**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HCM**



**GIẢI PHÁP CHO ĐỀ TÀI**

**ROBOT TƯỚI CÂY TỰ ĐỘNG ỨNG DỤNG CÔNG NGHỆ THỊ GIÁC MÁY TÍNH**

GVHD: TS. Cao Văn Kiên

Sinh viên : Nguyễn Cao Nguyên - 20124461

Nguyễn Tấn Phong - 20124691

TP.HCM

1. Đề tài nghiên cứu

A white background with black text

Description automatically generated

2. Nội dung thực hiện:

2.1. Phần cứng:

- Phần cứng dự kiến :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên phần cứng** | **Số lượng** | **Ghi chú** | **Giá tiền** |
| 1 | Raspberry Pi 5 | 1 |  | 2tr5-3tr |
| 2 | Webcam Logitech C270 HD 720P | 1 |  | 400k |
| 3 | Động Cơ Bơm Nước R385 Water Pump 12VDC | 1 |  | 47k |
| 4 | omnidirectional wheels | 4 | đường kính 80mm | 270k |
| 5 | Dc Motor Gear Box Reduce Ratio 1:48 | 4 | Điện áp hoạt động: 3-12VDC, Tỉ số truyền: 1:48, Moment: 0.8KG.CM | 21k x 4 = 84k |
| 6 | Module Cảm Biến Dò Line Đơn TCRT5000 4 Chân | 4 | Nguồn cung cấp: 5VDC.  Mạch sử dụng chip so sánh LM393.  Dòng điện tiêu thụ: <10mA.  Dải nhiệt độ hoạt động: 0oC ~ 50oC.  Ngõ giao tiếp: 4 dây VCC, GND, DO, AO.  Mức tín hiệu ngõ ra: TTL.  Kích thước: 3.2 x 1.4mm. | 10k x 4 = 40k |
| 7 | Cảm biến va chạm (MODULE CÔNG TẮC HÀNH TRÌNH) | 4 | Đo khoảng cách an toàn, tránh va chạm với vật cản | 25k x 4 = 100k |
| 8 | Mạch test pin, kiểm tra, đo pin LiPo | 1 | Kiểm tra dung lượng pin xem có hết | 32k |
| 9 | Cell Pin LiShen 18650 xám 2000mAh 10C | 4 | Điện áp: 3.7V  Điện áp đầy: 4.2V  Dung lượng: 2000mAh  Nội trở: 14 ~ 18Ω  Dòng xả: 10C  Xả liên tục: 10A  Xả tức thời: 30A  Dòng sạc min: 0.5A  Dòng sạc max: 1A  Kích thước: đường kính 18mm \* dài 65mm  Trọng lượng 1 pin: 45g | 40k x 4 = 160k |
| 10 | Dock sạc cho robot | 1 | Kết nối với nguồn 220V với điện áp đầu ra hiệu chỉnh sao cho phù hợp với robot | 250k |
| 11 | Bình chứa nước | 1 | Dung tích 1L |  |
| 12 | Dây cấm test board | 30 |  | 5k |
| 13 | Khung xe |  | Khung xe |  |
| 14 | Ống dây máy bơm nước 8/10 | 1m |  | 12k |
| 15 | Mạch Điều Khiển Động Cơ DC L298N | 2 | Điện áp đầu vào: 5~30VDC, Dòng tối đa cho mỗi cầu H là: 2A, Kích thước: 43x43x27mm | 38k x 2 = 76k |
| 16 | Module 1 Relay Với Opto Cách Ly Kích H/L (5VDC) | 1 | - Sử dụng điện áp nuôi DC 5V – Relay mỗi Relay tiêu thụ dòng khoảng 80mA – Điện thế đóng ngắt tối đa: AC250V ~ 10A hoặc DC30V ~ 10A – Có đèn báo đóng ngắt trên mỗi Relay – Có thể chọn mức tín hiệu kích 0 hoặc 1 qua jumper | 20k |
| 17 | Mạch giảm áp đầu ra 5V-3A | 1 | -Kích thước sản phẩm: L42mm \* W16mm \* H5mm  -Đầu vào: 7.5V-28V  -Đầu ra: 5V / 3A  -Hiệu suất chuyển đổi: 96% (Tối đa)  -Tần số chuyển mạch: 1.5MHZ (Max) 1MHz điển hình  -Nhiệt độ hoạt động: -40 ℃ – +85 ℃  -Công suất tải: đầu ra tối đa 3A, nên dùng 2A để đảm bảo độ bền | 30k |
| 18 | XL4005 Mạch Giảm Áp 5A | 1 | -Điện áp đầu vào: 5~32VDC  -Điện áp đầu ra: 1.25~30VDC  -Dòng đỉnh tối đa là 5A( dòng trung bình 3.5A).  -Hiệu suất : 90%  -Tần số:300Khz  -Nhiệt độ làm việc : -40 - 85 độ C  -Kích thước :45x21x14mm | 24k |
| Tổng cộng: | | | | 4000k-4500k |

2.2. Sơ đồ khối

2.2.1. Sơ đồ kết nối robot

Ảnh có chứa văn bản, biểu đồ, hàng, Hình chữ nhật

Mô tả được tạo tự động

2.2.2. Sơ đồ tổng quát

Ảnh có chứa biểu đồ, hàng, ảnh chụp màn hình, Hình chữ nhật

Mô tả được tạo tự động

2.2.3. Sơ đồ nguyên lý hoạt động

Ảnh có chứa văn bản, biểu đồ, Song song, Kế hoạch

Mô tả được tạo tự động

Mô tả nguyên lý hoạt động :Robot hoạt động trong môi trường giới hạn bởi đường viền đen, được xác định bởi cảm biến dò line. Robot có thể di chuyển tự do trong khu vực này và thực hiện các nhiệm vụ được giao. Robot được trang bị các cảm biến dò line, cảm biến va chạm, camera, và các động cơ. Cảm biến dò line quét môi trường xung quanh và nhận biết đường đi.Cảm biến va chạm phát hiện vật cản và gửi tín hiệu đến bộ điều khiển.Servo sẽ điều khiển camera quay qua quay lại để nhận diện đối tượng và môi trường xung quanh Motor kết nối với bánh xe điều khiển tốc độ và hướng di chuyển của robot. Raspberry Pi xử lý dữ liệu từ cảm biến và camera, điều khiển servo và động cơ để robot di chuyển tránh né vật cản và thực hiện bật/tắt relay điều khiển bơm tưới nước vào chậu cây có gắn sticker.Người dùng có thể tương tác với robot thông qua giao diện người dùng như Node-Red hoặc web

**2.3. Các yêu cầu thực hiện**

1. **Robot cần có khả năng tự di chuyển một cách linh hoạt.**

Giải pháp: Thiết kế mobile robot sử dụng 4 bánh xe đa hướng(omnidirectional wheels)

Điểm mạnh: Robot có khả năng di chuyển linh hoạt, có thể di chuyển trong không gian chật hẹp.

Hạn chế: Bánh đa hướng chỉ thích hợp hoạt động trong môi trường bằng phẳng, điều kiện thuận lợi, không thích hợp cho các môi trường gặp ghềnh.

**b.Robot cần có một cơ chế tưới nước đủ linh hoạt và chính xác để cung cấp nước cho cây cần tưới.**

Giải pháp: Sử dụng một hệ thống bơm nước được kết nối với bình chứa nước trên robot được bật tắt thông qua Relay. Hệ thống được điều khiển thông qua Raspberry Pi.

🡪 Hệ thống đơn giản, để lập trình, điều khiển dể dàng

**c. Robot cần có khả năng tự động di chuyển đến đế sạc pin khi cần.**

Giải pháp: Khi hoàn thành nhiệm vụ tưới cây hoặc pin xuống thấp một ngưỡng xác định,robot sẽ sử dụng thị giác máy tính kết hợp với cảm biến khoảng cách để xác định điểm sạc di chuyển đến đế sạc.

Phương án 1: Robot tìm đến dock thông qua thị giác máy tính để xác định vị trí của dock sạc (Sử dụng sticker để xác định vị trí) 🡪 Robot tự trở về dock sạc

Điểm mạnh: Có thể nhanh chóng xác định được vị trí của dock sạc.

Hạn chế: Vì thị giác máy tính trên Rasbperry pi không ổn định nên có khả năng không thể xác định được vị trí chính xác của dock sạc 🡪 tính ổn định của xe thấp.

Phương án 2: Robot tìm đến vị trí của dock sạc thông qua hệ thống cảm biến được trang bị trên xe.

Điểm mạnh: Robot có thể xác định được vị trí của dock sạc một cách chính xác

Hạn chế: Chỉ có thể tới gần dock sạc robot mới có thể xác định được vị trí của dock 🡪 thời gian tìm kiếm lâu

Kết luận: từ 2 phương án trên, có thể sử dụng đồng thời cả thị giác và cảm biến để tăng độ chính xác và tính ổn định của robot.

Hạn chế là chi phí sẽ tăng cao.

**d. Đế sạc robot kết nối với nguồn 220V**

Giải pháp: Đối với việc sạc, robot có thể được kết nối với một đế sạc có mức điện áp đầu vào là 220Vac.

**e. Robot chỉ hoạt động trong một khu vực giới hạn được xác định**

Giải pháp: Để robot chỉ hoạt động trong một khu vực mà người dùng chỉ định, ta có thể sử dụng các line màu đen để bao quanh khu vực hoạt động của robot 🡪Có thể dể dàng thu nỏ hoặc mở rộng khu vực làm việc của robot, Sử dụng các cảm biến dò line để đọc các line, các cảm biến dò line được gắn vào 4 phía của Robot để tăng độ chính xác cũng như tính ổn định của robot 🡪 tránh trường hợp phía không có cảm biến vượt qua khỏi line

**2. Phần Mềm:**

**a. Phần mềm cần có khả năng nhận dạng cây cần tưới và các vật cản trên đường đi của robot.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Các thuật toán thị giác máy tính** | **Tốc độ xử lý** | **Độ chính xác** | **Số lượng tài nguyên cần sử dụng** |
| 1 | CNN (Convolutional Neural Networks): | Phụ thuộc vào kiến trúc cụ thể của mô hình và phần cứng được sử dụng. CNN có thể tốn nhiều thời gian xử lý, đặc biệt là trên dữ liệu lớn | CNN có thể đạt được độ chính xác cao, nhưng cũng phụ thuộc vào cách mà mô hình được đào tạo và cấu hình | CNN đòi hỏi nhiều tài nguyên tính toán và bộ nhớ, đặc biệt là khi áp dụng cho các tác vụ phức tạp trên ảnh và video |
| 2 | YOLO (You Only Look Once) | YOLO nhanh hơn nhiều so với nhiều phương pháp truyền thống, vì nó áp dụng việc dự đoán trực tiếp trên toàn bức ảnh một cách hiệu quả | YOLO có thể đạt được độ chính xác tương đối cao, đặc biệt là khi cân nhắc giữa tốc độ và độ chính xác | Tùy thuộc vào phiên bản cụ thể và cấu hình, nhưng YOLO thường cần ít tài nguyên hơn so với một số phương pháp khác |
| 3 | YOLO-tiny | YOLO-tiny nhanh hơn YOLO do nó sử dụng một mạng nhỏ hơn với ít tầng | Độ chính xác của YOLO-tiny thường thấp hơn so với YOLO vì kích thước mạng nhỏ hơn | YOLO-tiny yêu cầu ít tài nguyên tính toán và bộ nhớ hơn so với YOLO |
| 4 | TensorFlow và TensorFlow Lite | TensorFlow Lite được tối ưu hóa để chạy trên các thiết bị di động và nhúng, do đó, thường nhanh hơn TensorFlow trên các thiết bị có tài nguyên hạn chế | TensorFlow và TensorFlow Lite có thể đạt được độ chính xác tương đối cao, tùy thuộc vào mô hình được sử dụng | TensorFlow Lite cung cấp một phiên bản nhỏ gọn và tối ưu hóa hơn so với TensorFlow, do đó cần ít tài nguyên hơn để triển khai trên các thiết bị có tài nguyên hạn chế |
| **Kết luận:** | | Với việc sử dụng phần cứng là Raspberry Pi 5 với phần cứng hạn chế, ta có thể thấy trong các thuật toán thị giác máy tính trên, Tensorflow lite là được tối ưu tốt nhất để có thể chạy ổn định trên Rasbperry Pi mà vẩn có thể thực hiện được việc Real time Detect Objection mà vẩn có sự ổn định và độ chính xác vùa phải. | | |

Giải pháp: Sử dụng TensorFlow Lite và OpenCV để nhận diện vật phẩm

Điểm mạnh:

Tối ưu hóa cho Raspberry Pi: TensorFlow Lite là một lựa chọn tốt cho Raspberry Pi với tài nguyên hạn chế về phần cứng, giúp đảm bảo hiệu suất ổn định và đáng tin cậy.

Độ chính xác: Mặc dù TensorFlow Lite là một phiên bản nhỏ gọn của TensorFlow, nhưng vẫn cung cấp độ chính xác tốt đối với nhiều nhiệm vụ nhận diện vật phẩm.

Sự hỗ trợ từ cộng đồng: TensorFlow Lite và OpenCV đều được sử dụng rộng rãi và có sự hỗ trợ mạnh mẽ từ cộng đồng người dùng và các nhà phát triển, giúp giải quyết các vấn đề và tìm kiếm giải pháp nhanh chóng.

Hạn chế và cách khắc phục:

Hiệu suất: Raspberry Pi có tài nguyên phần cứng hạn chế, do đó, việc chạy các mô hình nhận diện vật phẩm có thể gặp phải vấn đề về hiệu suất. Để khắc phục, có thể sử dụng các mô hình nhỏ hơn, tinh chỉnh siêu tham số hoặc tối ưu hóa mã nguồn.

Độ trễ: Do tài nguyên hạn chế của Raspberry Pi, có thể xảy ra độ trễ trong quá trình nhận diện vật phẩm. Để giảm thiểu độ trễ, có thể tối ưu hóa mã nguồn và sử dụng các phương pháp song song để tăng hiệu suất.

Vì tài nguyên hạn chế, việc đạt được độ chính xác cao có thể khá thách thức. Có thể sử dụng các vật phẩm đặc biệt như sticker cây, sticker sạc để thay thế cho cây và sạc có các đặc điểm đặc biệt cũng như dễ dàng nhận biết, giúp cho độ chính xác khi nhận diện có thể tăng cao hơn.

Tóm lại: Sử dụng TensorFlow Lite và OpenCV là một giải pháp phù hợp để nhận diện vật phẩm trên Raspberry Pi với tài nguyên hạn chế. Tuy nhiên, việc tối ưu hóa và điều chỉnh các thiết lập là quan trọng để đảm bảo hiệu suất và độ chính xác tốt nhất trong điều kiện thực tế.

**b. Phần mềm cần điều khiển robot di chuyển đến đúng vị trí của đế sạc pin khi cần.**

Giải pháp: Khi dung lượng pin xuống thấp hoặc sau khi tưới xong cây, robot có thể tự động quay lại ví trí của dock sạc/ đế sạc bằng thị giác máy tính hoặc hệ thống cảm biến, định vị.