Đề thi SRobot Liên trường 2019

(Thời gian: 180 phút)

(Đề thi có 3 trang)

1. Giới thiệu

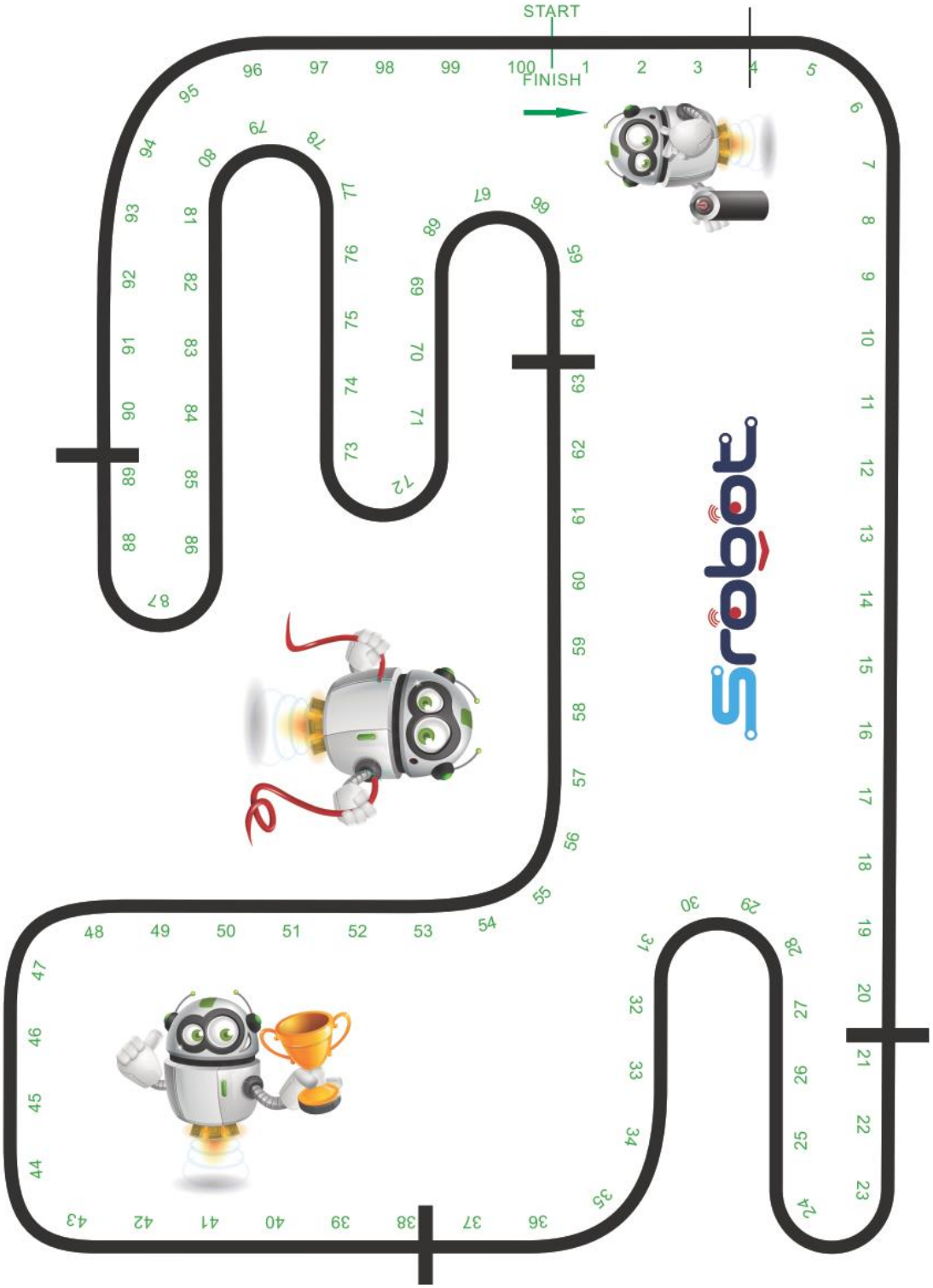
- Các đội dự thi viết chương trình điều khiển robot sử dụng ngôn ngữ lập trình Pascal hoặc C++ di chuyển theo địa hình thi đấu do ban tổ chức cung cấp ở mục số 2.
- Robot được lập trình để tự động di chuyển theo chiều mũi tên trên đường đi màu đen. Đường đi được đánh số theo thứ tự tăng dần một đơn vị trên nền màu trắng của địa hình.
- Robot được đặt trước vạch START, thời gian thi đấu được tính từ lúc robot đi qua vạch này. Thời điểm kết thúc được tính khi robot di chuyển qua vạch FINISH.
- Các đội dự thi không được phép tương tác với robot thông qua các bất cứ hình thức nào trong quá trình thi đấu.
- Tại vị trí cách vạch START 60 cm sẽ có một vật cản. Vật cản sẽ được di dời sau 5 giây kể từ lúc robot di chuyển qua vạch START. Các đội tham khảo phần gợi ý để có thể xử lý đoạn đường từ vạch START đến vật cản.
- Trên đường đua sẽ có các vạch báo trước và sau các đoạn quanh co, robot nên nhận biết để giảm hoặc tăng tốc độ để có thể hoàn thành bài thi tốt nhất.
- Quy định:
  - Các đội có **2** lượt thi đấu. Kết quả sẽ được chọn theo lượt thi đấu tốt nhất.
  - Các đội có quyền lựa chọn thi đấu trên sa bàn có vật cản hoặc không có vật cản.
  - Khi thi đấu trên sa bàn có vật cản, thành tích được công nhận của robot là thời gian di chuyển của robot (được hiển thị trên đồng hồ của Ban tổ chức) trong trường hợp robot hoàn thành đường đua (di chuyển hợp lệ từ START đến vạch FINISH) hoặc bằng quãng đường hợp lệ mà robot đi được trong trường hợp robot không thể hoàn thành đường đua.
  - Khi thi đấu trên sa bàn không có vật cản, nếu robot hoàn thành đường đua, thành tích được công nhận của robot bằng thời gian di chuyển của robot **cộng thêm 30 giây**. Ví dụ, nếu robot hoàn thành vòng đua với thời gian là 1 phút 30 giây, thành tích được công nhận của robot sẽ là 2 phút.
  - Khi thi đấu trên sa bàn không có vật cản, nếu robot không thể hoàn thành đường đua một cách hợp lệ, thành tích được công nhận của robot bằng quãng đường di chuyển hợp lệ **trừ 10 đơn vị**. Ví dụ, robot di chuyển hợp lệ được quãng đường bằng 15 đơn vị, thành tích được công nhận của robot sẽ là 5 đơn vị.
  - Các đội dự thi được yêu cầu lập trình cho robot di chuyển hợp lệ và hoàn thành **1** lượt thi đấu trong thời gian tối đa **5** phút.
- Robot bị xem là di chuyển không hợp lệ trong các trường hợp sau:

- Ra khỏi đường đi và không quay trở lại, hoặc quay trở lại nhưng cách điểm rời khỏi đường đi quá 3 đơn vị.
    - Điểm rời khỏi đường đi được tính tại vị trí cuối cùng của robot rời khỏi đường đi.
    - Điểm quay lại đường đi được tính tại vị trí đầu tiên của robot chạm vào đường đi.
- Di chuyển ngược chiều qua vạch START sau khi vượt qua vị trí chướng ngại vật

- Robot chỉ được để hai bánh xe tiếp xúc với mặt sân, ngoài ra không có bất kỳ các bộ phận nào khác tiếp xúc với mặt sân một cách liên tục.
- Lưu ý:
    - Giữa các lượt thi đấu, các đội dự thi được thực hiện một số chỉnh sửa nhỏ như thay dây nối, siết ốc, làm sạch bánh xe,… mà không được thay đổi các bộ phận của robot. Việc chỉnh sửa không quá 10 phút và được sự giám sát của BTC.
    - Các đội không được đổi robot trong suốt cuộc thi.
    - Các đội có thể kiểm thử robot trên địa hình thi đấu do ban tổ chức cung cấp trong suốt thời gian làm bài.
- **<u>Gợi ý:</u>**
    - Với cảm biến khoảng cách sử dụng sóng siêu âm, thời gian từ lúc phát sóng siêu âm đến lúc nhận được sóng siêu âm phụ thuộc vào khoảng cách từ cảm biến đến vật cản. Trong trường hợp vật cản đặt rất xa robot, thời gian thực thi hàm đo khoảng cách rất lớn, ảnh hưởng đến việc giữ thăng bằng của robot.
    - Các đội nên sử dụng hàm `pulseIn` kết hợp tham số `timeout`:
        - Với Arduino IDE, các đội sử dụng hàm `pulseIn` theo cấu trúc: `pulseIn(pin, value, timeout)`.
        - Với SArduino, các đội sử dụng hàm `pulseInTimeout(pin: integer; value: integer; timeout: longint)`.

        Với `pin` là chân nối với chân echo của cảm biến đo khoảng cách, `value` là `HIGH`, `timeout` là thời gian chờ tín hiệu. Nếu quá thời gian `timeout` mà chưa có tín hiệu trả về, các hàm `pulseIn` sẽ trả về giá trị 0.
    - Để đảm bảo giữ thăng bằng, các đội nên sử dụng giá trị `timeout` tối đa là 3000. Với giá trị trên, nếu vật cản cách cảm biến quá 40 cm, hàm `pulseIn` sẽ trả về giá trị 0.

2. Địa hình thi đấu



--- Hết ---