**Thuật toán sinh mã QR**

1. **Bước 1 - Nhận dữ liệu từ người dùng:**  
   Trong bước đầu tiên, chương trình yêu cầu người dùng nhập vào một chuỗi văn bản. Chuỗi văn bản này sẽ được sử dụng làm dữ liệu để sinh mã QR. Dữ liệu được nhập vào thông qua hàm input(), và sau đó được lưu trữ trong một biến có tên là text. Đây là bước nền tảng, quyết định nội dung thông tin được mã hóa trong mã QR.
2. **Bước 2 - Kiểm tra và thiết lập tên file:**  
   Sau khi nhận dữ liệu từ người dùng, chương trình tiến hành kiểm tra xem tên file để lưu mã QR có được cung cấp hay không. Nếu người dùng không chỉ định tên file, chương trình sẽ tự động tạo ra một tên file dựa trên chuỗi văn bản đã nhập. Để đảm bảo tính hợp lệ của tên file, các ký tự đặc biệt như dấu gạch chéo (/) và dấu gạch chéo ngược (\) sẽ được thay thế bằng dấu gạch dưới (\_). Việc tạo tên file hợp lệ đảm bảo tính tương thích của file với hệ thống tệp.
3. **Bước 3 - Khởi tạo đối tượng mã QR:**  
   Ở bước này, chương trình khởi tạo một đối tượng QRCode, trong đó các tham số quan trọng được thiết lập như sau:
   * Phiên bản (version): Xác định kích thước của mã QR. Ở đây, phiên bản 1 được chọn, tương ứng với kích thước ma trận 21x21 ô vuông.
   * Sửa lỗi (error\_correction): Chọn mức độ sửa lỗi thấp nhất là **ERROR\_CORRECT\_L**. Mức sửa lỗi này đảm bảo mã QR có thể khôi phục thông tin ngay cả khi một phần nhỏ bị hư hỏng, đồng thời giữ dung lượng thấp nhất.
4. **Bước 4 - Thêm dữ liệu vào mã QR:**  
   Sau khi đối tượng mã QR đã được khởi tạo, dữ liệu chuỗi văn bản từ bước 1 sẽ được thêm vào đối tượng thông qua phương thức add\_data(). Quá trình này giúp mã hóa thông tin văn bản thành các ô vuông trong mã QR. Tiếp theo, phương thức make(**fit=True**) được gọi để đảm bảo mã QR tự động điều chỉnh kích thước sao cho phù hợp với lượng dữ liệu đã thêm.
5. **Bước 5 - Chuyển đổi kích thước hình ảnh:**  
   Sau khi mã QR được tạo dưới dạng hình ảnh, chương trình tiến hành điều chỉnh kích thước ảnh về mức cố định 400x400 pixel. Việc này đảm bảo hình ảnh mã QR có kích thước chuẩn, dễ dàng sử dụng và hiển thị trên nhiều thiết bị khác nhau. Phương pháp lấy mẫu Lanczos (Image.Resampling.LANCZOS) được sử dụng trong quá trình thay đổi kích thước để giữ cho hình ảnh có độ phân giải và chất lượng tốt nhất.
6. **Bước 6 - Lưu hình ảnh mã QR:**  
   Cuối cùng, mã QR sau khi được tạo và điều chỉnh kích thước sẽ được lưu dưới dạng file hình ảnh. Tên file sẽ là tên được thiết lập từ bước 2, và file sẽ được lưu vào hệ thống dưới định dạng .png. Sau khi quá trình lưu hoàn tất, chương trình sẽ thông báo cho người dùng rằng mã QR đã được lưu thành công với tên file tương ứng.

**Thuật toán đọc mã QR và hành động tương ứng của UGOT**

1. **Bước 1 - Khởi tạo và thiết lập UGOT**:  
   Đầu tiên, hệ thống khởi tạo một đối tượng UGOT thông qua lớp ugot.UGOT(). Địa chỉ IP của robot được thiết lập để kết nối, đồng thời camera của UGOT được mở để nhận diện và xử lý mã QR.
2. **Bước 2 - Quay và đọc mã QR**:  
   Quá trình quét mã QR bắt đầu bằng cách liên tục lấy các khung hình từ camera của UGOT. Các khung hình này được xử lý để chuyển đổi sang dạng hình ảnh RGB. Để tăng khả năng nhận diện mã QR trong nhiều góc độ khác nhau, hệ thống xoay ảnh theo nhiều góc từ -50 đến 50 độ.

Một đối tượng **QRCodeDetector của OpenCV** được sử dụng để phát hiện và giải mã mã QR. Nếu tìm thấy mã QR hợp lệ, thông tin văn bản của mã sẽ được giải mã và lưu vào danh sách **qr\_codes.**

1. **Bước 3 - Hiển thị mã QR**:  
   Khi một mã QR được phát hiện, thông tin của mã sẽ được hiển thị thông qua phương thức screen\_print\_text\_newline của UGOT, giúp in thông tin lên màn hình và đồng thời lưu mã QR đã giải mã thành công vào danh sách để xử lý tiếp theo.
2. **Bước 4 - Xử lý thông tin mã QR và điều khiển UGOT**:  
   Khi có mã QR trong danh sách, hệ thống sẽ lấy mã đầu tiên để xử lý. Mỗi ký tự trong chuỗi mã QR sẽ đại diện cho một hành động cụ thể của UGOT. Chẳng hạn:
   * Ký tự '1' yêu cầu UGOT di chuyển thẳng với tốc độ và thời gian xác định.
   * Ký tự '2' yêu cầu UGOT di chuyển lùi .
   * Ký tự '3', '4' yêu cầu robot quay với các góc khác nhau.
   * Các ký tự như 'l' và 'o' đại diện cho các chuỗi hành động phức tạp hơn, bao gồm di chuyển và quay theo thời gian và tốc độ định sẵn.

Nếu gặp một ký tự không xác định, UGOT sẽ hiển thị thông báo và bỏ qua ký tự đó.

1. **Bước 5 - Vòng lặp xử lý liên tục**:  
   Hệ thống duy trì vòng lặp liên tục, kiểm tra và quét mã QR mới. Sau khi xử lý mỗi mã QR, hệ thống tạm dừng một thời gian ngắn để chờ thông tin mới.

Thuật toán này kết hợp khả năng quét mã QR và điều khiển hành động vật lý, giúp UGOT thực hiện các lệnh được mã hóa trong mã QR. Thông qua việc sử dụng camera và các thuật toán xử lý ảnh, hệ thống có thể phát hiện và xử lý các mã QR ngay cả khi chúng bị xoay góc.

* Kết quả là UGOT có thể thực hiện các chuyển động cơ học dựa trên thông tin quét được.