Quy tắc

* Bạn không được phép sửa đổi hoặc bỏ qua các giới hạn vật lý của rô bốt.
  + Một robot vật lý cần thời gian để tăng tốc và phanh. Điều này không thể xảy ra ngay lập tức. Các tối đa tỷ lệ thay đổi tốc độ được định nghĩa bởi DEFAULT\_SPEED\_CHANGE = 1 (mỗi bước thời gian) trong mẫu mã (structures.h) và không thể thay đổi .
  + Robot vật lý có tốc độ tối đa . Điều này được định nghĩa bởi MAX\_ROBOT\_SPEED = 25 trong mã mẫu (struct.h) và không thể thay đổi . Bạn luôn có thể đi chậm hơn mức tối đa này, nhưng bạn không thể vượt quá nó.
  + Một robot vật lý (BOE Shield-bot) quay bằng cách quay một cặp bánh xe ở hai bên của khung robot. Nó không thể "lật" 180 độ ngay lập tức. Nó không thể quay 90 độ trong một lần di chuyển. Robot có tốc độ quay tối đa được xác định bởi DEFAULT\_ANGLE\_CHANGE = 15 (độ trên mỗi bước thời gian) trong mã mẫu (cấu trúc.h). Nếu bạn muốn rô-bốt quay 180 độ - giả sử bạn đến ngõ cụt - bạn có thể đạt được điều này bằng cách thực hiện 12 lệnh 15 độ (LEFT hoặc RIGHT) liên tiếp. Điều này rõ ràng sẽ yêu cầu 12 bước thời gian.
  + Robot vật lý được cung cấp các cảm biến mặc định. Chúng là chùm tia hẹp (giống như tia laser). Chúng có phạm vi phát hiện tối đa được xác định bởi SENSOR\_VISION = 30 (pixel) trong mã mẫu (struct.h) và không thể thay đổi điều này . Để tham khảo, hãy nhớ rằng 1 pixel = 5mm.
* Mặc dù chúng rất tuyệt, nhưng bạn không được phép sử dụng Ommi và Mechanum bánh xe.
* Bạn không được phép sử dụng các bộ vi điều khiển thay thế - chẳng hạn như Teensy 4 - để xử lý nhanh hơn hoặc tăng số lượng chân I / O. Bạn phải làm việc trong giới hạn của Arduino Uno.

### **Điều gì đã được chấp thuận**

* Bạn có thể giảm tốc độ thay đổi tốc độ xuống dưới DEFAULT\_SPEED\_CHANGE tối đa là 1 cho mỗi bước thời gian. Điều này có ý nghĩa vì robot vật lý trong phòng thí nghiệm sử dụng động cơ servo quay liên tục trên các bánh xe có thể được điều khiển trên một loạt tốc độ (tối đa). Để làm cho mã của bạn dễ đọc, bạn nên sử dụng (x) \* (DEFAULT\_SPEED\_CHANGE), trong đó x là giá trị nhỏ hơn 1,0. Bằng cách đó, các gia sư tin tưởng rằng bạn không thực hiện thay đổi tốc độ lớn hơn tốc độ tối đa mặc định.
* Bạn có thể giảm tốc độ xoay xuống dưới DEFAULT\_ANGLE\_CHANGE tối đa là 15 độ mỗi bước thời gian. Một lần nữa, điều này có ý nghĩa khi robot vật lý quay / quay bằng cách sử dụng động cơ servo quay liên tục trên các bánh xe. Để làm cho mã của bạn dễ đọc, bạn nên sử dụng (x) \* (DEFAULT\_ANGLE\_CHANGE), trong đó x là giá trị nhỏ hơn 1,0. Bằng cách đó, người dạy kèm có thể tin tưởng rằng bạn không thực hiện thay đổi góc lớn hơn mức tối đa mặc định.
* Bạn có thể thay đổi vị trí của các cảm biến trên khung robot và có thể thay đổi hướng mà chúng trỏ đến.
* Bạn có thể tăng số lượng cảm biến được phép, với các hạn chế và cân nhắc sau:
  + Arduino Uno có một số chân I / O xác định. Bạn không thể thêm nhiều cảm biến hơn số chân để kết nối chúng.
  + Cảm biến có kích thước vật lý và robot chỉ là 100mm x 100mm. Bạn phải có thể "lắp" tất cả các cảm biến của mình vào rô bốt. Nếu bạn cần thêm không gian, bạn sẽ cần phải làm cho robot của mình lớn hơn và điều này có ý nghĩa đối với việc điều hướng các mê cung hẹp. Nếu bạn đã thêm các cảm biến bổ sung, bạn nên lập một sơ đồ đơn giản trong báo cáo của mình cho thấy khu vực bị chiếm dụng bởi từng cảm biến và vị trí của chúng trên rô bốt để chứng minh với người đánh dấu rằng các cảm biến bạn đã chọn đều phù hợp. Kích thước của cảm biến có thể được lấy từ bảng dữ liệu của chúng.
* Bạn có thể chọn các loại cảm biến khác nhau cho các loại cảm biến mặc định được cung cấp trong mã mẫu, với các hạn chế và cân nhắc sau:
  + Bạn cần tìm một cảm biến vật lý hợp lý (nhỏ, cấp độ sở thích) và phù hợp với hiệu suất của nó trong mã của bạn. Bạn cần bảng dữ liệu để chứng minh / xác nhận cách bạn đã khớp giữa cảm biến vật lý và cảm biến ảo.
  + Góc chùm tia đối với cảm biến siêu âm là khoảng 30 độ (tức là chúng không hẹp như cảm biến mặc định). Nếu bạn có nhiều cảm biến, bạn nên đảm bảo rằng góc chùm của chúng không trùng nhau. Hãy suy nghĩ về cách bạn có thể diễn giải việc phát hiện một đối tượng nằm trong vùng chồng lấn.
  + Một số cảm biến tầm xa hơn có thể có khoảng cách phát hiện tối thiểu . Ví dụ, cảm biến siêu âm HC-SR04 không thể phát hiện bất cứ thứ gì gần hơn 2cm. Hãy suy nghĩ về cách bạn có thể xây dựng điều đó vào mô hình.
  + Nếu bạn phát hiện thứ gì đó cách xa 1m bằng cảm biến tầm xa, hãy nghĩ xem robot có thể thực hiện hành động gì vì nó ở rất xa vật thể đó.
* Robot có thể turn on the spot - tức là nó không cần phải tiến hoặc lùi trong khi nó đang quay. Điều này có thể đạt được bằng cách đưa robot dừng lại (chú ý đến tốc độ làm chậm tối đa) và sau đó thực hiện các lệnh LEFT hoặc RIGHT liên tiếp (một lệnh cho mỗi bước thời gian). Điều này có thể hữu ích để đối phó với một ngõ cụt.
* Bạn có thể sử dụng bất kỳ thuật toán nào bạn thích để tự động điều hướng các mê cung.
* Bạn có thể sử dụng tọa độ tuyệt đối thay vì tọa độ tương đối nếu muốn, nhưng phải nhất quán trong toàn bộ mã của bạn. NASA nổi tiếng đã [mất Tàu quỹ đạo khí hậu sao Hỏa của họ (Liên kết đến một trang bên ngoài.)](https://www.simscale.com/blog/2017/12/nasa-mars-climate-orbiter-metric/) bởi các đơn vị trộn.
* Bạn có thể thêm chướng ngại vật mới vào mê cung mẫu của mình (như đá) hoặc yếu tố chuyển động (như cửa đóng / mở).
* Bạn có thể thêm các bức tường không trực giao vào mê cung mẫu của mình. Ví dụ bao gồm các bức tường chéo (ở bất kỳ góc độ nào bạn chọn) hoặc các bức tường cong.
* Khi khung rô bốt (và các cảm biến) được xoay, hộp va chạm của rô bốt không được quay. Nếu bạn cho rằng điều này có thể dẫn đến sự cố 'giả' khi rô bốt của bạn đi qua rất gần một bức tường, bạn được phép sửa đổi mã để hộp va chạm có thể xoay với phần còn lại của rô bốt. Một lựa chọn tốt hơn là làm việc trên các cảm biến và đường dẫn điều hướng của bạn để giữ robot cách xa các bức tường hơn.