TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN

**KHOA CÔNG NGHỆ THÔNG TIN**

\*\*\*\*\*\*\*\*



**BÁO CÁO ĐỒ ÁN 4**

ĐỀ TÀI

**HỆ THỐNG ĐỊNH VỊ ÁNH SÁNG MẶT TRỜI**

**ỨNG DỤNG TRONG TẤM PIN NĂNG LƯỢNG MẶT TRỜI**

Giáo viên hướng dẫn: **Nguyễn Vinh Quy**

Sinh viên: **Nguyễn Văn Thuấn**

Mã Sinh viên: 10115275

Hưng Yên – 2018

Mục lục

[Chương 1: TỔNG QUAN 4](#_Toc517008299)

[1.1 Lý do chọn đề tài 4](#_Toc517008300)

[1.2 Mục tiêu 4](#_Toc517008301)

[1.3 Phương pháp nghiên cứu 4](#_Toc517008302)

[Chương 2: CƠ SỞ LÝ THUYẾT 5](#_Toc517008303)

[2.1 Các linh kiện chính 5](#_Toc517008304)

[2.2 Tìm hiểu về cách hoạt động của quang trở vào servo trong đề tài 5](#_Toc517008305)

[2.2.1 Quang trở 5](#_Toc517008306)

[2.2.2 Servo 6](#_Toc517008307)

[2.2.3 Arduino UNO R3 7](#_Toc517008308)

[Chương 3: NỘI DUNG THỰC HIỆN 10](#_Toc517008309)

[3.1 Sơ đồ nối dây trong proteus 10](#_Toc517008310)

[3.2 Lưu đồ thuật toán 10](#_Toc517008311)

[3.2.1 Nguyên lý hoạt động 10](#_Toc517008312)

[3.2.2 Lưu đồ thuật toán chương trình chính 11](#_Toc517008313)

[3.3 Sản phẩm thực hiện 12](#_Toc517008314)

[3.4 Minh họa sản phẩm 12](#_Toc517008315)

[Chương 4: KẾT LUẬN 15](#_Toc517008316)

[4.1 Kết quả đạt được 15](#_Toc517008317)

[4.2 Hạn chế 15](#_Toc517008318)

[4.3 Hướng phát triển 15](#_Toc517008319)

[Tài liệu tham khảo 16](#_Toc517008320)

[PHỤ LỤC 17](#_Toc517008321)

Danh mục hình ảnh

[Hình 2.1 Cách mắc quang trở với uno R3 5](#_Toc517008291)

[Hình 2.2 Cách mắc nối servo với uno R3 6](#_Toc517008292)

[Hình 2.3 Minh họa uno R3 7](#_Toc517008293)

[Hình 2.4 Minh họa trình biên dịch 9](#_Toc517008294)

[Hình 3.1 Sơ đồ nối dây trong proteus 10](#_Toc517008295)

[Hình 3.2 Mặt ngang của sản phẩm 12](#_Toc517008296)

[Hình 3.3 Hệ thống quang trở 13](#_Toc517008297)

[Hình 3.4 Mặt trước sản phẩm 14](#_Toc517008298)

Chương 1: TỔNG QUAN

1.1 Lý do chọn đề tài

Như chúng ta biết các thiết bị ứng dụng năng lượng mặt trời cung cấp năng lượng nhiệt và điện đang phát triển mạnh mẽ trên thế giới trong nhiều năm gần đây. Để thu được nguồn năng lượng cao nhất có thể, đã có nhiều nghiên cứu, sản phẩm cơ cấu định hướng theo vị trí mặt trời.

Ở Việt Nam, cơ cấu định hướng mặt thu theo vị trí mặt trời chưa được quan tâm nhiều, vì vậy từ những xu hướng phát triển và nhu cầu thực tế nên em đã chọn đề tài:”*Hệ thống định vị ánh sáng mặt trời ứng dụng trong pin mặt trời*”

1.2 Mục tiêu

Nghiên cứu hệ thống tự điều chỉnh góc quay bề mặt thu năng lượng mặt trời theo vị trí mặt trời/nguồn sáng để thu được tối đa năng lượng có thể.

Thiết kế, chế tạo hệ thống tự động điều chỉnh bề mặt thu của thiết bị pin mặt trời nhỏ và vừa.

1.3 Phương pháp nghiên cứu

Dựa theo các công trình, sản phẩm đã có, phân tích - chọn công nghệ phù hợp với điều kiện của bản thân cũng như tính ứng dụng có thể.

Chương 2: CƠ SỞ LÝ THUYẾT

2.1 Các linh kiện chính

- Arduino UNO R3

- Quang trở

- micro Servo tower pro 9g

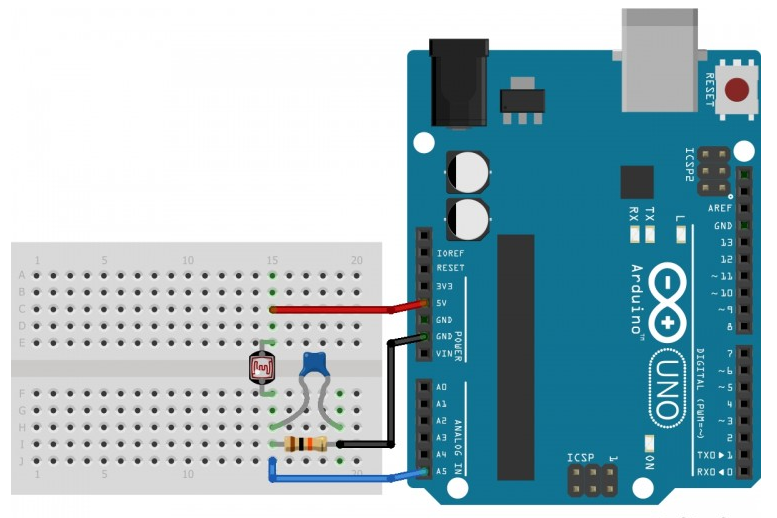
-Điện trở 10k …

2.2 Tìm hiểu về cách hoạt động của quang trở vào servo trong đề tài

2.2.1 Quang trở

Quang trở là một loại "vật liệu" điện tử rất hay gặp và được sử dụng trong những mạch cảm biến ánh sáng[**.**](http://arduino.vn/bai-viet/208-cach-doc-du-lieu-tu-quang-tro-va-xay-dung-cam-bien-anh-sang) Có thể hiểu một cách dễ dàng rằng, quang trở là một loại ĐIỆN TRỞ có điện trở thay đổi theo cường độ ánh sáng. Nếu đặt ở môi trường có ít ánh sáng, có bóng râm hoặc tối thì điện trở của quang trở sẽ tăng cao còn nếu đặt ở ngoài nắng, hoặc nơi có ánh sáng thì điện trở sẽ giảm.

Đây là phương pháp có tên gọi là đọc hiệu điến thế từ cầu phân áp, để xem về cầu phân áp (gợi ý, quang trở bạn cứ xem là điện trở R1, vào phía trước mỗi chân Analog của Arduino có một điện trở, và xem đó là điện trở R2 = 10k).

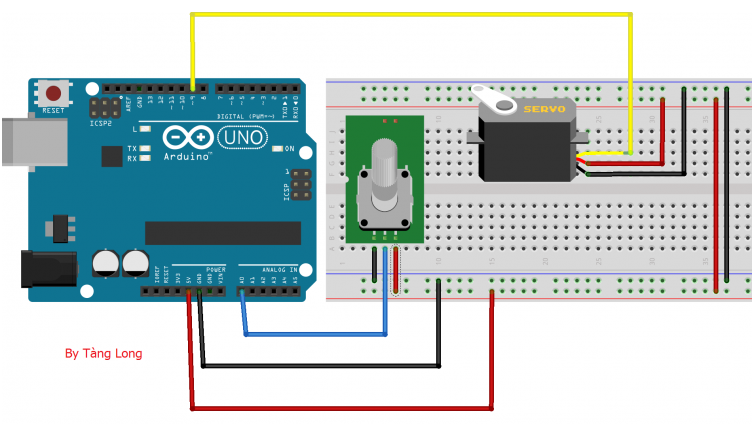


Hình 2.1 Cách mắc quang trở với uno R3

2.2.2 Servo

-Khái niệm: **Servo** là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng [xung PPM](http://arduino.vn/reference/xung-ppm)) với góc quay nằm trong khoảng bất kì từ 0 độ - 180 độ. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô mình), có loại thì sở hữu một momen lực bá đạo (vài chục Newton/m), hoặc có loại thì khỏe và nhông sắc chắc chắn,...

Động cơ servo được thiết kế những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bầt kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác. Các động cơ servo điều khiển bằng liên lạc vô tuyến được gọi là động cơ servo RC (radio-controlled). Trong thực tế, bản thân động cơ servo không phải được điều khiển bằng vô tuyến, nó chỉ nối với máy thu vô tuyến trên máy bay hay xe hơi. Động cơ servo nhận tín hiệu từ máy thu này.



Hình 2.2 Cách mắc nối servo với uno R3

2.2.3 Arduino UNO R3

Arduino UNO R3 là dòng mạch dung để lập trình đã phát triển đến thế hệ thứ 3 (R3). Arduino UNO có thể sử dụng 3 vi điều khiển họ 8bit AVR là ATmega8, ATmega168, ATmega328. Bộ não này có thể xử lí những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lí tín hiệu cho xe điều khiển từ xa, làm một trạm đo nhiệt độ - độ ẩm và hiển thị lên màn hình LCD,… hay những ứng dụng khác.

Dưới đây là một vài thông số của Arduino UNO R3

|  |  |
| --- | --- |
| Vi điều khiên | ATmega họ 8bit |
| Điện áp hoạt động | 5V DC ( với cổng USB) |
| Tần số hoạt động | 16 MHz |
| Dòng tiêu thụ | Khoảng 30mA |
| Điện áp khuyên dung | 7 – 12V DC |
| Điện áp giới hạn | 6 – 20V DC |
| Số chân I/O | 14 (6 chân hardware PWM) |
| Số chân Ânlog | 6 ( độ phân giản 10bit) |
| Dòng tối đa trên mỗi chân I/O | 30mA |
| Dòng ra tối đa(5V) | 500mA |
| Dòng ra tối đa (3.3V) | 50mA |
| Bộ nhớ flash | 32KB (ATmega328) với 0,5KB dung bởi bootloader |
| SRAM | 2KB (ATmega328) |
| EEPROM | 1KB (ATmega328) |



Hình 2.3 Minh họa uno R3

**Các chân năng lượng**

+ GND (Ground): cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.

+ 5V: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.

+ 3.3V: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.

+ Vin (Voltage Input): để cấp nguồn ngoài cho Arduino UNO, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.

+ IOREF: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy bạn không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.

+ RESET: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

Lưu ý:

Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó bạn phải hết sức cẩn thận, kiểm tra các cực âm – dương của nguồn trước khi cấp cho Arduino UNO. Việc làm chập mạch nguồn vào của Arduino UNO sẽ biến nó thành một miếng nhựa chặn giấy. mình khuyên bạn nên dùng nguồn từ cổng USB nếu có thể.

Các chân 3.3V và 5V trên Arduino là các chân dùng để cấp nguồn ra cho các thiết bị khác, không phải là các chân cấp nguồn vào. Việc cấp nguồn sai vị trí có thể làm hỏng board. Điều này không được nhà sản xuất khuyến khích.

Cấp nguồn ngoài không qua cổng USB cho Arduino UNO với điện áp dưới 6V có thể làm hỏng board.

Cấp điện áp trên 13V vào chân RESET trên board có thể làm hỏng vi điều khiển ATmega328.

Cường độ dòng điện vào/ra ở tất cả các chân Digital và Analog của Arduino UNO nếu vượt quá 200mA sẽ làm hỏng vi điều khiển.

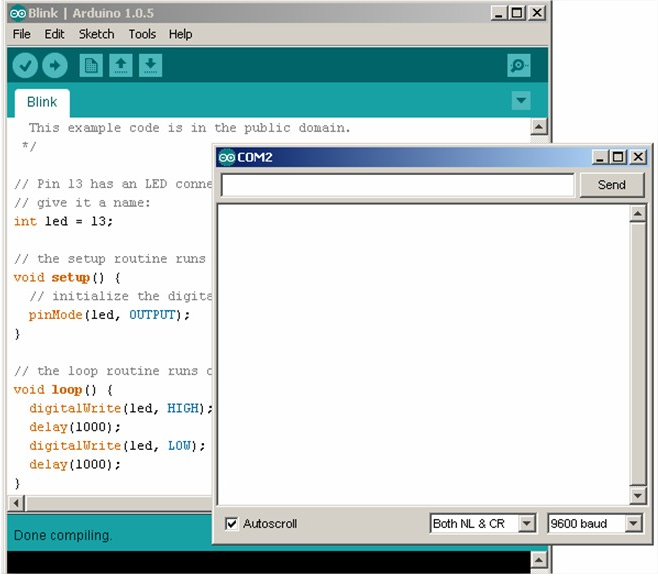
Cấp điệp áp trên 5.5V vào các chân Digital hoặc Analog của Arduino UNO sẽ làm hỏng vi điều khiển.

Cường độ dòng điện qua một chân Digital hoặc Analog bất kì của Arduino UNO vượt quá 40mA sẽ làm hỏng vi điều khiển. Do đó nếu không dùng để truyền nhận dữ liệu, bạn phải mắc một điện trở hạn dòng.

**Lập trình cho Arduino**

Các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung. Và Wiring lại là một biến thể của C/C++. Một số người gọi nó là Wiring, một số khác thì gọi là C hay C/C++. Riêng mình thì gọi nó là “ngôn ngữ Arduino”, và đội ngũ phát triển Arduino cũng gọi như vậy. Ngôn ngữ Arduino bắt nguồn từ C/C++ phổ biến hiện nay do đó rất dễ học, dễ hiểu. Nếu học tốt chương trình Tin học 11 thì việc lập trình Arduino sẽ rất dễ thở đối với bạn.

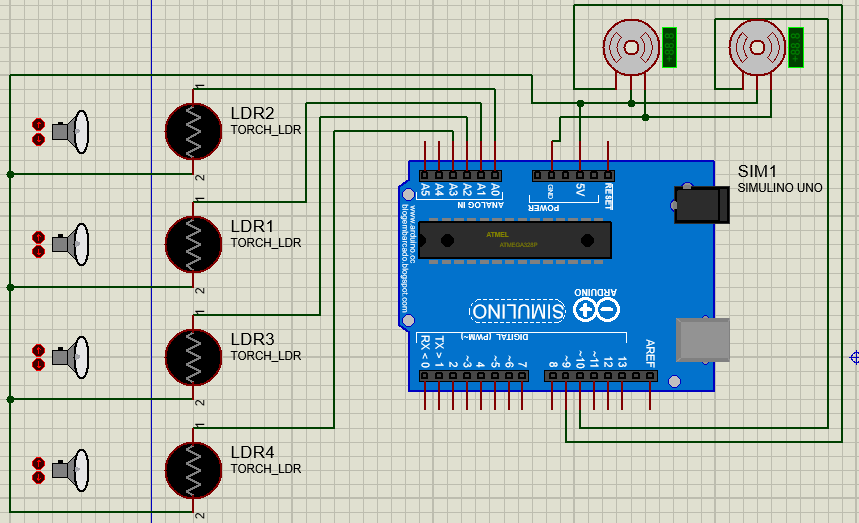
Để lập trình cũng như gửi lệnh và nhận tín hiệu từ mạch Arduino, nhóm phát triển dự án này đã cũng cấp đến cho người dùng một môi trường lập trình Arduino được gọi là Arduino IDE (**I**ntergrated **D**evelopment **E**nvironment) như hình dưới đây.



Hình 2.4 Minh họa trình biên dịch

Chương 3: NỘI DUNG THỰC HIỆN

3.1 Sơ đồ nối dây trong proteus



Hình 3.1 Sơ đồ nối dây trong proteus

3.2 Lưu đồ thuật toán

3.2.1 Nguyên lý hoạt động

Quang trở nhận tín hiệu ánh sáng rồi gửi về Arduino. Arduino sẽ đọc tín hiệu dưới dạng Analog và xử lí tính toán nơi có cường độ sáng lớn nhất sau đó sẽ gửi tín hiệu cho 2 servo quay ra hướng đó (2 servo xếp chồng lên nhau sẽ có thể quay được tối đa 359 độ).

Việc tính toán góc quay dựa theo 4 quang trở được thiết kế như trên: ta tính hiệu số của từng cặp quang trở theo các hướng trên – dưới – trái – phải để tìm ra hướng có cường độ ánh sáng lớn nhất.

3.2.2 Lưu đồ thuật toán chương trình chính

Xử lý + điều khiển

main

Khai báo quang trở

tr trở

Khai báo servo

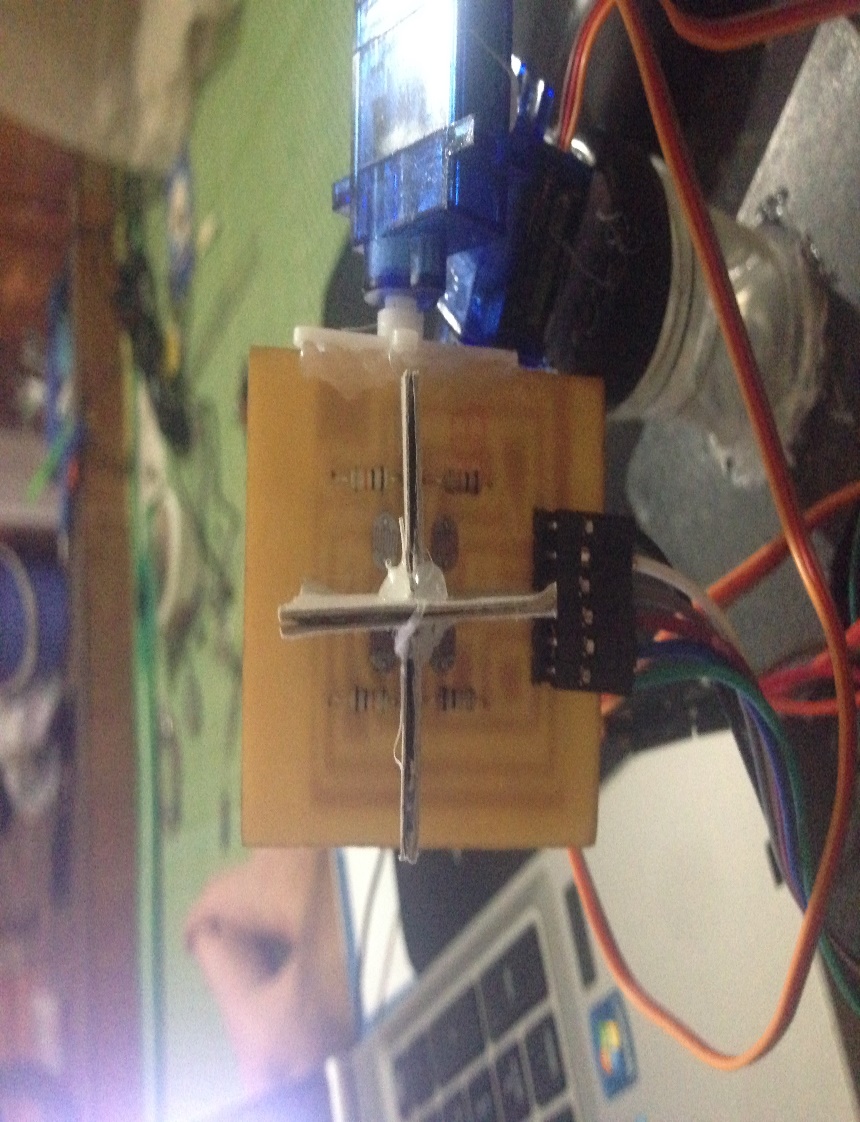
3.3 Sản phẩm thực hiện

Sản phẩm sau khi hoàn thành có thể nhận biết hướng sáng và quay theo hướng có ánh sáng mạnh nhất.

3.4 Minh họa sản phẩm



Hình 3.2 Mặt ngang của sản phẩm



Hình 3.3 Hệ thống quang trở



Hình 3.4 Mặt trước sản phẩm

Chương 4: KẾT LUẬN

4.1 Kết quả đạt được

Sau thời gian nghiên cứu và phát triển, em đã hoàn thành cơ bản các mục tiêu ban đầu đề ra của đề tài. Sản phẩm có tính ứng dụng cao, có thể tiếp tục phát triển thêm các tính năng hữu ích khác.

Hệ thống được thiết kế rất gọn nhẹ,tiện lời. Có tính mở, có thể tiếp tục cập nhật các chức năng khá dễ dàng.

4.2 Hạn chế

Do điều kiện thời gian cũng như kinh phí nên các linh kiện sử dụng không phải là loại có chất lượng tốt nhất, khó đảm bảo hoạt động dưới mưa nắng trong thời gian dài.

Thiết kế sản phẩm có phần chưa hợp lý

4.3 Hướng phát triển

Có thể tích hợp bộ sạc, bộ chuyển đổi điện năng trong của pin mặt trời như bình acquy, pin với bộ tự động nạp vừa có nguồn nuôi cho chính hệ thống vừa có nguồn điện cung cấp cho các thiết bị khác.

Nâng cấp bằng các linh kiện có chất lương tốt hơn cũng như cải thiện việc hoạt động thời gian dài dưới điều kiên thời tiết khắc nghiệt.

Tài liệu tham khảo

<https://www.youtube.com/watch?v=vfc6TxTiFUQ&t=46s>

<http://arduino.vn/bai-viet/42-arduino-uno-r3-la-gi>

<http://arduino.vn/bai-viet/208-cach-doc-du-lieu-tu-quang-tro-va-xay-dung-cam-bien-anh-sang>

<http://arduino.vn/bai-viet/181-gioi-thieu-servo-sg90-va-cach-dieu-khien-bang-bien-tro>

PHỤ LỤC

// code cua chuong trinh

#include <Servo.h> // Thư viện điều khiển servo

Servo Ser1;

Servo Ser2;

int servoPin = 9;

int servoPin1 = 10;

int dem=90;

int dem1=90;

int bugiat = 30;

void setup()

{

pinMode(A0,INPUT);

pinMode(A1,INPUT);

pinMode(A2,INPUT);

pinMode(A3,INPUT);

Ser1.attach(servoPin);

Ser2.attach(servoPin1);

Ser1.write(90);

Ser2.write(90);

delay(5000);

Serial.begin(9600);

}

void loop()

{

int a = analogRead(A0);

int b = (analogRead(A1)+20)\*3;

int c = analogRead(A2);

int d = analogRead(A3);

// delay(100);

Ser1.write(dem);

Ser2.write(dem1);

Serial.print("\nA0= ");

Serial.print(a);

Serial.print("\tA1= ");

Serial.print(b);

Serial.print("\tA2= ");

Serial.print(c);

Serial.print("\tA3= ");

Serial.println(d);

//servo 1

int sa = a + b;

int sb = c + d;

Serial.print("sa: "); Serial.println(sa);

Serial.print("sb: "); Serial.println(sb);

Serial.println("");

if(sa-sb > bugiat || sb-sa >bugiat)

{

if(sa > sb)

{

dem = dem + 1;

if(dem >= 179)

{

dem = 179;

}

}

if(sa < sb)

{

dem = dem - 1;

if(dem <= 0)

{

dem = 0;

}

}

}

// servo 2

int sa2 = a + c;

int sb2 = b + d;

Serial.print("sa2: "); Serial.println(sa2);

Serial.print("sb2: "); Serial.println(sb2);

Serial.println("");

if(sa2-sb2 > bugiat || sb2-sa2 > bugiat )

{

if(sa2 > sb2)

{

Serial.println("aaaaaa");

dem1 = dem1+1;

if(dem1 >= 179)

{

dem1 = 179;

}

}

if(sa2 < sb2)

{

Serial.println("bbbbbb");

dem1 = dem1 + 1;

if(dem1 <= 0)

{

dem1 = 0;

}

}

}

Serial.print("\tdem= ");

Serial.print(dem);

Serial.print("\tdem1= ");

Serial.print(dem1);

Ser1.write(dem);

Ser2.write(dem1);

delay(50);

}