mà nó nhận được thông qua các mô-đun khác nhau của hệ thống.

Để đạt được trạng thái tự chủ hoàn toàn, robot phải có khả năng lấy thông tin từ các cảm biến, nhận thức môi trường, định vị chính xác bản thân trên thế giới và cuối cùng đưa ra kế hoạch tối ưu để đạt được mục tiêu.

Các hướng dẫn đạt được từ các mô-đun được đề cập ở trên phải được rô-bốt tích hợp trong thời gian thực và được cung cấp cho một nút điều khiển để thực sự di chuyển hệ thống trong thế giới thực.

Độ chính xác và sự tích hợp phù hợp của tất cả các mô-đun là vô cùng quan trọng đối với một rô-bốt tự động hoạt động. Một lỗi trong bất kỳ mô-đun nào có thể gây ra hậu quả nghiêm trọng và thậm chí có thể gây nguy hiểm cho những người xung quanh. Luận án này nhằm mục đích thực hiện hoàn hảo tất cả các mô-đun đã đề cập để đạt được hoạt động hoàn toàn tự động của robot trong môi trường ngoài trời.

Mô-đun lập kế hoạch là xương sống của hệ thống lái xe tự hành. Người lập kế hoạch chịu trách nhiệm tìm các đường đi tối ưu trên bản đồ để rô-bốt di chuyển và tạo quỹ đạo hiệu quả cũng như hò sơ vận tốc để rô-bốt di chuyển cục bộ khi có chướng ngại vật tĩnh/động, tuân thủ quy tắc làn đường, ưu tiên sự an toàn của con người,...

## 6.2 Xe tự hành vận chuyển

Robot được thiết kế để hoạt động như một nền tảng robot di động tự động cho các ứng dụng khác nhau như trong các khu công nghiệp để vận chuyển, trong bệnh viện để vận chuyển thực phẩm và thuốc men, trong các nhiệm vụ không người lái và cũng vì mục đích an ninh.

Mục tiêu của chúng tôi là phát triển một robot di động giao hàng tự động có thể tự động giao một gói hàng từ A đến B trong khuôn viên.

Cấu trúc phần cứng và phần mềm của robot đã được thiết kế bằng cách xem xét các yếu tố trên tiêu chuẩn.

### 6.2.1 Tiêu chí thiết kế:

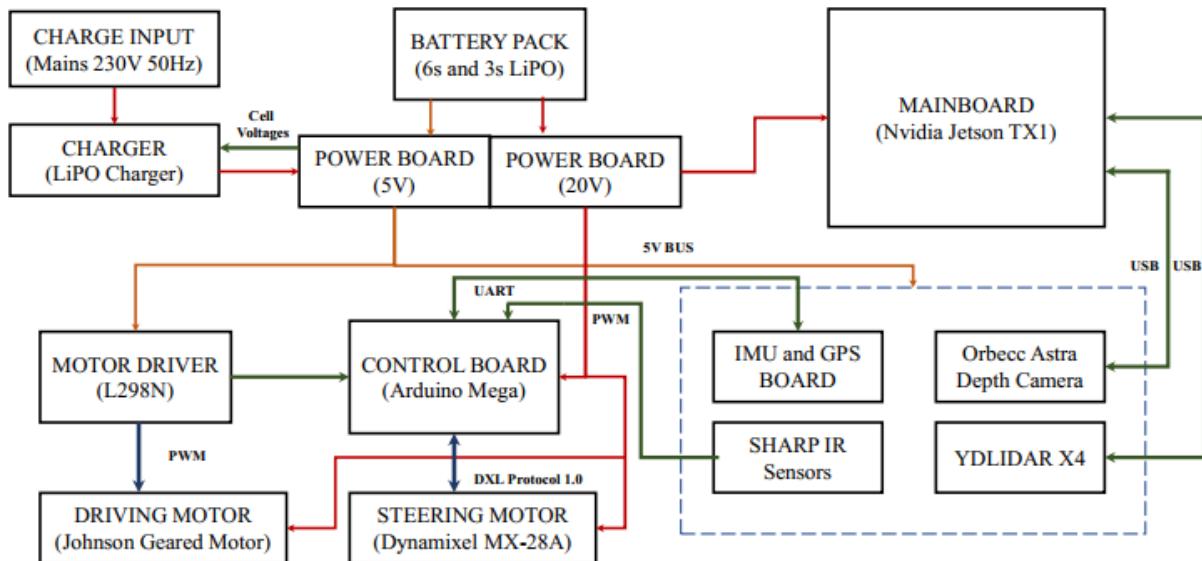
Có năm hạn chế chính trong thiết kế phần cứng.

1. **Tính mô-đun:** Để dễ dàng thêm các đơn vị hoặc bộ phận vào robot, nên robot phải ở dạng mô-đun. Các đơn vị hoặc bộ phận này có thể là cảm biến điều hướng bổ sung, chỗ cho tải trọng, nguồn điện bổ sung trên tàu hoặc các thiết bị khác nhau để giao diện người-máy hiệu quả.
2. **Chi phí sản xuất thấp:** Mặc dù robot di động có sẵn trên thị trường nhưng chúng có xu hướng đắt tiền, do đó làm tăng chi phí nghiên cứu và phát triển. Hơn nữa, việc sử dụng robot làm sẵn sẽ làm tăng chi phí sản xuất hơn nữa trong trường hợp sản xuất hàng loạt.
3. **Cấu trúc rút gọn:** Trong giai đoạn tạo mẫu, cấu trúc được giữ đơn giản và rút gọn để sử dụng tài nguyên tối thiểu và tập trung vào chức năng được chỉ định.
4. **Tính phù hợp với điều kiện môi trường:** Robot được lên kế hoạch sử dụng trong môi trường ngoài trời, nghĩa là robot phải di chuyển trên đường và có thể vượt qua các chướng ngại vật nhỏ. Ngoài ra, thiết bị điện tử trên robot phải được bảo vệ.
5. **Độc đáo:** Để đóng góp cho nghiên cứu và phát triển khoa học, robot phải khác biệt và mới mẻ.

## 6.2.2 Cấu trúc phần cứng:

Cấu trúc phần cứng của robot được thực hiện bằng cách xem xét các yêu cầu thiết kế.

Các đường màu xanh lá cây tượng trưng cho các kết nối tín hiệu và liên lạc, trong khi các đường màu đỏ và cam là các kết nối nguồn chính và các đường màu xanh lam là các kết nối giữa động cơ và bộ điều khiển động cơ.



Bộ điều khiển và bảng điều khiển gắn trên rô-bốt được đưa ra như sau.

### 1. Mainboard (Nvidia Jetson TX1):

Mainboard là bộ não của robot. Nvidia Jetson TX1 được sử dụng để điều khiển cấp cao hơn. ROS được sử dụng để liên lạc giữa các mô-đun khác nhau. Đó là GPU Nvidia Maxwell (256 lõi CUDA) cho phép thời gian suy luận nhanh đối với các mạng nơ-ron sâu. Nó chạy tất cả các quy trình khác như thuật toán lập kế hoạch toàn cục, thuật toán lập kế hoạch cục bộ, thuật toán điều khiển, v.v. Các giai đoạn tiếp theo của công việc, phương pháp điều khiển và thuật toán tổng hợp cảm biến sẽ được triển khai trên bo mạch chính.

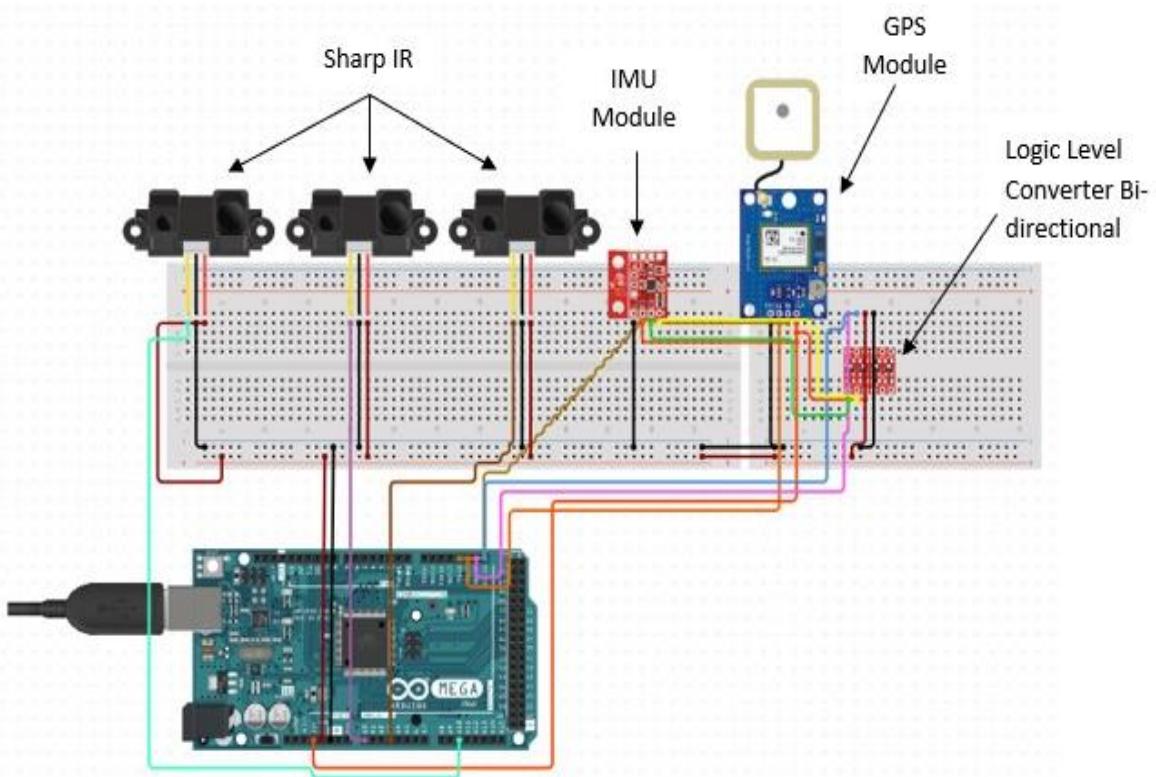
### 2. Bảng điều khiển (Arduino Mega):

Arduino Mega được sử dụng để điều khiển cấp thấp hơn. Nó điều khiển cả hai động cơ (truyền động chính và lái) và xử lý tất cả các cảm biến. Nó giao tiếp với bộ xử lý chính (Nvidia Jetson TX1) thông qua rosserial để nhận lệnh cho động cơ và xuất bản cảm biến dữ liệu.

### 3. Trình điều khiển động cơ DC (L298):

L298 là mạch nguyên khối tích hợp trong gói Multiwatt và PowerSO20 15 dây dẫn. Đây là trình điều khiển toàn cầu kép dòng điện cao, điện áp cao được thiết kế để chấp nhận các mức logic TTL tiêu chuẩn và điều khiển các tải quy nạp như relay, solenoid, DC và động cơ bước. Hai đầu vào kích hoạt được cung cấp để bật hoặc tắt thiết bị một cách độc lập với các tín hiệu đầu vào. Bộ phát của bóng bán dẫn thấp hơn của mỗi cầu nối được kết nối và thiết bị đầu cuối bên ngoài tương ứng có thể được sử dụng để kết nối điện trở cảm biến bên ngoài. Một đầu vào cung cấp bổ sung được cung cấp để logic hoạt động ở điện áp thấp hơn. Động cơ DC giảm tốc Johnsons được điều khiển bởi bảng này.

Sơ đồ mạch kết nối của các cảm biến.



Các cảm biến được sử dụng trong robot như sau:

#### 1. Orbecc Astra Depth Camera:

Camera RGBD được gắn để có được hình ảnh xem trước và bản đồ độ sâu 3D phía trước để nhận thức và lập kế hoạch. Nó có phạm vi 8 mét cho bản đồ độ sâu 3D. Hữu ích cho việc bản địa hóa (sử dụng phép đo thị giác) và để xác định các chướng ngại vật.

## 2. YDLIDAR X4:

Công cụ tìm phạm vi laser cung cấp bản đồ độ sâu 360 độ 2D (phẳng), được sử dụng để nhận thức và lập kế hoạch. Nó có phạm vi quét 10 mét. Hữu ích cho việc bản địa hóa robot và xác định chướng ngại vật.

## 3. SparkFun IMU Breakout MPU9250:

Một đơn vị đo lường quán tính (IMU). Nó bao gồm một gia tốc kế 3 trục, con quay hồi chuyển 3 trục và từ kế 3 trục. Hữu ích cho việc bản địa hóa robot.

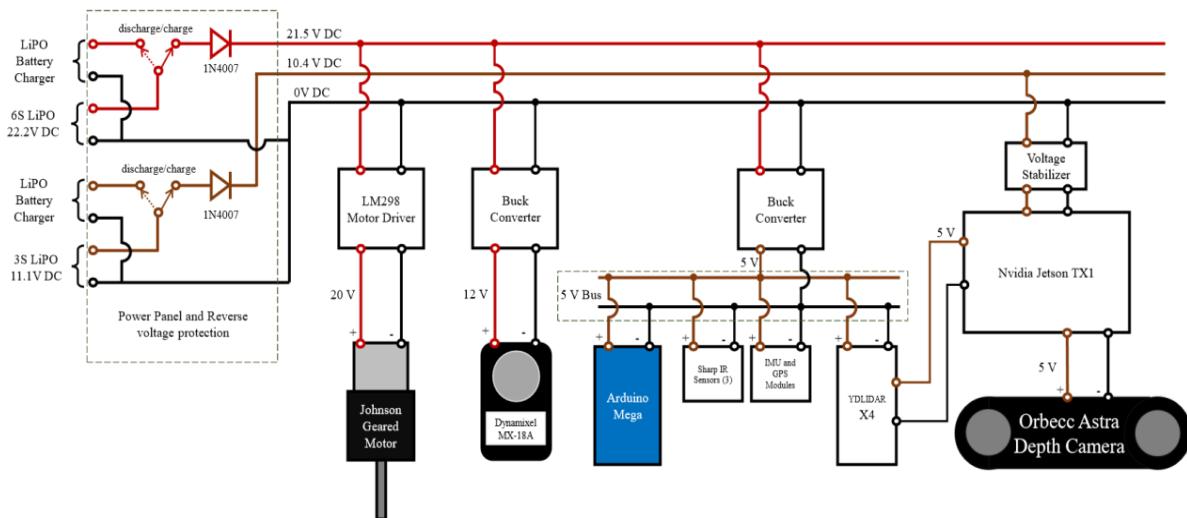
## 4. Cảm biến hồng ngoại SHARP:

Cảm biến đo khoảng cách, được sử dụng làm tuyen phòng thủ cuối cùng chống va chạm. Nó có phạm vi từ 4-30 cm.

## 5. Mô-đun GPS Neo-M8N:

Cung cấp vị trí toàn cầu (vĩ độ, kinh độ và độ cao) hữu ích cho việc lập kế hoạch toàn cầu. Mô-đun GPS Neo-M8N được sử dụng cung cấp độ nhạy điều hướng 167 dBm và hỗ trợ tất cả các hệ thống tăng cường vệ tinh.

Sơ đồ hệ thống điện của robot:



Hệ thống điện được thiết kế theo yêu cầu của hệ thống truyền động. Hệ thống điện có ba phần. Những bộ phận này là bộ pin, bảng điện và bộ sạc. Nó có hai bộ pin Lithium Polymer (LiPo) để cung cấp năng lượng cho hệ thống của nó (6s và 3s).

Pin LiPo có dung lượng cao hơn so với các loại pin khác có cùng kích thước và trọng lượng. Nhưng pin LiPo có vấn đề về an toàn. Vì lý do này, pin LiPo phải được theo dõi trong khi sạc và xả.

Bus 5V được thêm vào hệ thống để cấp nguồn cho các thiết bị phụ trợ như cảm biến và trình điều khiển động cơ. Hệ thống quản lý pin giám sát trạng thái pin bằng cách đo điện áp pin, nhiệt độ pin và dòng điện được rút ra từ bộ pin.

Bảng nguồn MCU trong bộ sạc LiPo sẽ cắt điện trong trường hợp xảy ra tình huống nguy hiểm, chẳng hạn như quá điện áp hoặc ngắn mạch, trong khi sạc hoặc xả pin. Bảng nguồn được cung cấp ở bên cạnh robot để đặt các công tắc BẬT/TẮT và chuyển đổi giữa các chế độ sạc/xả.

### 6.2.3 Thiết kế phần cứng:

Robot nặng khoảng 5kg và có khả năng chịu tải 2kg.

#### 1. **Thân máy:**

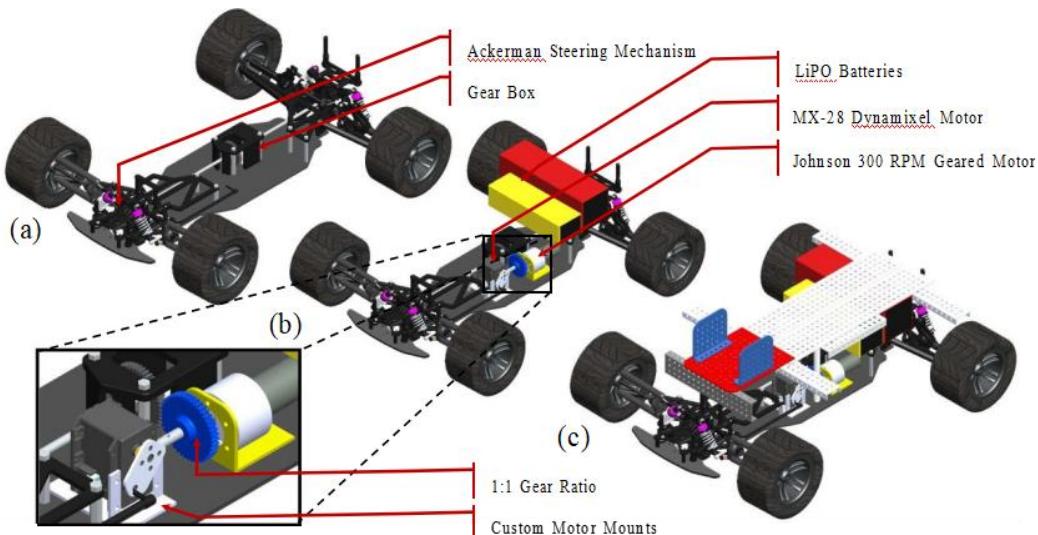
Bao gồm khung gầm robot, động cơ, pin, cảm biến nhiệt độ và bảng điều khiển. Các bộ phận ở cấp độ cơ thể là cố định và duy nhất cho nền tảng robot.

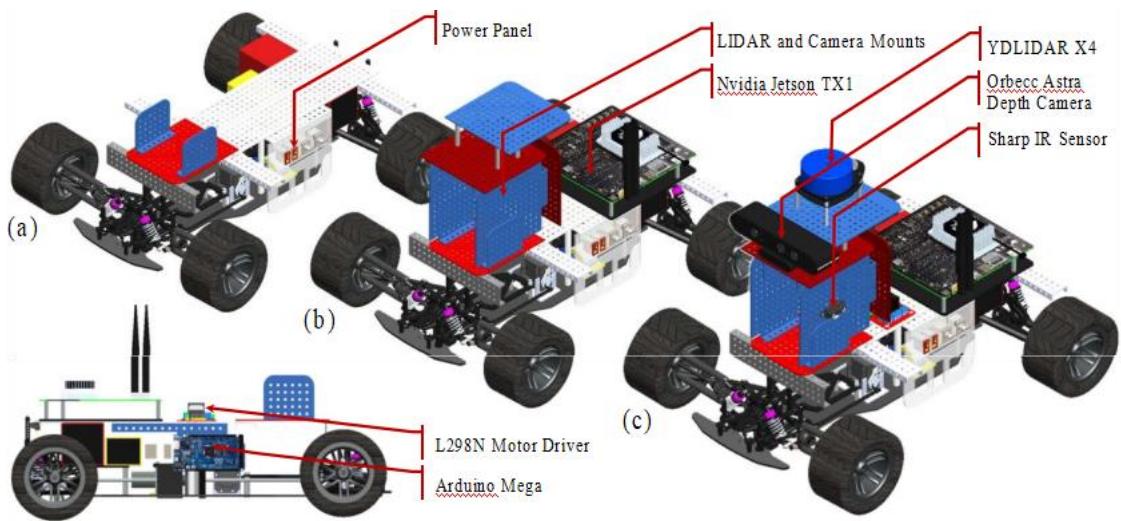
#### 2. **Điều khiển:**

Bảng nguồn, bo mạch chính, bus 5V DC, bảng điều khiển động cơ và cảm biến khoảng cách được đặt ở cấp này. Mức này có thể tháo rời để có thể thực hiện các thay đổi. Điều khiển có thể được sử dụng trên bất kỳ nền tảng nào khác, miễn là trình điều khiển động cơ và pin được lắp.

#### 3. **Tầng mái:**

Tầng này được thiết kế cho cảm biến quang học. Máy ảnh và LiDAR được đặt ở cấp độ này để đảm bảo không có tắc nghẽn trong phạm vi. Trong các giai đoạn sau của công việc này, các thành phần điều hướng bổ sung và thiết bị dành riêng cho ứng dụng có thể được thêm vào cấp độ này.



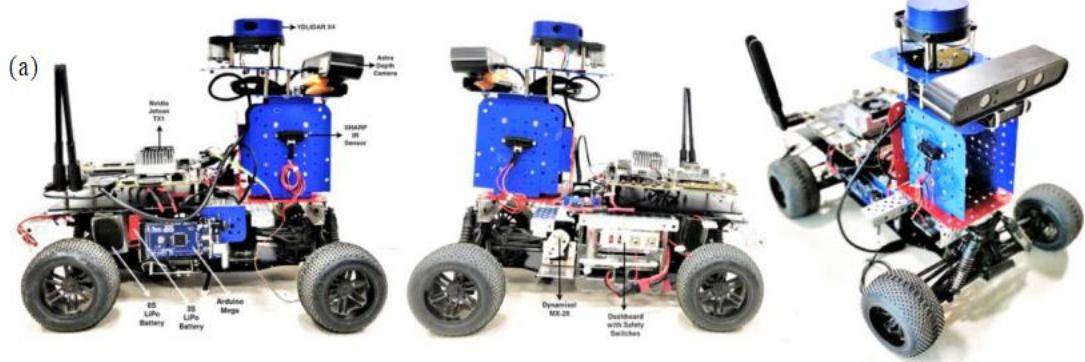


#### 6.2.4 Mô tả sản phẩm

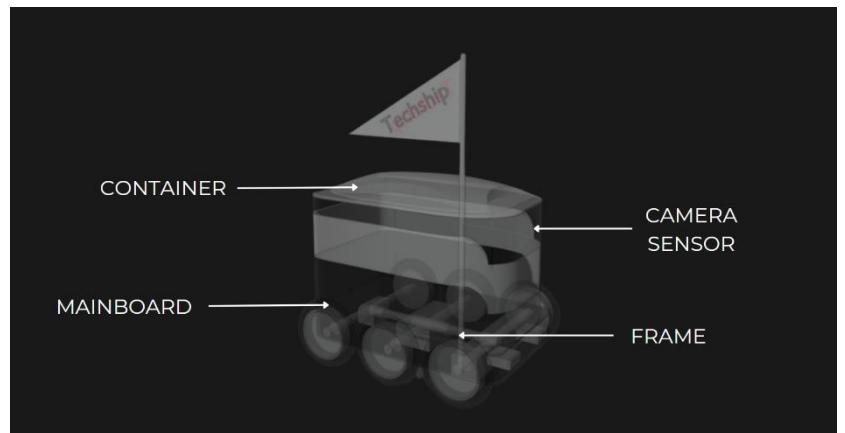
Mô phỏng sản phẩm:



Mô tả sản phẩm thực tế:

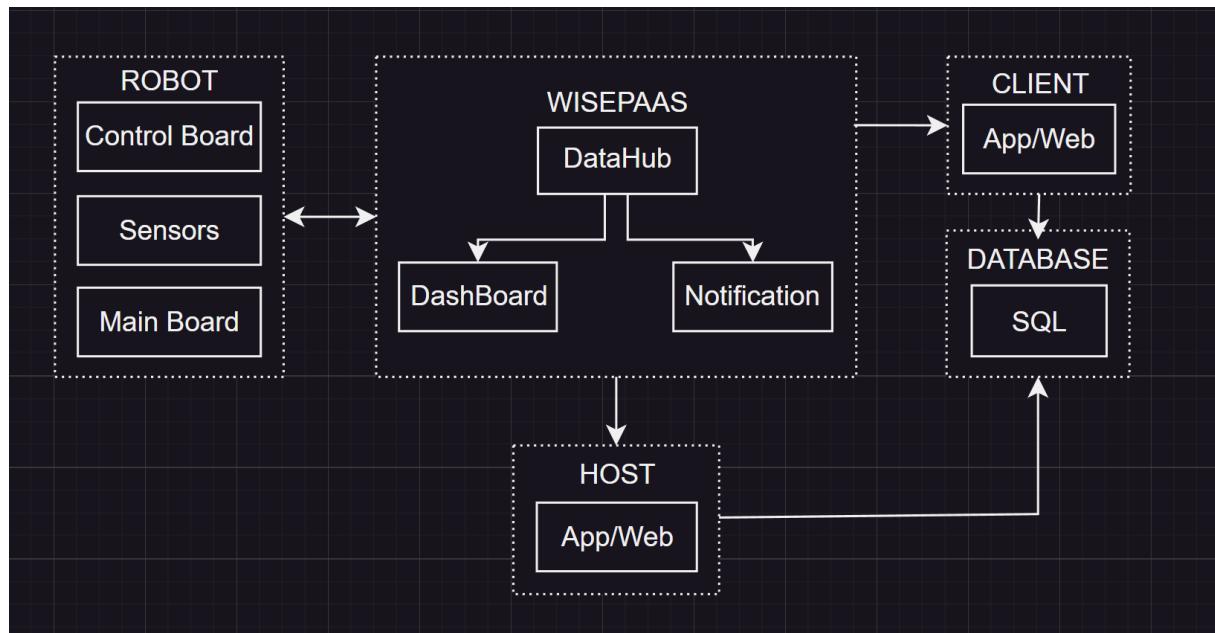


Mô phỏng sản phẩm hoàn thiện 3D:



## 6.3 WISE-PaaS

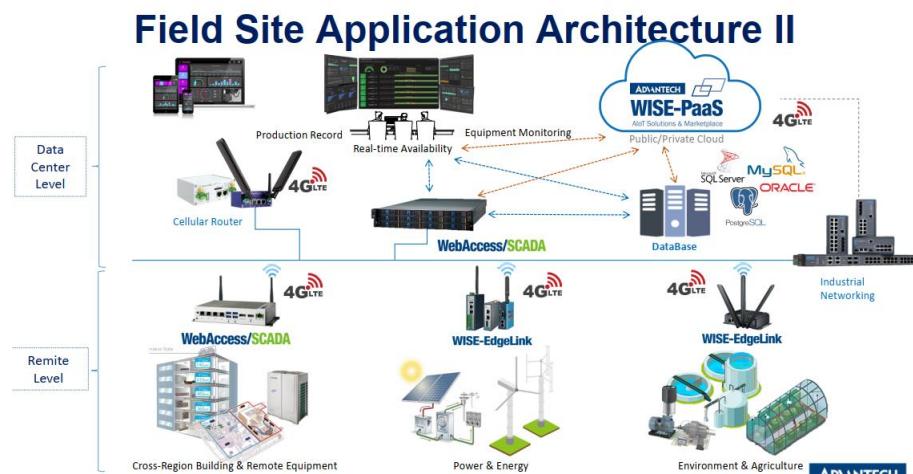
Sơ đồ nguyên lý:



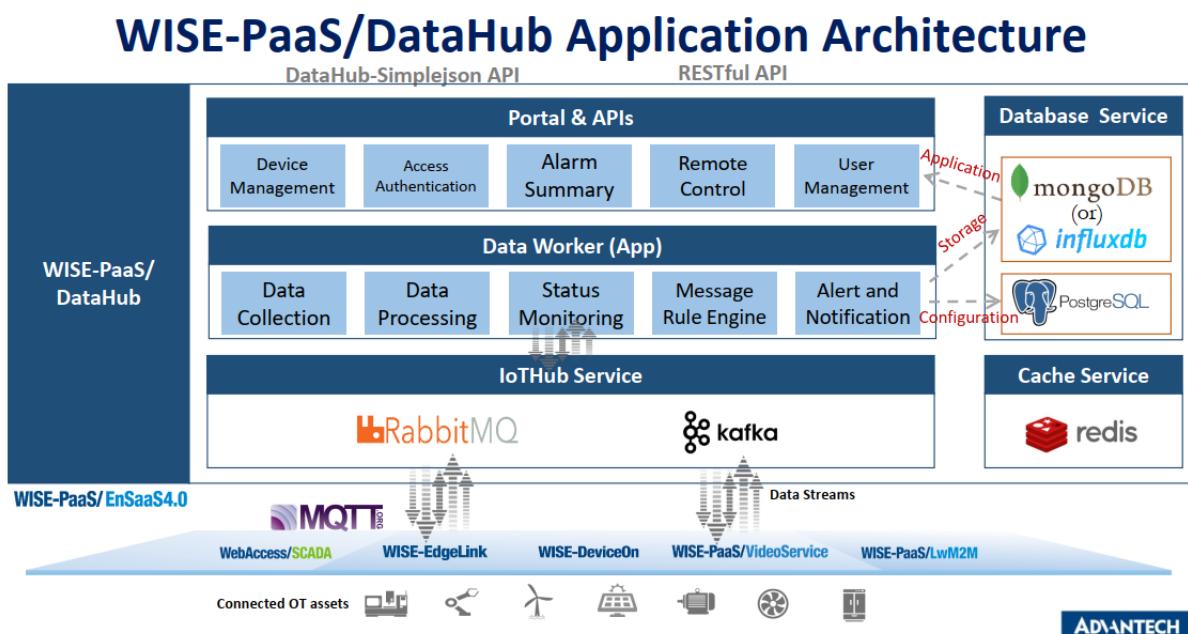
DataHub sẽ là ứng dụng nhận trực tiếp dữ liệu được tải lên từ Robot.

Để sử dụng DashBoard và Notification phải kết nối với DataHub để lấy dữ liệu từ đó vẽ đồ thị, hiển thị dữ liệu, thông báo,....

Một số thông tin từ Advantech về Kiến trúc WISE-PaaS:



Một số thông tin từ Advantech về DataHub:



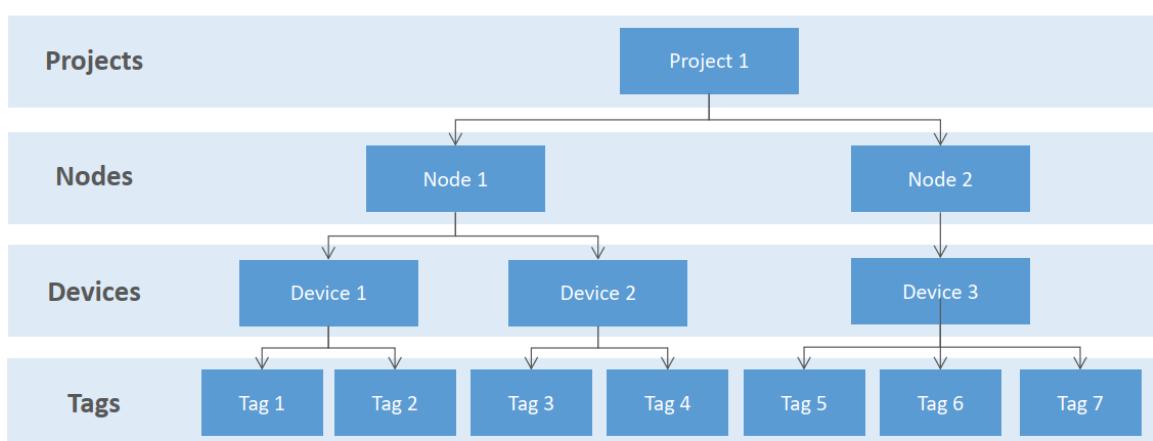
## WISE-PaaS/DataHub Portal

The screenshot shows the WISE-PaaS/DataHub Portal's Project List interface:

- Left Sidebar:**
  - DataHub
  - Device Management
  - Alarm
  - Account Setting
  - Retention Policy
  - WISE-PaaS Dashboard
  - Notification Service
  - User Guide
  - API Document
- Header:** tony.fu@advantech.com.tw
- Project List Table:**

Project Name	Description	Detail	Delete
00_Advantech_Demo	III 培訓公用項目，請勿操作此Project !!!	...	trash bin >
Advantech		...	trash bin >
asg_Tony_Fu		...	trash bin >
Datahub_test	Datahub教學測試	...	trash bin >
fdadff	asdf	...	trash bin >
HansLee		...	trash bin >
sdk_0305_lixiang_xu		...	trash bin >
sdk_1215_SK_Chen		...	trash bin >
- Bottom Right:** ADVANTECH

## DataHub Layer Structure



ADVANTECH

## Kết nối DataHub với DashBoard:

### Connect to WISE-PaaS/Dashboard

The screenshot shows the DataHub application's sidebar with various options like Device Management, Alarm, and User Guide. The 'WISE-PaaS Dashboard' option is highlighted with a red box and a hand cursor icon. A blue arrow points from this sidebar to a larger screenshot of the WISE-PaaS Dashboard. The dashboard displays a table of data, several chart types (Grouped Bar Chart, Singlestat, Graph, Pie Chart), and a navigation bar at the top.

ADVANTECH

## Kết nối DataHub với Notification Portal:

### Connect to Notification Portal

The screenshot shows the DataHub application's sidebar with various options. The 'Notification Service' option is highlighted with a red box and a hand cursor icon. A blue arrow points from this sidebar to a larger screenshot of the 'Notification' section in the WISE-PaaS portal. This section shows a table of notification settings with columns for Group Name, Channel, Level, Description, Member, and Duplicate. The table lists several entries such as 'line\_test' (LINE, Critical), 'LineTony' (LINE, Critical), etc.

ADVANTECH

## Kết nối DataHub với DataHub User Guide:

### Connect to Data Hub User Guide

The screenshot shows the DataHub application's sidebar with various options. The 'User Guide' option is highlighted with a red box and a hand cursor icon. A blue arrow points from this sidebar to a larger screenshot of the WISE-PaaS Technical Documentation page. The page title is '什麼是 DataHub'. It features a diagram illustrating the DataHub architecture, showing a central 'DataHub' node connected to various industrial devices (represented by icons like a factory, car, and house) via a network. Below the diagram, there is explanatory text about DataHub's functions and its role in the WISE-PaaS ecosystem.

URL: [https://docs.wise-paas.advantech.com/zh-tw/Guides\\_and\\_API\\_References/Data\\_Acquisition/DataHub/1599029545716843927/v1.0.0](https://docs.wise-paas.advantech.com/zh-tw/Guides_and_API_References/Data_Acquisition/DataHub/1599029545716843927/v1.0.0)

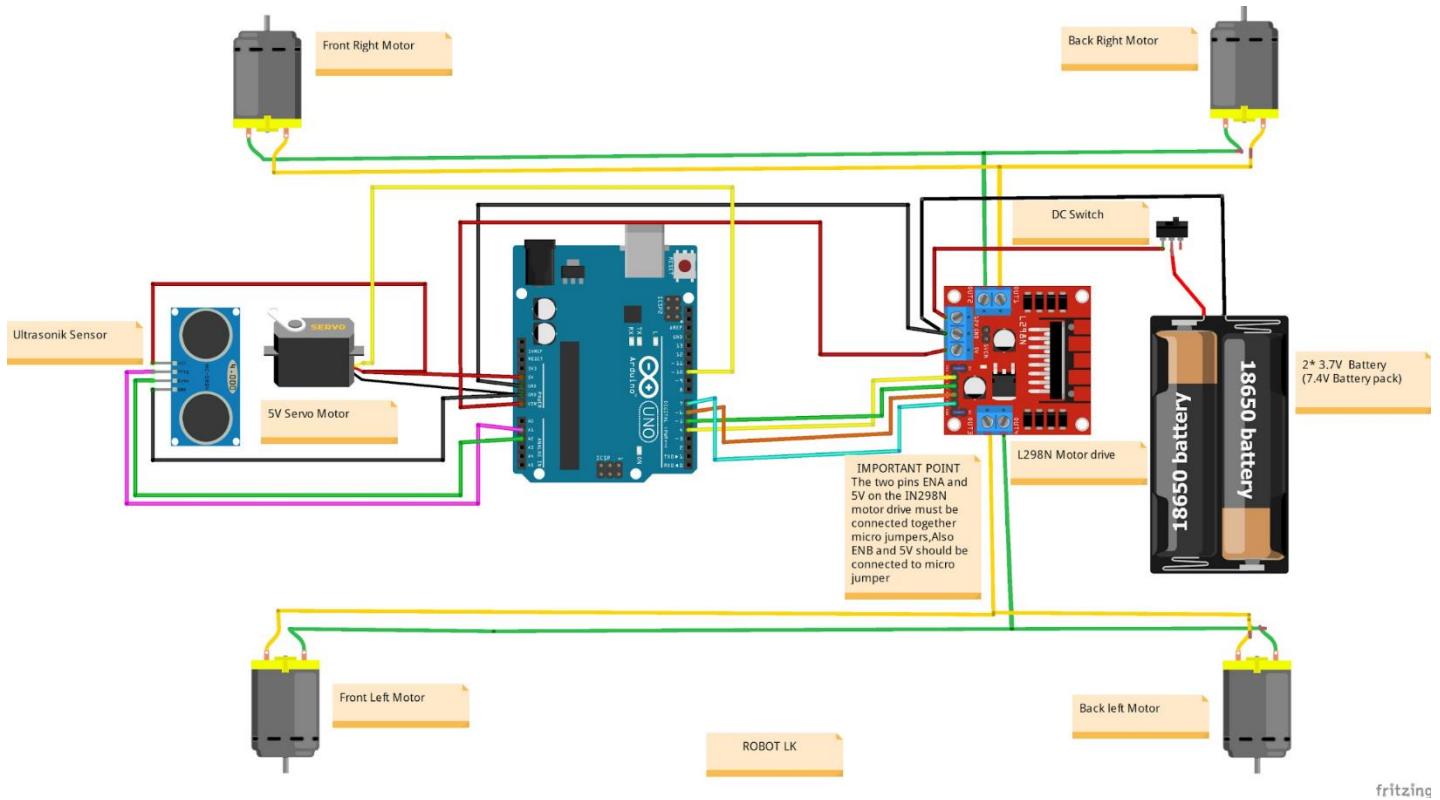
ADVANTECH

# PHẦN VII: KẾT QUẢ VÀ PHÂN TÍCH

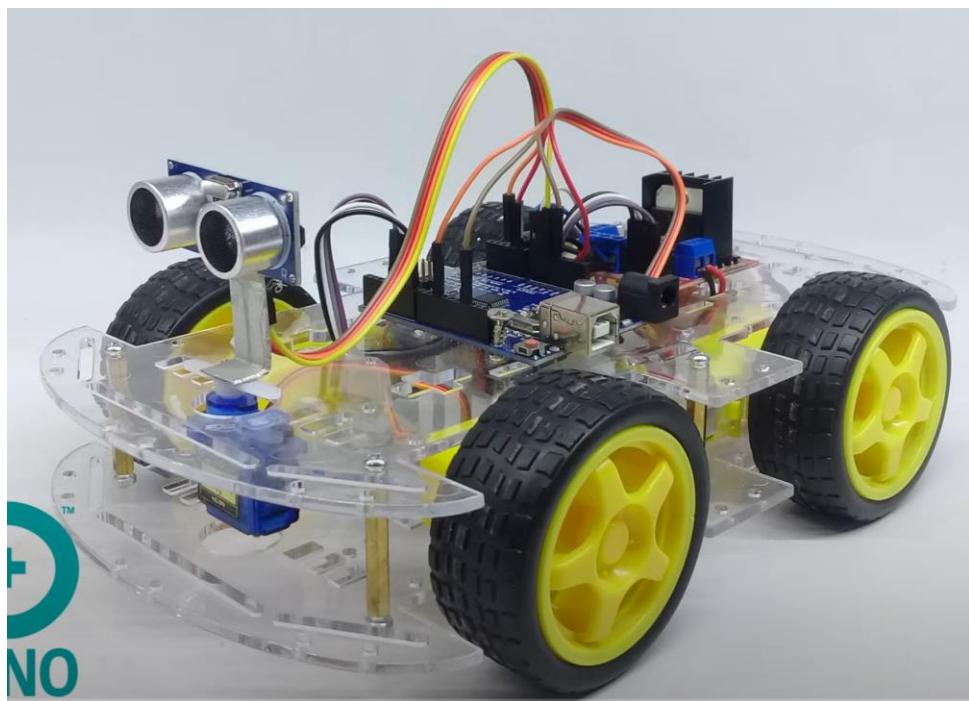
## 7.1 Obstacle avoiding Robot

Mô hình xe tự hành tránh vật cản sử dụng Adruino

### Sơ đồ châñ:



### Mô phỏng xe:



## Nội dung Code:

```
#include <Servo.h>
#include <NewPing.h>

const int LeftMotorForward = 7;
const int LeftMotorBackward = 6;
const int RightMotorForward = 5;
const int RightMotorBackward = 4;

#define trig_pin A1 //analog input 1
#define echo_pin A2 //analog input 2

#define maximum_distance 200
boolean goesForward = false;
int distance = 100;

NewPing sonar(trig_pin, echo_pin, maximum_distance);
Servo servo_motor;

void setup(){

    pinMode(RightMotorForward, OUTPUT);
    pinMode(LeftMotorForward, OUTPUT);
    pinMode(LeftMotorBackward, OUTPUT);
    pinMode(RightMotorBackward, OUTPUT);

    servo_motor.attach(10); //our servo pin

    servo_motor.write(115);
    delay(2000);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
    distance = readPing();
    delay(100);
}

void loop(){

    int distanceRight = 0;
    int distanceLeft = 0;
    delay(50);

    if (distance <= 35){
        moveStop();
        delay(300);
        moveBackward();
        delay(400);
        moveStop();
        delay(300);
    }
}
```

```

distanceRight = lookRight();
delay(300);
distanceLeft = lookLeft();
delay(300);

if (distance >= distanceLeft){
    turnRight();
    moveStop();
}
else{
    turnLeft();
    moveStop();
}
else{
    moveForward();
}
distance = readPing();
}

int lookRight(){
servo_motor.write(50);
delay(500);
int distance = readPing();
delay(100);
servo_motor.write(115);
return distance;
}

int lookLeft(){
servo_motor.write(170);
delay(500);
int distance = readPing();
delay(100);
servo_motor.write(115);
return distance;
}
delay(100);

int readPing(){
delay(70);
int cm = sonar.ping_cm();
if (cm==0){
    cm=250;
}
return cm;
}

void moveStop(){

digitalWrite(RightMotorForward, LOW);
digitalWrite(LeftMotorForward, LOW);
digitalWrite(RightMotorBackward, LOW);
digitalWrite(LeftMotorBackward, LOW);
}

```

```
void moveForward(){

    if(!goesForward){

        goesForward=true;

        digitalWrite(LeftMotorForward, HIGH);
        digitalWrite(RightMotorForward, HIGH);

        digitalWrite(LeftMotorBackward, LOW);
        digitalWrite(RightMotorBackward, LOW);
    }
}

void moveBackward(){

    goesForward=false;

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorForward, LOW);

}

void turnRight(){

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorBackward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorForward, LOW);

    delay(250);

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);

}

void turnLeft(){

    digitalWrite(LeftMotorBackward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorBackward, LOW);

    delay(250);
}
```

```

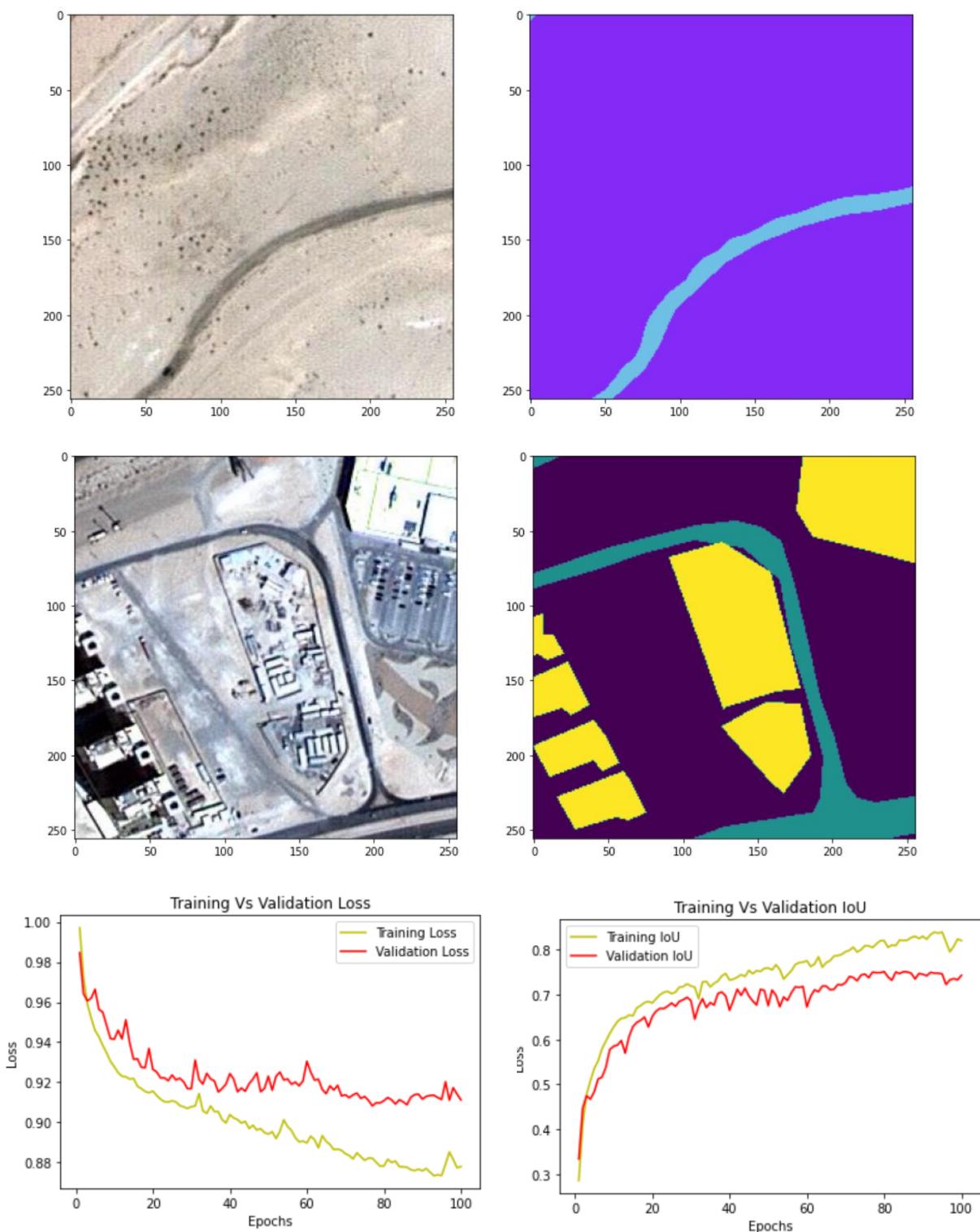
digitalWrite(LeftMotorForward, HIGH);
digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);
digitalWrite(RightMotorBackward, LOW);
}

```

## 7.2 Semantic Segmentation of Aerial imagery using U-net

Hình ảnh kết quả:



## Nội dung Code:

```
import os
import cv2
import numpy as np

from matplotlib import pyplot as plt
from patchify import patchify
from PIL import Image

import segmentation_models as sm
from tensorflow.keras.metrics import MeanIoU

from sklearn.preprocessing import MinMaxScaler, StandardScaler

scaler = MinMaxScaler()

from keras.models import Model
from keras.layers import Input, Conv2D, MaxPooling2D, UpSampling2D,
concatenate, Conv2DTranspose, BatchNormalization, \
    Dropout, Lambda
from keras import backend as K

def jacard_coef(y_true, y_pred):
    y_true_f = K.flatten(y_true)
    y_pred_f = K.flatten(y_pred)
    intersection = K.sum(y_true_f * y_pred_f)
    return (intersection + 1.0) / (K.sum(y_true_f) + K.sum(y_pred_f) - intersection + 1.0)

def multi_unet_model(n_classes=4, IMG_HEIGHT=256, IMG_WIDTH=256,
IMG_CHANNELS=1):
    s = inputs

    c1 = Conv2D(16, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same')(s)
    c1 = Dropout(0.2)(c1) # Original 0.1
    c1 = Conv2D(16, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same')(c1)
    p1 = MaxPooling2D((2, 2))(c1)

    c2 = Conv2D(32, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same')(p1)
    c2 = Dropout(0.2)(c2) # Original 0.1
    c2 = Conv2D(32, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same')(c2)
    p2 = MaxPooling2D((2, 2))(c2)

    c3 = Conv2D(64, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same')(p2)
    c3 = Dropout(0.2)(c3)
    c3 = Conv2D(64, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same')(c3)
    p3 = MaxPooling2D((2, 2))(c3)

    c4 = Conv2D(128, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same')(p3)
    c4 = Dropout(0.2)(c4)
    c4 = Conv2D(128, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same')(c4)
    p4 = MaxPooling2D(pool_size=(2, 2))(c4)

    c5 = Conv2D(256, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same')(p4)
    c5 = Dropout(0.3)(c5)
    c5 = Conv2D(256, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same')(c5)
```

```

        u6 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same') (c5)
        u6 = concatenate([u6, c4])
        c6 = Conv2D(128, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same') (u6)
        c6 = Dropout(0.2) (c6)
        c6 = Conv2D(128, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same') (c6)

        u7 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same') (c6)
        u7 = concatenate([u7, c3])
        c7 = Conv2D(64, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same') (u7)
        c7 = Dropout(0.2) (c7)
        c7 = Conv2D(64, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same') (c7)

        u8 = Conv2DTranspose(32, (2, 2), strides=(2, 2), padding='same') (c7)
        u8 = concatenate([u8, c2])
        c8 = Conv2D(32, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same') (u8)
        c8 = Dropout(0.2) (c8) # Original 0.1
        c8 = Conv2D(32, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same') (c8)

        u9 = Conv2DTranspose(16, (2, 2), strides=(2, 2), padding='same') (c8)
        u9 = concatenate([u9, c1], axis=3)
        c9 = Conv2D(16, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same') (u9)
        c9 = Dropout(0.2) (c9) # Original 0.1
        c9 = Conv2D(16, (3, 3), activation='relu',
kernel_initializer='he_normal', padding='same') (c9)

outputs = Conv2D(n_classes, (1, 1), activation='softmax') (c9)
model = Model(inputs=[inputs], outputs=[outputs])
return model

root_directory = 'C:/Users/Xii/Downloads/archive/Semantic segmentation dataset/'

patch_size = 256

image_dataset = []
for path, subdirs, files in os.walk(root_directory):
    print(path)
    dirname = path.split(os.path.sep)[-1]
    if dirname == 'images':
        images = os.listdir(path)
        for i, image_name in enumerate(images):
            if image_name.endswith(".jpg"):

                image = cv2.imread(path + "/" + image_name, 1)
                SIZE_X = (image.shape[1] // patch_size) * patch_size
                SIZE_Y = (image.shape[0] // patch_size) * patch_size
                image = Image.fromarray(image)
                image = image.crop((0, 0, SIZE_X, SIZE_Y))
                image = np.array(image)

                print("Now patchifying image:", path + "/" + image_name)
                patches_img = patchify(image, (patch_size, patch_size, 3),
                                      step=patch_size)

                for i in range(patches_img.shape[0]):
                    for j in range(patches_img.shape[1]):
                        single_patch_img = patches_img[i, j, :, :]

                        single_patch_img = scaler.fit_transform(
                            single_patch_img.reshape(-1,
single_patch_img.shape[-1])).reshape(single_patch_img.shape)

```

```

        single_patch_img = single_patch_img[
            0]
        image_dataset.append(single_patch_img)

mask_dataset = []
for path, subdirs, files in os.walk(root_directory):
    dirname = path.split(os.path.sep)[-1]
    if dirname == 'masks':
        masks = os.listdir(path)
        for i, mask_name in enumerate(masks):
            if mask_name.endswith(".png"):

                mask = cv2.imread(path + "/" + mask_name,
                                   1)
                mask = cv2.cvtColor(mask, cv2.COLOR_BGR2RGB)
                SIZE_X = (mask.shape[1] // patch_size) * patch_size
                SIZE_Y = (mask.shape[0] // patch_size) * patch_size
                mask = Image.fromarray(mask)
                mask = mask.crop((0, 0, SIZE_X, SIZE_Y))
                mask = np.array(mask)

                print("Now patchifying mask:", path + "/" + mask_name)
                patches_mask = patchify(mask, (patch_size, patch_size, 3),
                                         step=patch_size)

                for i in range(patches_mask.shape[0]):
                    for j in range(patches_mask.shape[1]):
                        single_patch_mask = patches_mask[i, j, :, :]
                        single_patch_mask = single_patch_mask[
                            0]
                        mask_dataset.append(single_patch_mask)

image_dataset = np.array(image_dataset)
mask_dataset = np.array(mask_dataset)

import random
import numpy as np

image_number = random.randint(0, len(image_dataset))
plt.figure(figsize=(12, 6))
plt.subplot(121)
plt.imshow(np.reshape(image_dataset[image_number], (patch_size, patch_size,
3)))
plt.subplot(122)
plt.imshow(np.reshape(mask_dataset[image_number], (patch_size, patch_size,
3)))
plt.show()

a = int('3C', 16)
print(a)
Building = '#3C1098'.lstrip('#')
Building = np.array(tuple(int(Building[i:i + 2], 16) for i in (0, 2, 4)))

Land = '#8429F6'.lstrip('#')
Land = np.array(tuple(int(Land[i:i + 2], 16) for i in (0, 2, 4)))
Road = '#6EC1E4'.lstrip('#')
Road = np.array(tuple(int(Road[i:i + 2], 16) for i in (0, 2, 4)))
Vegetation = 'FEDD3A'.lstrip('#')
Vegetation = np.array(tuple(int(Vegetation[i:i + 2], 16) for i in (0, 2, 4)))
Water = 'E2A929'.lstrip('#')
Water = np.array(tuple(int(Water[i:i + 2], 16) for i in (0, 2, 4)))
Unlabeled = '#9B9B9B'.lstrip('#')
Unlabeled = np.array(tuple(int(Unlabeled[i:i + 2], 16) for i in (0, 2, 4)))
label = single_patch_mask

def rgb_to_2D_label(label):
    label_seg = np.zeros(label.shape, dtype=np.uint8)
    label_seg[np.all(label == Building, axis=-1)] = 0

```

```

label_seg[np.all(label == Land, axis=-1)] = 1
label_seg[np.all(label == Road, axis=-1)] = 2
label_seg[np.all(label == Vegetation, axis=-1)] = 3
label_seg[np.all(label == Water, axis=-1)] = 4
label_seg[np.all(label == Unlabeled, axis=-1)] = 5

label_seg = label_seg[:, :, 0]
return label_seg

labels = []
for i in range(mask_dataset.shape[0]):
    label = rgb_to_2D_label(mask_dataset[i])
    labels.append(label)

labels = np.array(labels)
labels = np.expand_dims(labels, axis=3)

print("Unique labels in label dataset are: ", np.unique(labels))

import random
import numpy as np

image_number = random.randint(0, len(image_dataset))
plt.figure(figsize=(12, 6))
plt.subplot(121)
plt.imshow(image_dataset[image_number])
plt.subplot(122)
plt.imshow(labels[image_number][:, :, 0])
plt.show()

n_classes = len(np.unique(labels))
from keras.utils import to_categorical

labels_cat = to_categorical(labels, num_classes=n_classes)

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(image_dataset,
labels_cat, test_size=0.20, random_state=42)

weights = [0.1666, 0.1666, 0.1666, 0.1666, 0.1666, 0.1666]
dice_loss = sm.losses.DiceLoss(class_weights=weights)
focal_loss = sm.losses.CategoricalFocalLoss()
total_loss = dice_loss + (1 * focal_loss)  #

IMG_HEIGHT = X_train.shape[1]
IMG_WIDTH = X_train.shape[2]
IMG_CHANNELS = X_train.shape[3]

metrics = ['accuracy', jacard_coef]

def get_model():
    return multi_unet_model(n_classes=n_classes, IMG_HEIGHT=IMG_HEIGHT,
IMG_WIDTH=IMG_WIDTH, IMG_CHANNELS=IMG_CHANNELS)

model = get_model()
model.compile(optimizer='adam', loss=total_loss, metrics=metrics)
model.summary()

history1 = model.fit(X_train, y_train,
batch_size=16,
verbose=1,
epochs=2,
validation_data=(X_test, y_test),
shuffle=False)
BACKBONE = 'resnet34'
preprocess_input = sm.get_preprocessing(BACKBONE)

```

```

X_train_prep = preprocess_input(X_train)
X_test_prep = preprocess_input(X_test)

model_resnet_backbone = sm.Unet(BACKBONE, encoder_weights='imagenet',
classes=n_classes, activation='softmax')

model_resnet_backbone.compile(optimizer='adam',
loss='categorical_crossentropy', metrics=metrics)

print(model_resnet_backbone.summary())

history2 = model_resnet_backbone.fit(X_train_prep,
                                      y_train,
                                      batch_size=16,
                                      epochs=2,
                                      verbose=1,
                                      validation_data=(X_test_prep,
y_test))

history = history1
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(loss) + 1)
plt.plot(epochs, loss, 'y', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

acc = history.history['jaccard_coef']
val_acc = history.history['val_jaccard_coef']

plt.plot(epochs, acc, 'y', label='Training IoU')
plt.plot(epochs, val_acc, 'r', label='Validation IoU')
plt.title('Training and validation IoU')
plt.xlabel('Epochs')
plt.ylabel('IoU')
plt.legend()
plt.show()

import keras
from IPython.display import clear_output

%matplotlib
inline

class PlotLoss(keras.callbacks.Callback):
    def on_train_begin(self, logs={}):
        self.i = 0
        self.x = []
        self.losses = []
        self.val_losses = []

        self.jaccard_coef = []
        self.val_jaccard_coef = []

        self.fig = plt.figure()
        self.logs = []

    def on_epoch_end(self, epoch, logs={}):
        self.logs.append(logs)
        self.x.append(self.i)
        self.losses.append(logs.get('loss'))
        self.val_losses.append(logs.get('val_loss'))

        self.jaccard_coef.append(logs.get('jaccard_coef'))
```



```

        callbacks=[plot_loss],
        shuffle=False)

from keras.models import load_model

model = load_model("models/satellite_standard_unet_100epochs.hdf5",
                    custom_objects={'dice_loss_plus_2focal_loss':
total_loss,
                           'jaccard_coef': jaccard_coef})

y_pred = model.predict(X_test)
y_pred_argmax = np.argmax(y_pred, axis=3)
y_test_argmax = np.argmax(y_test, axis=3)

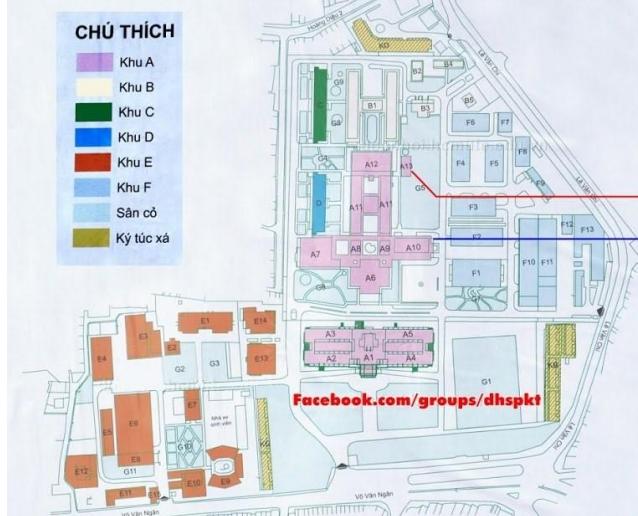
from keras.metrics import MeanIoU

n_classes = 6
IOU_keras = MeanIoU(num_classes=n_classes)
IOU_keras.update_state(y_test_argmax, y_pred_argmax)
print("Mean IoU =", IOU_keras.result().numpy())

```

### 7.3 Dijkstra Algorithms

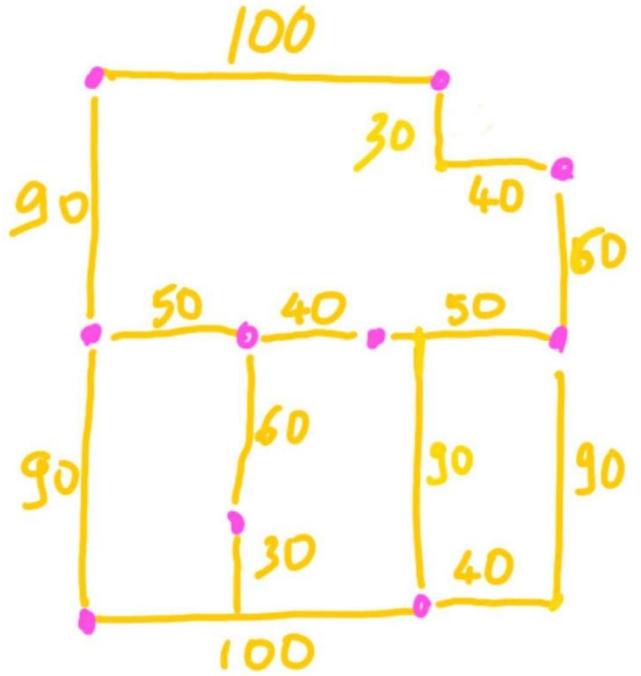
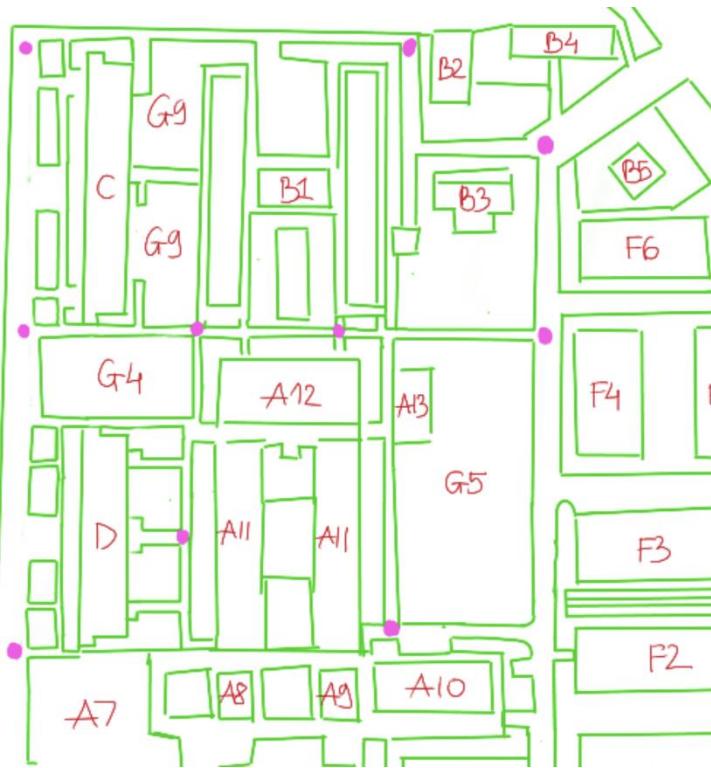
#### Bản đồ trường Đại học Sư Phạm Kỹ Thuật TP.HCM



#### Kết quả tìm đường đi ngắn nhất:

Start	Shortest Path	Destination	Cost
D	D - C - A	A	140
D	D - E - G - F - B	B	220
D	D - C	C	50
D	D - E - G - F	E	150
D	D - E	F	40
D	D - G	G	90
D	D - C - H	H	140
D	D - I	I	60
D	D - E - J	J	140

## Xác định các node:

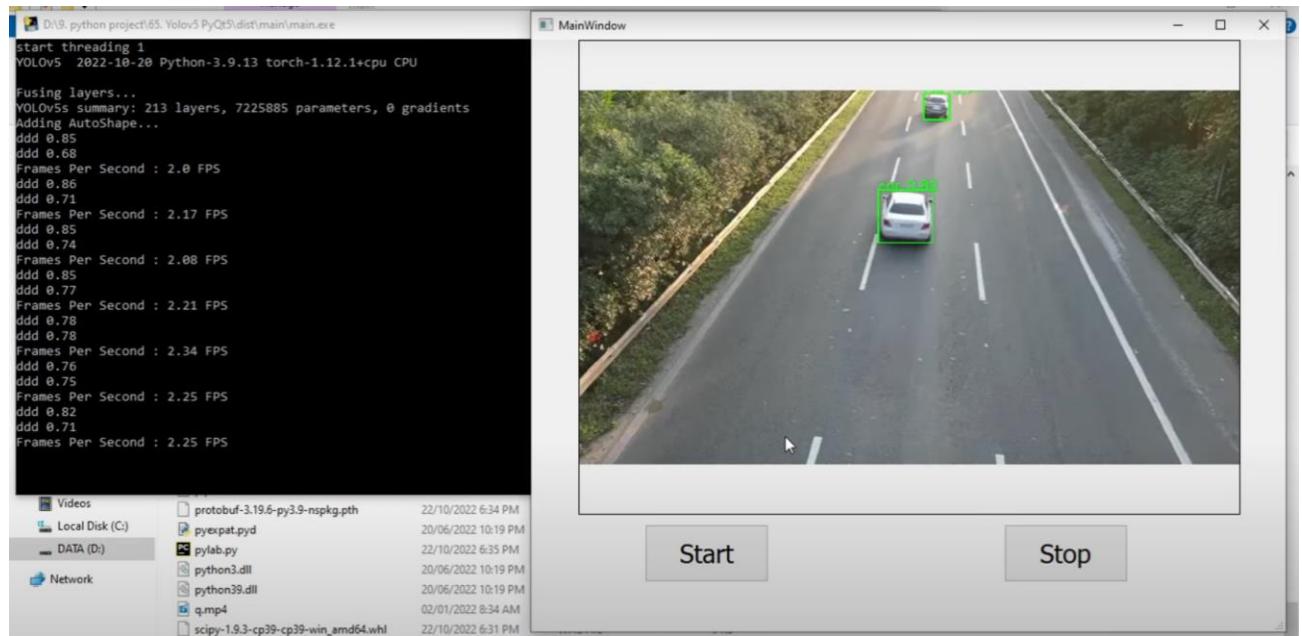


## 7.4 Obstacle Detection

Văn đang trong quá trình tìm hiểu.

Kết quả dự kiến:

YOLOV5 + PyQt5 => file.exe



## 7.5 DataHub

Hình ảnh kết quả:

The screenshot displays the DataHub application interface across four main sections: Project List, Node List, Device List, and a central configuration area.

**Project List:** Shows a list of projects with columns for Project Name, Description, Detail, and Delete. Projects listed include ATW-Jacey, demo, Demo\_TS, SANNIN, STMS, and XLNT.

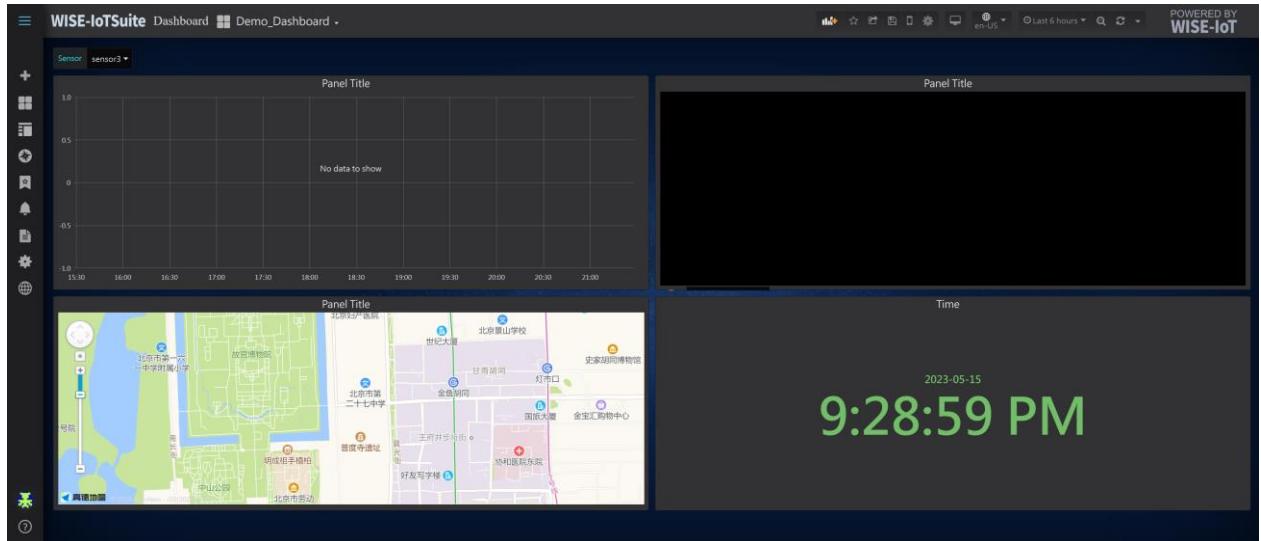
**Node List:** Shows a list of nodes under the demo project with columns for Node Name, Description, Status, Detail, Move to, Lock Config, and Delete. One node, data\_demo, is listed.

**Device List:** Shows a list of devices with columns for Device Name, Device Type, Description, Status, Detail, and Delete. One device, Device1, is listed.

**SDK Test:** A configuration panel for DCCS and MQTT. It includes fields for API Url (paas.com/) and Credential Key (7062f537w). It also shows NodId: 2d1175975, DTag1 Value: 1, and Device Count: 1. The panel features a "Connected" status indicator and various control buttons: Connect, Disconnect, Update Device Status, Send Data, Delete All Config, Delete Devices, and Delete Tag.

## 7.6 DashBoard

Hình ảnh kết quả:



## 7.7 Notification

Hình ảnh kết quả:

The notification settings page shows a group named 'Demo\_techship' configured to send notifications via LINE. Below is a sample message from 'LINE Notify':

【BOT-0421】 Hello, this is the test message from WISE PaaS Notification Service.  
Current Time:2020-05-26T09:29:33.196Z  
Powered by WISE-PaaS

下方为尚未閱讀的訊息

輸入訊息

## **PHẦN VIII: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN SẢN PHẨM TRONG TƯƠNG LAI**

### **8.1 Kết luận**

Báo cáo đã xây dựng, thiết kế được về phần cứng cũng như nêu ra được các thuật toán, kỹ thuật sử dụng. Các thuật toán, kỹ thuật cũng đã được chạy thử và triển khai trên các mô – đun. Các vấn đề gặp phải trong thiết kế và vận hành robot trong môi trường ngoài trời cũng đã được đề cập đến.

Mục tiêu của chúng tôi là tạo ra một hệ thống robot hoàn toàn tự động, đủ mạnh để vận hành trơn tru trên các con đường với quy hoạch và địa phương hóa phù hợp. Điều này sẽ giải quyết vấn đề robot được điều khiển thủ công bởi con người và giúp thực hiện các nhiệm vụ cho con người để giúp cuộc sống của họ dễ dàng hơn.

Mặc dù mục đích của robot trong đồ án này hướng tới robot giao hàng, nhưng vẫn có thể được sử dụng cho bất kỳ robot nào hoạt động tự động trong môi trường ngoài trời.

Ví dụ: hệ thống được trình bày cũng có thể đóng một vai trò quan trọng trong các ứng dụng hỗ trợ y tế, chẳng hạn như trong tình hình đại dịch COVID-19, phân phối thực phẩm và thuốc tự động cũng như lấy mẫu thu gom tại các khu vực xung yếu.

### **8.2 Định hướng phát triển sản phẩm trong tương lai**

Hiện tại tập trung khá nhiều vào triển khai lý thuyết hơn là chức năng thực tế của phương tiện giao hàng. Do đó, trong tương lai, phần cứng sẽ tập trung nhiều hơn vào các khía cạnh của phương tiện giao hàng như giao diện người-máy hiệu quả cũng như bảng điều khiển, hệ thống giám sát từ xa và quản lý an toàn khi có sự cố, thời gian chạy dài và khả năng tải trọng lớn hơn.

Cũng như nghiên cứu các tình huống rất phức tạp để có thể tránh trong quá trình thử nghiệm như môi trường ngoài trời rất khó dự đoán do các chướng ngại vật động như người đi bộ, động vật, xe cộ,...

Robot có thể được chế tạo mạnh mẽ hơn để hoạt động trong các điều kiện thời tiết khác nhau như mưa, tuyết,.. và trên địa hình đường không bằng phẳng.

Công việc trong tương lai có thể bao gồm phát triển các thuật toán hiệu quả hoặc chọn các cảm biến thích hợp để giải quyết các vấn đề gặp phải khi bản địa hóa rô-bốt trong môi trường ngoài trời như nghiên cứu các phần cứng thay thế cho Lidar, IR khắc phục độ nhiễu sóng khi hoạt động ngoài trời. Cũng như thay thế cho GPS khi sai số quá cao khiến cho khoảng cách gần không thể xác định một cách chính xác được.

## PHẦN IX: NHẬN XÉT CỦA GIÁO VIÊN BỘ MÔN VÀ TIẾN HÀNH VĂN ĐÁP VỀ ĐỀ TÀI

# TÀI LIỆU THAM KHẢO

- Tài liệu về lắp đặt và code

<https://youtu.be/sYWVR2QkZ1Y>

<https://youtu.be/etVX0XkBX4o>

<https://youtu.be/jvZm8REF2KY>

[https://github.com/prodramp/DeepWorks/blob/main/DL-SatelliteImagery/Satellite\\_Imagery\\_DeepLearning-Base.ipynb](https://github.com/prodramp/DeepWorks/blob/main/DL-SatelliteImagery/Satellite_Imagery_DeepLearning-Base.ipynb)

[https://github.com/bnsreenu/python\\_for\\_microscopists/tree/master/228\\_semantic\\_segmentation\\_of\\_aerial\\_imagery\\_using\\_unet](https://github.com/bnsreenu/python_for_microscopists/tree/master/228_semantic_segmentation_of_aerial_imagery_using_unet)

<https://github.com/srihari-humbarwadi/YOLOv1-TensorFlow2.0>

[https://www.researchgate.net/publication/350104659\\_Design\\_and\\_Development\\_of\\_Autonomous\\_Delivery\\_Robot](https://www.researchgate.net/publication/350104659_Design_and_Development_of_Autonomous_Delivery_Robot)

- Tài liệu tham khảo về phần cứng

<https://www.thegioiic.com/>

<https://nshopvn.com/>

- Tài liệu tham khảo khác

[https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things)

<https://tek4.vn/tong-quan-ve-internet-of-things-iots>

<https://www.advantech.com/>