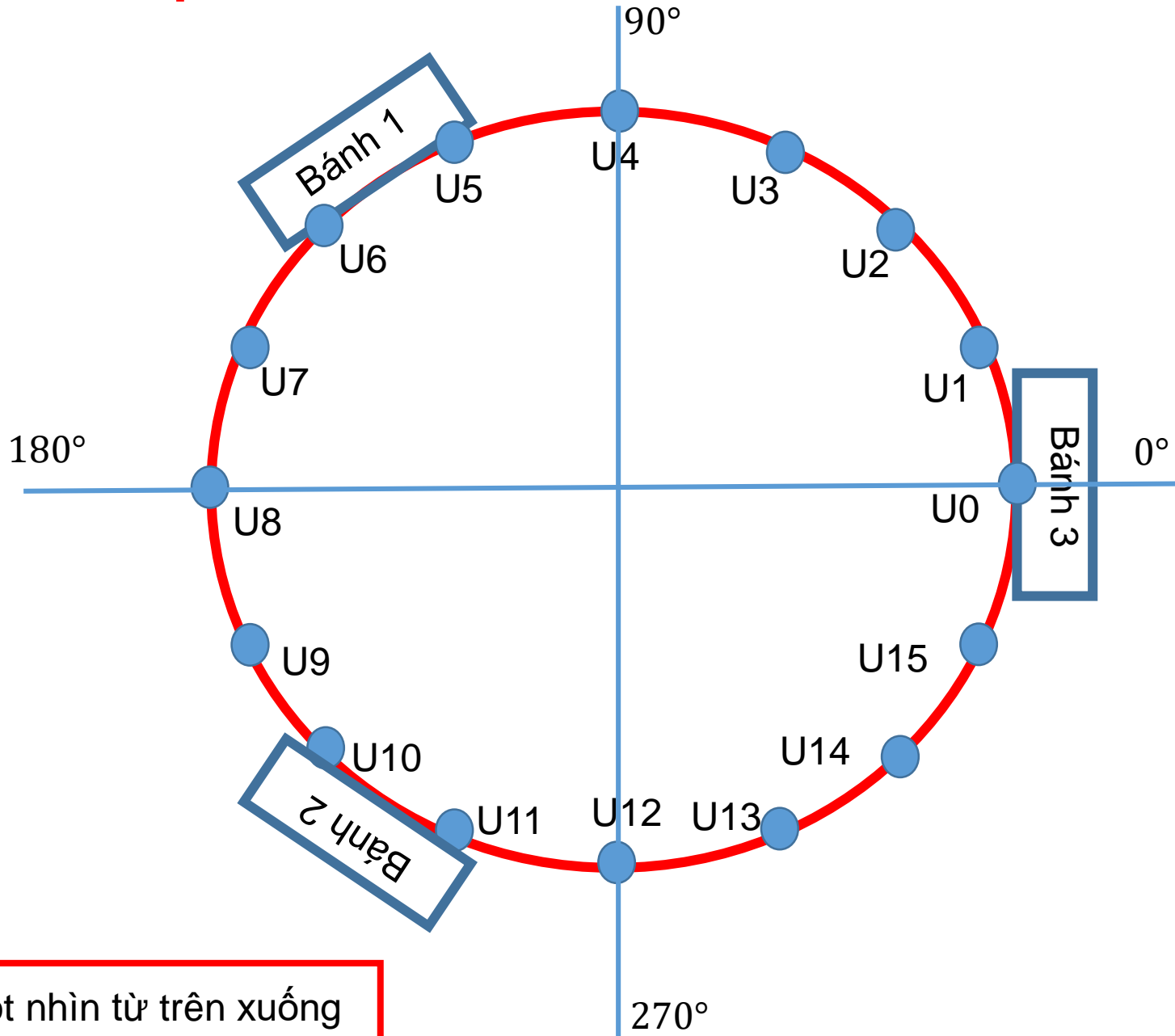


# Hướng dẫn dò line cho Omni-directional Robot

SArduino Training 2019 cho THPT  
Saigon Institute of Technology



# Nhắc lại sơ đồ bố trí cảm biến

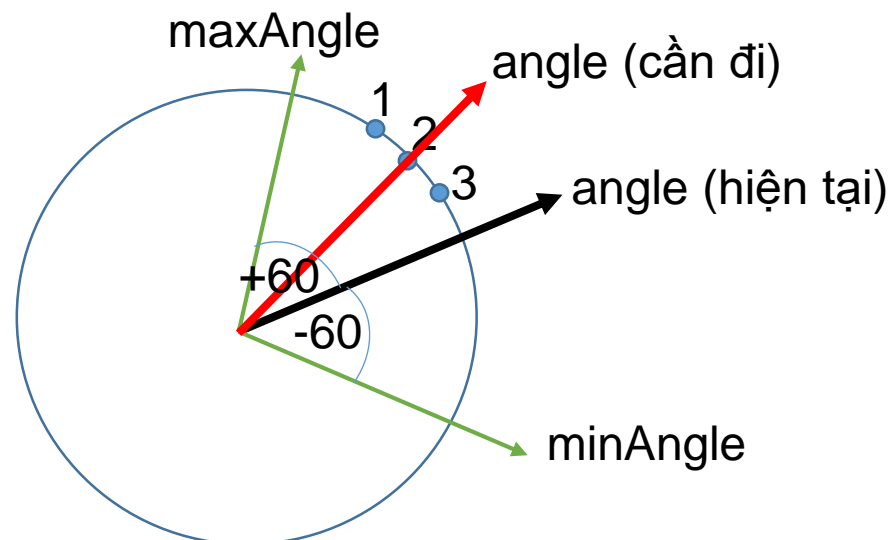


\*Robot nhìn từ trên xuống

# Đề xuất phương án bám line

(Lưu ý phương án sau là phương án đề xuất, thí sinh được khuyến khích sáng tạo các phương thức bám line cho riêng mình để đạt hiệu quả cao nhất):

- Robot xuất phát theo 1 góc do thí sinh tự chọn.
- Từ góc đó, ta xét  $+60^\circ$  và  $-60^\circ$  để kiểm tra các cảm biến gần (hướng đang đi) nhất, và cảm biến đó đang nằm trên line (Kiểm tra dựa trên giá trị Analog đọc vào).
- Chọn hướng đi kế tiếp là góc trung bình của tổng các góc tương ứng với các cảm biến này.



## Pseudo Code (Code giả)\*

//angle là góc hiện tại robot đang di chuyển.

//minAngle là 60 độ về phải phải của angle.

//maxAngle là 60 độ về phía trái của angle.

//Lưu ý chiều dương của góc là chiều ngược chiều kim đồng hồ

**minAngle = angle - 60;**

**maxAngle = angle + 60;**

//Chú ý cần chia 2 trường hợp để xét

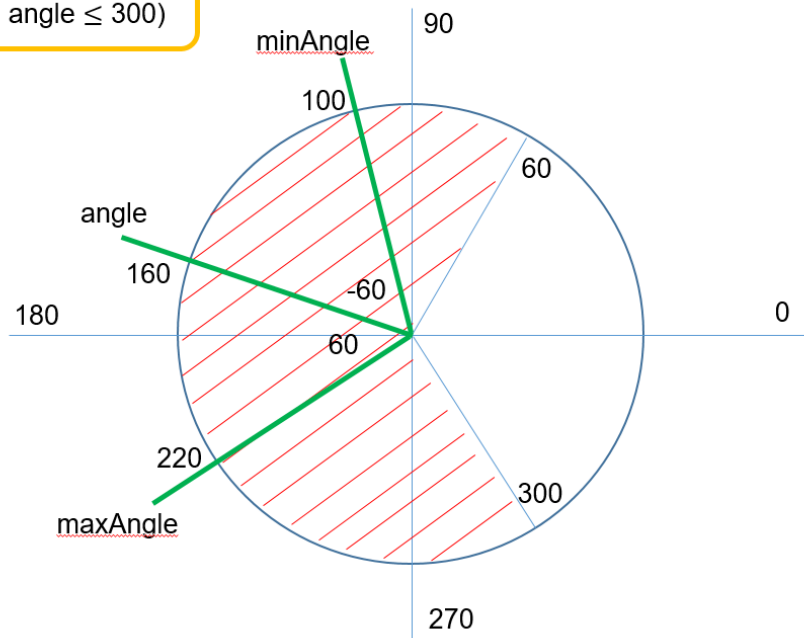
//Trường hợp 1: minAngle < maxAngle

//Trường hợp 2: minAngle > maxAngle

*\*Lưu ý: Pseudo code là chương trình đề xuất, người dùng cần điều chỉnh cú pháp và chỉnh sửa cho phù hợp với ngôn ngữ mình sử dụng*

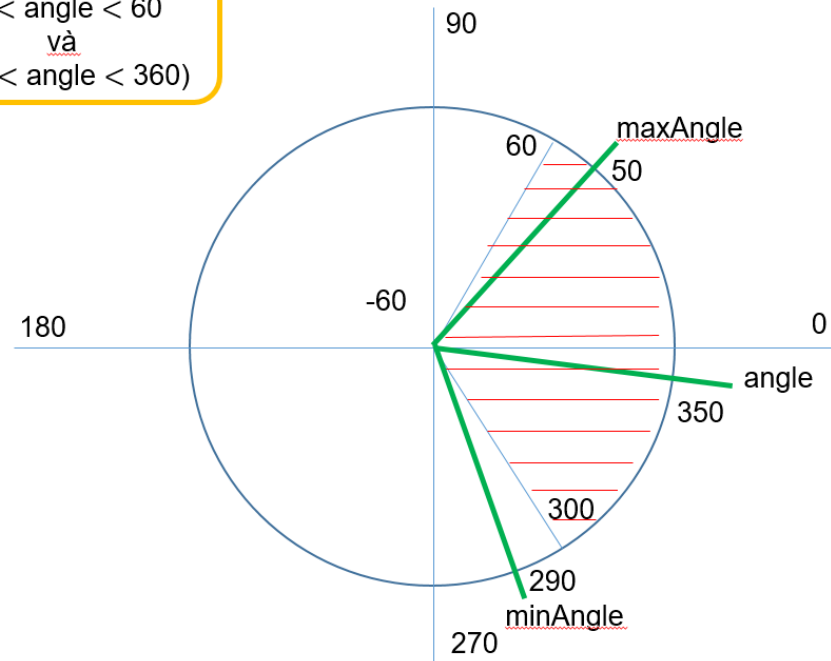
# Pseudo Code (Code giả)\*

$\text{minAngle} < \text{maxAngle}$   
( $60 \leq \text{angle} \leq 300$ )



Trường hợp 1

$\text{minAngle} > \text{maxAngle}$   
( $0 < \text{angle} < 60$   
và  
 $300 < \text{angle} < 360$ )



Trường hợp 2

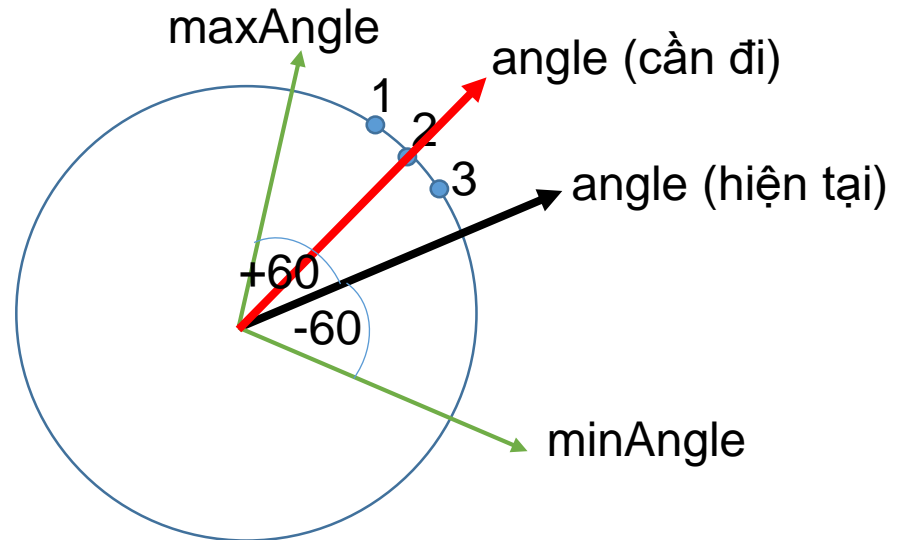
*\*Lưu ý: Pseudo code là chương trình đề xuất, người dùng cần điều chỉnh cú pháp và chỉnh sửa cho phù hợp với ngôn ngữ mình sử dụng*

# Pseudo Code (Code giả)\*

```
//Nếu giá trị đọc được từ cảm biến cho biết cảm biến nằm trong khoảng từ  
//minAngle đến maxAngle, và nằm trong khu vực màu đen → Thì hướng  
//cần đi sẽ là trung bình của các góc tương ứng //của những cảm biến này
```

Ví dụ:

-Góc đang đi là  
angle(hiện tại), 3 cảm biến  
đang **nằm trên màu**  
**Đen**, và **trong vùng +- 60 độ**  
là 1,2,3 tương  
ứng 337.5°, 315°, 292.5°.



→Góc cần đi kế tiếp sẽ là:

$$\text{angle (cần đi)} = (337.5 + 315 + 292.5) / 3 = 315$$

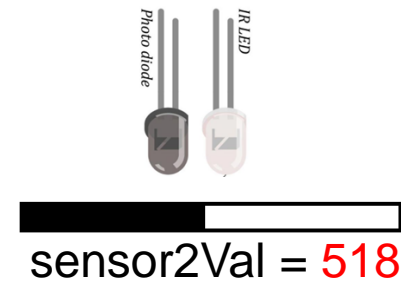
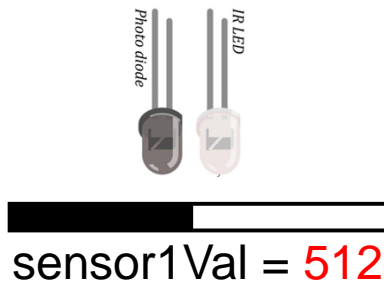
# Nâng cao: Calibrate cảm biến

- Đặt vấn đề:

-Các cảm biến khác nhau sẽ đọc giá trị không hoàn toàn giống nhau tại cùng 1 vị trí. Ví dụ có 2 cảm biến sensor1 và sensor2 như dưới (Do vấn đề sản xuất):

```
sensor1Val = analogRead(sensorPin1);
```

```
sensor2Val = analogRead(sensorPin2);
```



-Tuy đọc cùng 1 vị trí nhưng giá trị lại có chênh lệch nhỏ.

# Phương pháp

- Người dùng cần tìm giá trị nhỏ nhất (minSensorVal) và giá trị lớn nhất (maxSensorVal) mà **mỗi** cảm biến có thể đọc được trong 1 khoảng thời gian định sẵn (ví dụ 5 giây). Sau đó đưa về cùng 1 khoảng giá trị (Ví dụ: 0-1023) bằng cách sử dụng hàm map().
- Link tham khảo:

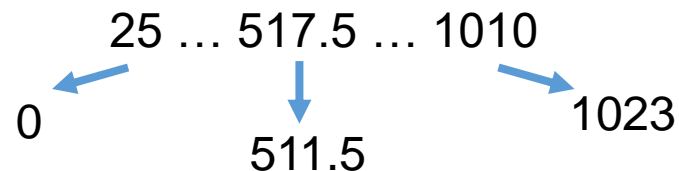
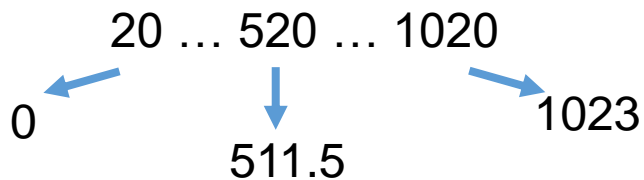
<https://www.arduino.cc/en/Tutorial/Calibration>



# Ví dụ mô tả



Tùy vào điều kiện sản xuất của cảm biến mà cảm biến 1 đọc được giá trị từ 20 đến 1020, cảm biến 2 đọc được từ 25 đến 1010. Do đó, nếu xét ngưỡng 511.5 ( $>511.5$  là màu đen và  $<511.5$  là màu trắng) thì không chính xác vì lúc này đối với cảm biến 1 ngưỡng cần xét là  $(20 + 1020)/2 = 520$  và cảm biến 2 là  $(25 + 1010)/2 = 517.5$ . Do đó, nếu calibrate để:



thì khi đó, chỉ cần xét ngưỡng là  $(0 + 1023)/2 = 511.5$  có thể đảm bảo các cảm biến đồng bộ với nhau.

*\*Lưu ý: Pseudo code là chương trình đề xuất, người dùng cần điều chỉnh cú pháp và chỉnh sửa cho phù hợp với ngôn ngữ mình sử dụng*

## Pseudo code (Code giả)\*

//Lưu ý cần khai báo minSensorVal với giá trị tương đối lớn (ví dụ 1023). Vì nếu khai  
//báo minSensorVal = 0, và giá trị nhỏ nhất cảm biến có thể đọc là 20, thì thay vì  
//minSensorVal = 20 (là kết quả mong muốn), đoạn chương trình sẽ lưu  
//minSensorVal = 0 (là kết quả không mong muốn)

```
int minSensorVal = 1023;  
int maxSensorVal = 0;
```

```
Setup {  
    //Trong thời gian 5 giây:  
    while (millis() < 5000) {  
        //Đọc liên tục giá trị từ cảm biến. Chú ý cần di chuyển cảm biến liên tục  
        //từ vùng trắng sang vùng đen để cảm biến có thể đọc giá trị nhỏ nhất  
        //và lớn nhất  
        sensorVal = analogRead(sensorPin);  
        //Lấy giá trị nhỏ nhất trong các lần đọc làm giá trị minSensorVal.  
        if (sensorVal < minSensorVal) {  
            minSensorVal = sensorVal;}  
        //Lấy giá trị lớn nhất trong các lần đọc làm giá trị maxSensorVal.  
        if (sensorValue > maxSensorVal) {  
            maxSensorVal = sensorValue;}  
    }  
}
```

*\*Lưu ý: Pseudo code là chương trình đề xuất, người dùng cần điều chỉnh cú pháp và chỉnh sửa cho phù hợp với ngôn ngữ mình sử dụng*

---

## Pseudo code (Code giả)\*

```
loop{
//Trong chương trình chính (Main/loop):
//Đọc giá trị cảm biến
    sensorVal = analogRead(sensorPin) ;
//Đưa giá trị này về tỉ lệ tương ứng trong khoảng 0-255 để đảm bảo
//tất cả các cảm biến đều tương đồng.
    sensorVal = map(sensorVal, minSensorVal,
maxSensorVal, 0, 1023) ;
//Trường hợp có giá trị nằm ngoài khoảng minSensorVal và maxSensorVal
//đã calibrate ban đầu: Dùng hàm constrain(0, 1023) để nếu giá trị nhỏ hơn
0 thì bằng 0, lớn hơn 10123 thì bằng 1023
    sensorVal = constrain(sensorVal, 0, 1023) ;
}
```