

## **BÁO CÁO TIỂU LUẬN**

# **THIẾT KẾ VÀ CHẾ TẠO ROBOT CÂN BẰNG BÓNG TRÊN MẶT PHẪNG**

**NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA**

**GIẢNG VIÊN HƯỚNG DẪN: ThS. PHẠM NGUYỄN NHỰT THANH**

<b>Sinh viên thực hiện:</b>	<b>MSSV:</b>	<b>Lớp:</b>
Nguyễn Văn Đạt	2286300010	22DRTA1
Huỳnh Long	2286300028	22DRTA1
Nguyễn Chân Huy	2286300020	22DRTA1
Lê Minh Nghĩa	2286300030	22DRTA1

**TP. Hồ Chí Minh, năm 2023**

# LỜI CẢM ƠN

Sau một thời gian nỗ lực tìm hiểu và thực hiện đề tài "**Thiết kế và chế tạo Robot cân bằng bóng trên mặt phẳng**", em rất vui mừng vì đã hoàn thành báo cáo này, dù vẫn còn một số phần cần hoàn thiện thêm trong tương lai. Quá trình thực hiện đề tài đã giúp em tích lũy được nhiều kiến thức và kinh nghiệm quý báu trong lĩnh vực điện tử và lập trình điều khiển PID.

Em xin gửi lời cảm ơn chân thành đến **Thầy Phạm Nguyễn Nhựt Thanh**, người đã hướng dẫn chúng em trong các buổi học. Tuy Thầy không trực tiếp hướng dẫn đề tài này, nhưng những kiến thức từ các buổi học của Thầy đã giúp nhóm em rất nhiều trong việc tính toán và tìm hiểu sâu hơn về đề tài.

Em cũng xin gửi lời cảm ơn đến bạn **Nguyễn Thanh Nhân**, người đã giúp đỡ nhóm một phần quan trọng trong việc tìm hiểu và nghiên cứu đề tài này.

Ngoài ra, em cũng muốn bày tỏ lòng biết ơn đến các thành viên trong nhóm, những người đã cùng em làm việc chăm chỉ, chia sẻ kiến thức và hỗ trợ lẫn nhau trong quá trình thực hiện đề tài. Sự đoàn kết và nỗ lực của từng thành viên đã góp phần quan trọng vào sự thành công của dự án.

Em cũng không thể không nhắc đến những công cụ, phần mềm hỗ trợ trong quá trình nghiên cứu và thực hiện đề tài. Những công cụ này đã giúp nhóm em dễ dàng hơn trong việc lập trình, điều chỉnh và thử nghiệm robot.

Mặc dù còn một số hạn chế trong báo cáo này, em hy vọng sẽ nhận được những góp ý quý báu từ Thầy để hoàn thiện hơn trong những lần sau. Một lần nữa, em xin chân thành cảm ơn!

# MỤC LỤC

<b>CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI</b>	<b>5</b>
1.1 Lý do chọn đề tài	5
1.2 Mục đích chọn đề tài	6
1.3 Sơ lược các bước thực hiện	6
<b>CHƯƠNG 2: THIẾT KẾ PHẦN CỨNG</b>	<b>7</b>
1.4 Cấu trúc tổng thể	7
1. Vi điều khiển:	8
2. Cấu trúc và cơ chế hoạt động:	8
3. Nguồn năng lượng:	8
1.5 Sơ đồ kết nối và mạch PCB	8
1.6 Danh sách linh kiện	9
<b>CHƯƠNG 3: THIẾT KẾ PHẦN MỀM</b>	<b>12</b>
1.7 Ý tưởng lập trình và lưu đồ giải thuật	12
1.7.1 Ý tưởng lập trình	12
1.8 Lập trình cho cho Robot	13
1.8.1 Sử dụng Arduino và PCA9685 để điều khiển các servo	13
1.8.2 Điều Khiển Servo Cho Robot Cân Bằng Bóng	14
<b>CHƯƠNG 4 :TỔNG HỢP HỆ THỐNG</b>	<b>17</b>
1.9 Tổng hợp mô hình	17
1.10 Thực nghiệm hệ thống	19
<b>CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN</b>	<b>23</b>
1.11 Kết quả đạt được	23
1.12 Các hạn chế	23
1.13 Phương hướng phát triển trong tương lai	23

# MỤC LỤC HÌNH ẢNH

<b>Hình 2.1:</b> Cấu trúc tổng thể.	3
<b>Hình 2.2:</b> Sơ đồ kết nối.	3
<b>Hình 2.3:</b> Arduino MEGA2560 R3 CH340 cổng Type-C.	4
<b>Hình 2.4:</b> Servo MG996R bánh răng kim loại.	5
<b>Hình 3.1:</b> Hai mặt phẳng với 6 thanh giằng giúp nâng lên hạ xuống để đẩy bóng về vị trí cân bằng.	6
<b>Hình 3.2:</b> Lưu đồ giải thuật theo sơ đồ khối cho đề tài.	7
<b>Hình 3.3:</b> Chọn board.	8
<b>Hình 4.1:</b> Mô hình khi gắn các linh kiện.	10
<b>Hình 4.2:</b> Mô hình sau khi được gắn linh kiện.	10
<b>Hình 4.3:</b> Robot sau khi hoàn thành phần cứng.	11
<b>Hình 4.4:</b> Khởi động Robot.	11
<b>Hình 4.5:</b> Robot cân bằng bóng trên mặt phẳng.	12
<b>Hình 4.6:</b> Robot điều chỉnh khi bóng lệch khỏi trung tâm.	12
<b>Hình 4.7:</b> Robot trả về trạng thái trung lập sau khi cân bằng thành công.	13

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## 1.1 Lý do chọn đề tài

Trong giai đoạn hiện nay, sự phát triển mạnh mẽ của công nghệ và kỹ thuật đã đem lại nhiều tiến bộ vượt bậc, đặc biệt là trong lĩnh vực robot và tự động hóa. Robot không chỉ tham gia vào các quy trình sản xuất công nghiệp mà còn góp phần làm tăng năng suất, giảm thiểu sai sót, và đảm bảo an toàn trong các môi trường làm việc nguy hiểm như hàn, phun sơn, lắp ráp linh kiện điện tử, và thậm chí là các nhiệm vụ ngoài vũ trụ.

Trong số các loại robot, robot cân bằng là một thách thức kỹ thuật đáng chú ý, đòi hỏi sự kết hợp giữa nhiều lĩnh vực như cơ học, điện tử, và lập trình điều khiển. Khả năng cân bằng của robot trên một mặt phẳng là một bài toán thú vị, thể hiện sự phức tạp trong việc xử lý tín hiệu cảm biến, điều khiển động cơ và duy trì trạng thái ổn định trong điều kiện thực tế.

Với sự phát triển của ngành robot học, việc nghiên cứu và phát triển các hệ thống cân bằng tự động không chỉ giúp nâng cao kiến thức về điều khiển PID và các phương pháp điều khiển tiên tiến khác mà còn mở ra nhiều ứng dụng trong thực tế. Ví dụ, các hệ thống cân bằng này có thể được áp dụng trong việc phát triển các phương tiện tự hành, các thiết bị ổn định trong công nghiệp, hay thậm chí là trong các công cụ hỗ trợ trong lĩnh vực y tế.

Vì những lý do trên, nhóm chúng em đã quyết định chọn đề tài "Thiết kế và chế tạo robot cân bằng bóng trên mặt phẳng" với mong muốn không chỉ áp dụng các kiến thức đã học vào thực tế mà còn góp phần vào việc nghiên cứu, phát triển các hệ thống cân bằng tự động, đáp ứng nhu cầu ngày càng cao của xã hội trong việc tự động hóa và tối ưu hóa các quy trình công nghệ.

## **1.2 Mục đích chọn đề tài**

Mục tiêu của đề tài là thiết kế, chế tạo, và điều khiển robot cân bằng bóng trên mặt phẳng. Robot này cần hoạt động ổn định, duy trì trạng thái cân bằng của quả bóng trên một mặt phẳng bằng cách điều chỉnh vị trí của các động cơ theo thời gian thực. Đề tài này không chỉ có tính ứng dụng cao trong thực tế mà còn tạo điều kiện cho sinh viên vận dụng những kiến thức đã học trong lĩnh vực điều khiển tự động.

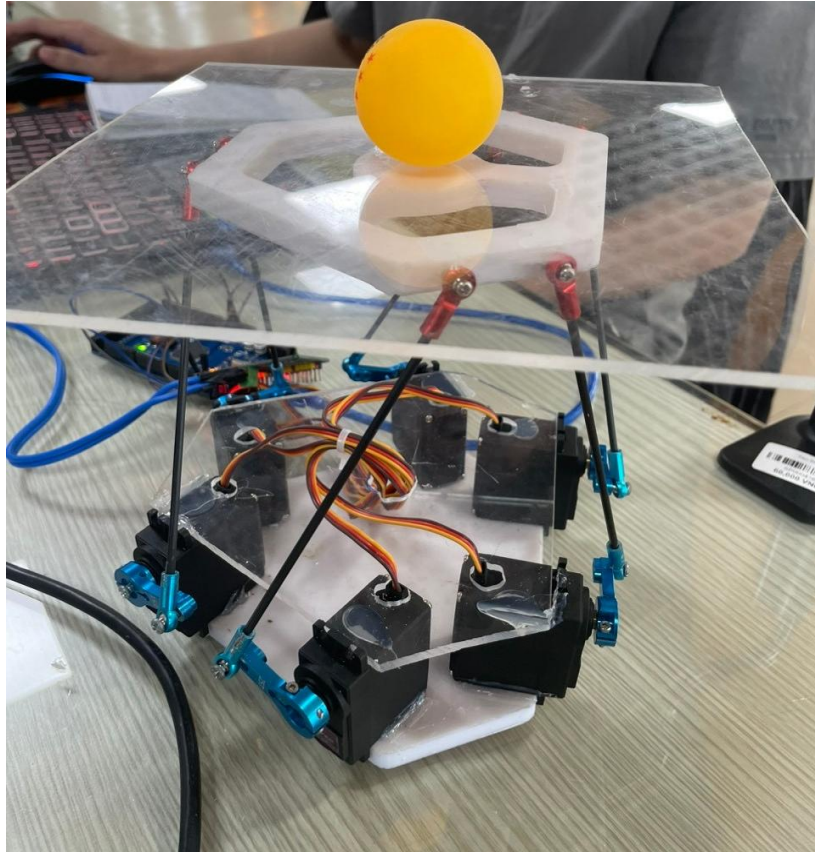
Với thiết kế cơ khí đơn giản nhưng linh hoạt, kết hợp với các thành phần điện tử như vi điều khiển, cảm biến độ nghiêng, và các động cơ servo, robot cân bằng bóng là một hệ thống lý tưởng để sinh viên học tập và nghiên cứu sâu hơn về các nguyên lý điều khiển PID cũng như các kỹ thuật điều khiển tự động khác. Đề tài này cung cấp một nền tảng thực hành tốt, giúp sinh viên có cơ hội áp dụng lý thuyết vào thực tế và phát triển kỹ năng giải quyết vấn đề.

## **1.3 Sơ lược các bước thực hiện**

Đầu tiên, cần thiết kế và chế tạo khung của robot. Khung này phải đảm bảo độ bền và độ chính xác nhất định trong việc bố trí các động cơ servo và cảm biến. Sau khi hoàn thiện phần cơ khí, bước tiếp theo là lựa chọn vi điều khiển, khối xử lý trung tâm của robot, và các mạch điện tử cần thiết cho việc điều khiển hệ thống. Cuối cùng, tiến hành lập trình hệ thống điều khiển dựa trên những kiến thức về điều khiển PID và các nguyên lý điều khiển đã học được để robot có thể duy trì trạng thái cân bằng của quả bóng trên mặt phẳng.

## CHƯƠNG 2: THIẾT KẾ PHẦN CỨNG

### 1.4 Cấu trúc tổng thể



Hình 2.1: Cấu trúc tổng thể.

- **Kích thước tổng thể của hệ thống:**
  - **Chiều dài:** 15 cm
  - **Chiều rộng:** 15 cm
  - **Chiều cao:** 18 cm
  - **Chiều dài camera đến mặt phẳng :** Tùy chỉnh sao cho khớp với khung hình ( 30 – 35 cm )
  - **Bóng bàn với kích thước đường kính 40mm**

## 1. Vi điều khiển:

- **Vi điều khiển:** Arduino Mega
  - Sử dụng Arduino Mega để điều khiển toàn bộ hệ thống.
  - Kết hợp với mạch điều khiển 16 kênh PWM PCA9685 để điều khiển 6 servo motor, giúp điều chỉnh độ nghiêng của mặt phẳng.

## 2. Cấu trúc và cơ chế hoạt động:

- **Cơ chế điều khiển:**
  - 6 servo motor điều chỉnh độ nghiêng của mặt phẳng theo các trục x, y để giữ quả bóng cân bằng.
  - Camera kết hợp với laptop được sử dụng để nhận diện và theo dõi vị trí của quả bóng trên mặt phẳng.
- **Phương pháp nhận diện quả bóng:**
  - Sử dụng YOLOv8 kết hợp với Python trên laptop để mở camera và xác định tâm của quả bóng cũng như vị trí hiện tại của mặt phẳng.
  - Sau khi xác định được vị trí, hệ thống sẽ thực hiện thuật toán điều khiển để di chuyển quả bóng về trung tâm của mặt phẳng và duy trì sự cân bằng.

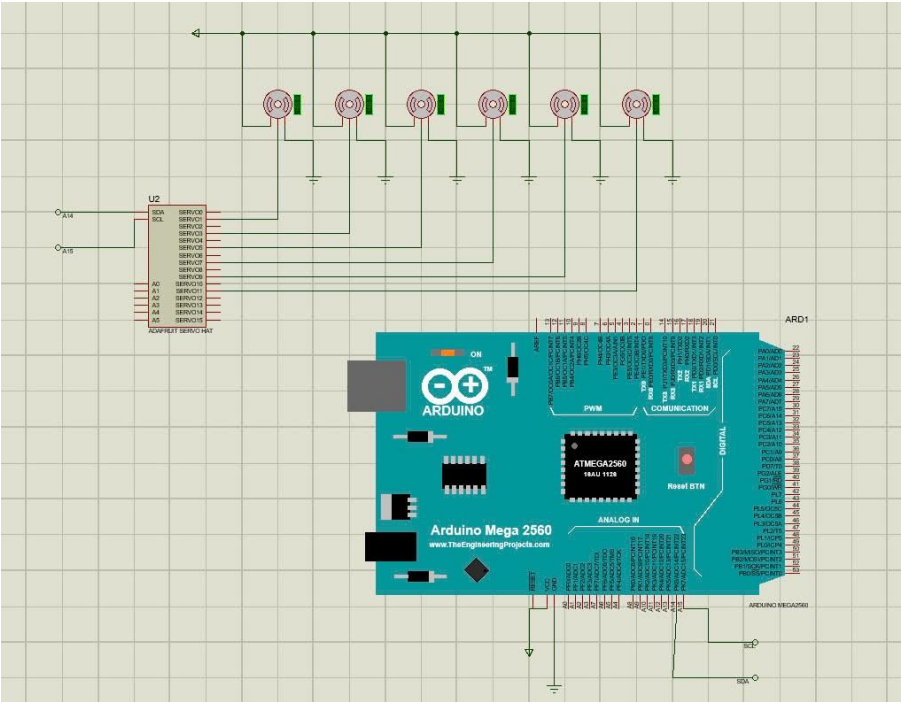
## 3. Nguồn năng lượng:

- **Nguồn tổ ong:**
  - 5V, 20A để cung cấp năng lượng cho hệ thống điều khiển và servo.
- **Laptop:**
  - Sử dụng để nhận diện bóng qua camera bằng cách chạy các đoạn mã Python.

## 1.5 Sơ đồ kết nối và mạch PCB

Sơ đồ kết nối chân các linh kiện:



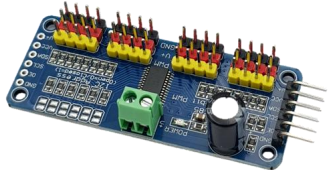



Hình 2.2: Sơ đồ kết nối.

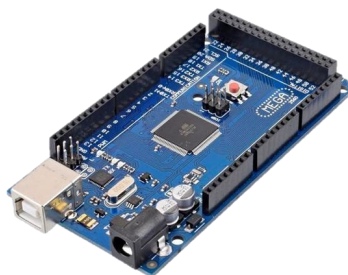
1.6 Danh sách linh kiện

Bảng 2.1: Danh sách linh kiện

STT	Tên	Hình ảnh
1	Arduino MEGA2560 R3 CH340 cổng Type-C	

2	Mạch điều khiển 16 Chanel PWM PCA9685	
3	Servo MG996R bánh răng kim loại X6	

Vi điều khiển Arduino MEGA2560 R3 CH340 cổng Type-C:



**Hình 2.3:** Arduino MEGA2560 R3 CH340 cổng Type-C.

Arduino MEGA2560 R3 CH340 với cổng Type-C là một trong những bo mạch vi điều khiển phổ biến và mạnh mẽ trong hệ sinh thái của Arduino. Được thiết kế dựa trên vi điều khiển ATmega2560, nó cung cấp một loạt các tính năng và khả năng cho các ứng dụng nhúng và điều khiển phức tạp. Arduino MEGA2560 R3 CH340 có 54 chân GPIO (General Purpose Input/Output), trong đó có 15 chân hỗ trợ PWM, mang lại sự linh hoạt trong kết nối và điều khiển nhiều loại thiết bị khác nhau.

Servo MG996R bánh răng kim loại:



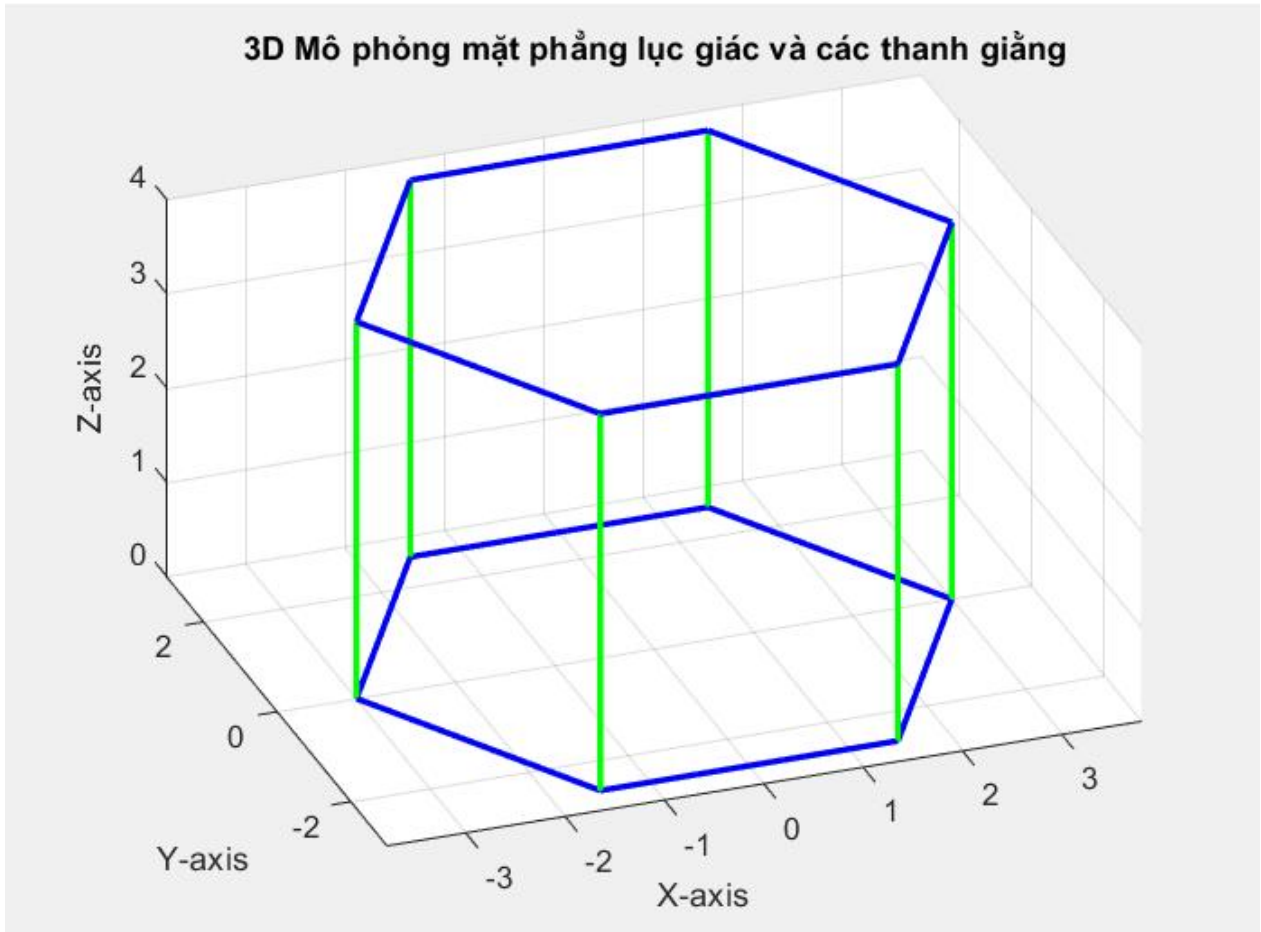
**Hình 2.4:** Servo MG996R bánh răng kim loại.

Servo MG996R là một loại servo mạnh mẽ và phổ biến, được thiết kế với bánh răng kim loại bền bỉ, giúp tăng độ bền và khả năng chịu tải cao. Với mô-men xoắn lên đến 9.4 kg.cm (ở 6V) và tốc độ phản hồi nhanh, MG996R thường được sử dụng trong các ứng dụng yêu cầu độ chính xác và lực mạnh như robot, cánh tay cơ học, và các dự án điều khiển từ xa. Servo này hỗ trợ điều khiển vị trí góc quay từ 0 đến 180 độ, mang lại sự linh hoạt và chính xác trong việc điều khiển các cơ cấu chuyển động..

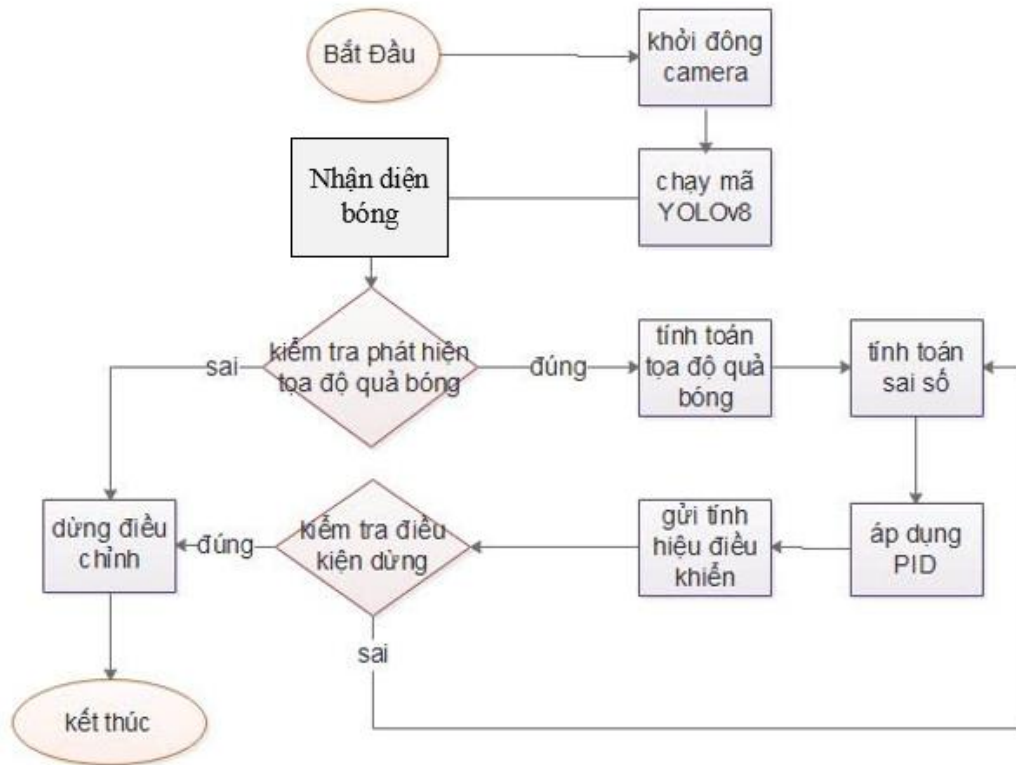
## CHƯƠNG 3: THIẾT KẾ PHẦN MỀM

### 1.7 Ý tưởng lập trình và lưu đồ giải thuật

#### 1.7.1 Ý tưởng lập trình



**Hình 3.1:** Hai mặt phẳng với 6 thanh giằng giúp nâng lên hạ xuống để đẩy bóng về vị trí cân bằng.

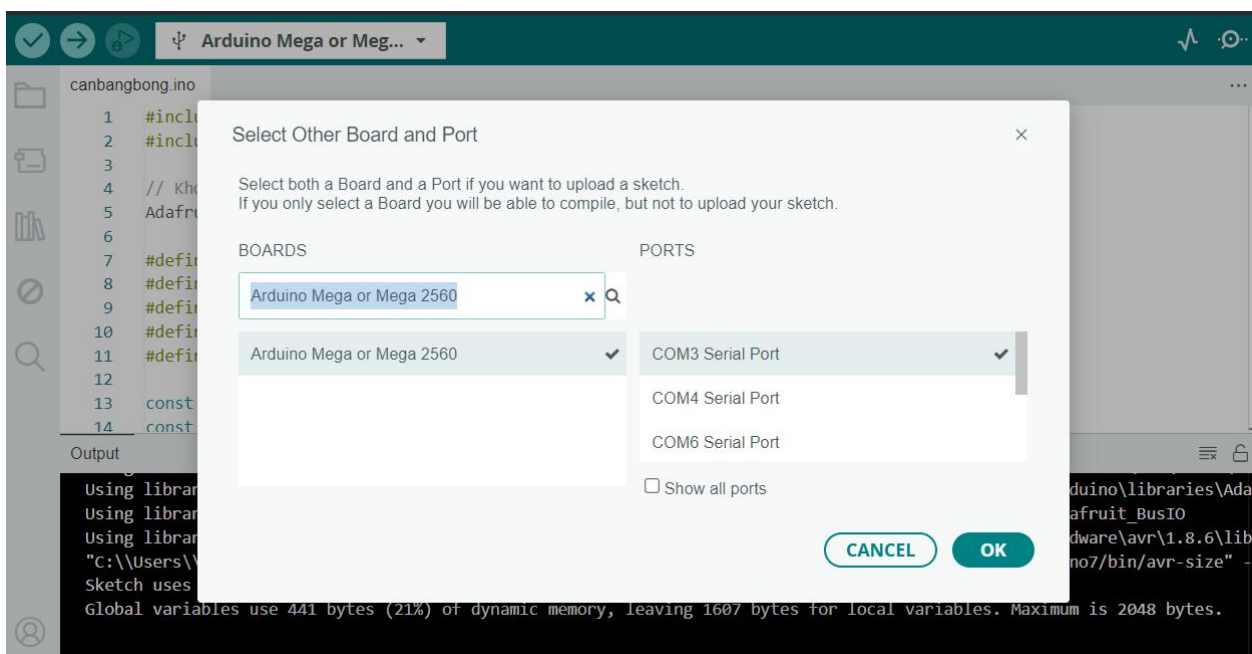


**Hình 3.2:** Lưu đồ giải thuật theo sơ đồ khối cho đề tài.

## 1.8 Lập trình cho cho Robot

### 1.8.1 Sử dụng Arduino và PCA9685 để điều khiển các servo

Đối tượng lập trình là Arduino Mega, một vi điều khiển mạnh mẽ với nhiều chân I/O, giúp điều khiển các thiết bị ngoại vi và xử lý tín hiệu một cách hiệu quả. Môi trường lập trình khuyến nghị là Arduino IDE 2.1.0. So với các phiên bản trước, Arduino IDE 2.1.0 có nhiều cải tiến trong khả năng biên dịch, giúp lập trình linh hoạt cho nhiều dòng vi điều khiển khác nhau, trong đó có Arduino Mega. Để lập trình cho Arduino Mega trên Arduino IDE 2.1.0, trước tiên cần chọn board phù hợp và cổng COM, điều này phụ thuộc vào máy tính của mỗi người.



**Hình 3.3:** Chọn board.

Mặc định, Arduino IDE sẽ nạp chương trình cho Arduino thông qua cổng COM với tốc độ cao nhất mà Arduino Mega hỗ trợ là 115200 bps. Tuy nhiên, đôi khi nạp chương trình có thể không đáp ứng được tốc độ này, vì vậy để dễ dàng nạp chương trình cho Arduino Mega, người dùng có thể giảm tốc độ nạp xuống cho đến khi nạp thành công

### 1.8.2 Điều Khiển Servo Cho Robot Cân Bằng Bóng

Robot cân bằng bóng sử dụng sáu servo để điều chỉnh mặt phẳng nhằm giữ quả bóng ở vị trí trung tâm. Các servo được điều khiển thông qua mạch PCA9685, một mạch điều khiển PWM 16 kênh, cho phép điều khiển độc lập nhiều servo cùng lúc. Mạch PCA9685 hỗ trợ tần số PWM tối đa là 60 Hz, phù hợp với yêu cầu điều khiển chính xác vị trí của các servo.

```

#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

// Khởi tạo PCA9685 tại địa chỉ mặc định (0x40)
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);

#define SERVOMIN 150 // Xung tối thiểu (servo về 0 độ)
#define SERVOMAX 635 // Xung tối đa (servo về 180 độ)
#define NEUTRAL 381 // Vị trí trung lập của servo
#define OFFSET0 30 // Offset cho servo 0
#define OFFSET5 30 // Offset cho servo 5

const int x0 = 320;
const int y0 = 240;
const int maxVector = 275;
const int maxServoDiff = 50;

void setup() {
  Serial.begin(921600); // Khởi tạo Serial với tốc độ baud rate cao
  pwm.begin();
  pwm.setPWMFreq(60); // Tần số PWM 60 Hz
  delay(10);
}

```

Trong mã nguồn, các giá trị xung PWM tương ứng với các góc từ 0 đến 180 độ được định nghĩa qua các hằng số như SERVOMIN, SERVOMAX, và NEUTRAL. Qua đó, ta có thể điều chỉnh vị trí của servo bằng cách thay đổi độ rộng xung PWM; độ rộng xung càng lớn thì góc servo càng lớn. Các chân điều khiển sẽ quy định chế độ hoạt động của servo, như nâng lên, hạ xuống hoặc giữ ở vị trí trung lập.

```

void loop() {
  // Thiết lập vị trí trung lập của các servo
  int servo0 = NEUTRAL - OFFSET0;
  int servo1 = NEUTRAL;
  int servo2 = NEUTRAL;
  int servo3 = NEUTRAL;
  int servo4 = NEUTRAL;
  int servo5 = NEUTRAL + OFFSET5;

  if (Serial.available() > 0) {
    int x = Serial.parseInt();
    int y = Serial.parseInt();
    int pwm_ser = Serial.parseInt();

    // Điều kiện thêm để đưa tất cả servo về mức trung lập
    if (x >= 290 && x <= 360 && y >= 200 && y <= 265) {
      servo0 = NEUTRAL - OFFSET0;
      servo1 = NEUTRAL;
      servo2 = NEUTRAL;
      servo3 = NEUTRAL;
      servo4 = NEUTRAL;
      servo5 = NEUTRAL + OFFSET5;
    }
  }
}

```

Khi nhận được tọa độ x và y của quả bóng từ camera, robot sẽ tính toán góc nghiêng cần thiết để điều chỉnh mặt phẳng. Nếu quả bóng ở gần trung tâm, các servo sẽ được điều chỉnh để giữ quả bóng ở vị trí mong muốn. Trong trường hợp quả bóng lăn ra ngoài, các servo sẽ được kích hoạt để nghiêng mặt phẳng về hướng của quả bóng, giúp nó trở lại trung tâm.

```
}  
    // Nếu không có tín hiệu từ Serial, đưa tất cả servo về vị trí trung lập  
    servo0 = NEUTRAL - OFFSET0;  
    servo1 = NEUTRAL;  
    servo2 = NEUTRAL;  
    servo3 = NEUTRAL;  
    servo4 = NEUTRAL;  
    servo5 = NEUTRAL + OFFSET5;  
}  
  
// Điều khiển các cặp servo  
pwm.setPWM(0, 0, servo0);  
pwm.setPWM(1, 0, servo1);  
pwm.setPWM(2, 0, servo2);  
pwm.setPWM(3, 0, servo3);  
pwm.setPWM(4, 0, servo4);  
pwm.setPWM(5, 0, servo5);  
  
delay(10); // Chờ 10ms trước khi tiếp tục lặp lại  
}
```

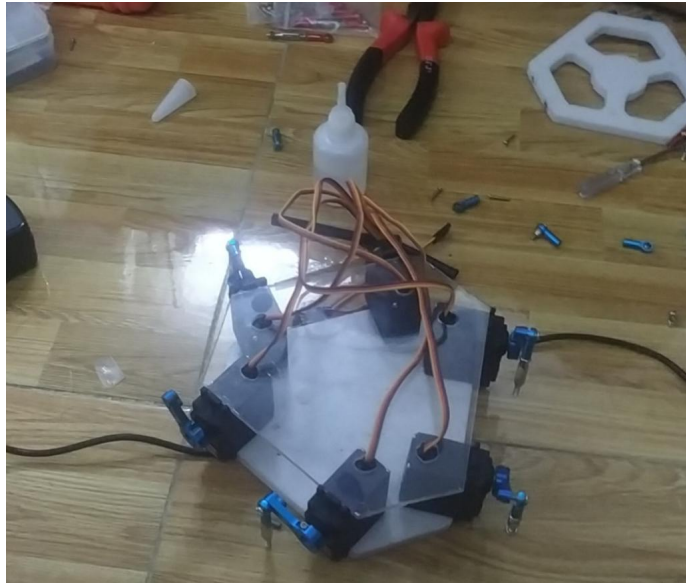
Việc điều chế độ rộng xung PWM trên Arduino Mega được thực hiện thông qua các hàm như setPWM(), cho phép gửi tín hiệu đến từng servo một cách linh hoạt và chính xác. Code sử dụng các điều kiện để xác định hướng và mức độ điều chỉnh servo dựa trên vị trí của quả bóng, đảm bảo rằng robot nhanh chóng phản ứng với các thay đổi trong vị trí của quả bóng.



# CHƯƠNG 4 :TỔNG HỢP HỆ THỐNG

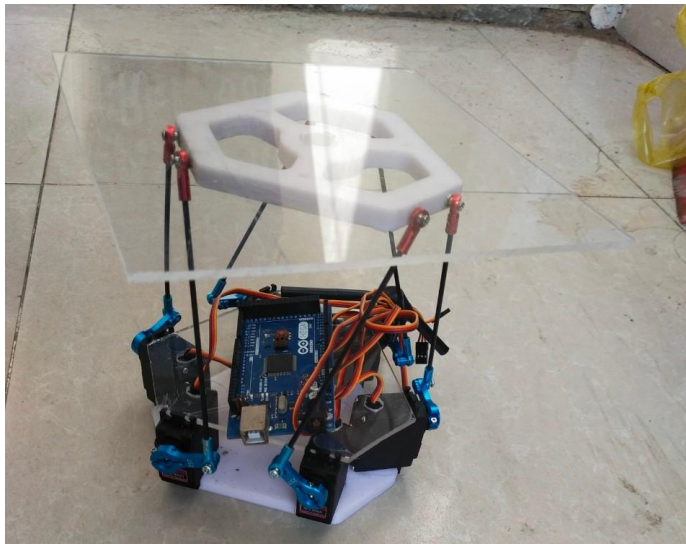
## 1.9 Tổng hợp mô hình

Phần mô hình khi gắn các linh kiện



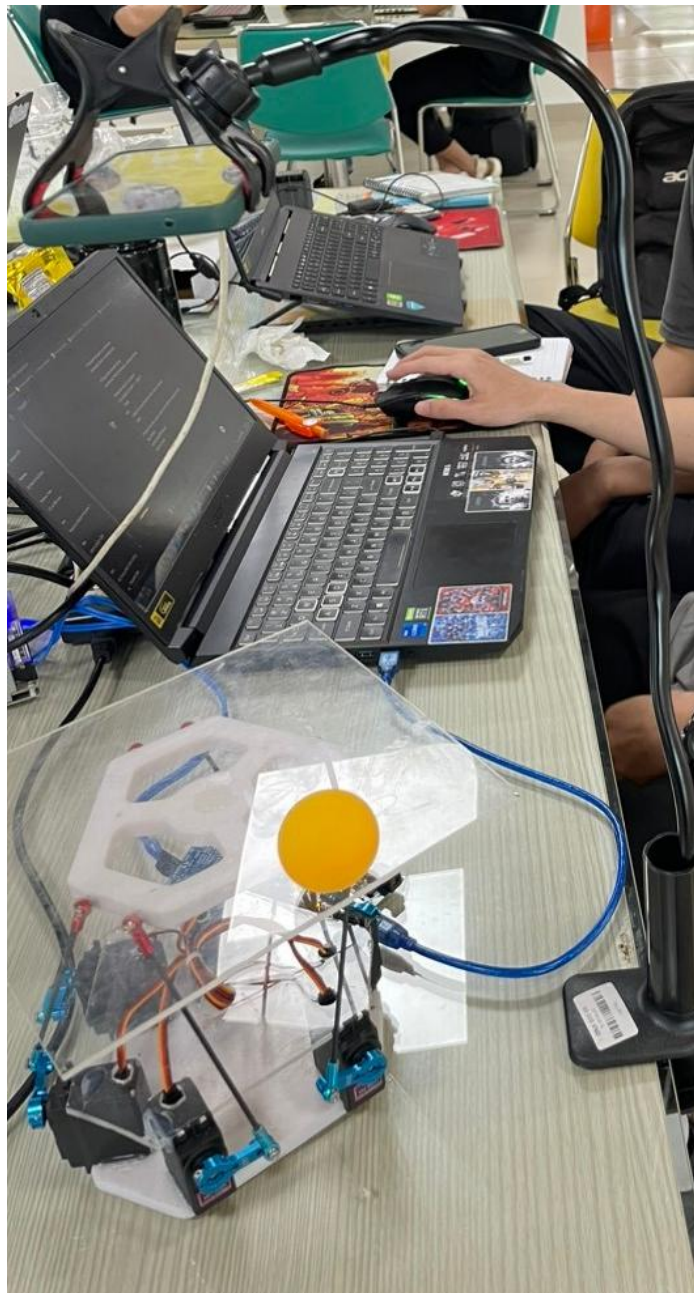
**Hình 4.1:** *Mô hình khi gắn các linh kiện.*

Mô hình được gắn và hoàn thành các linh kiện



**Hình 4.2:** *Mô hình sau khi được gắn linh kiện.*

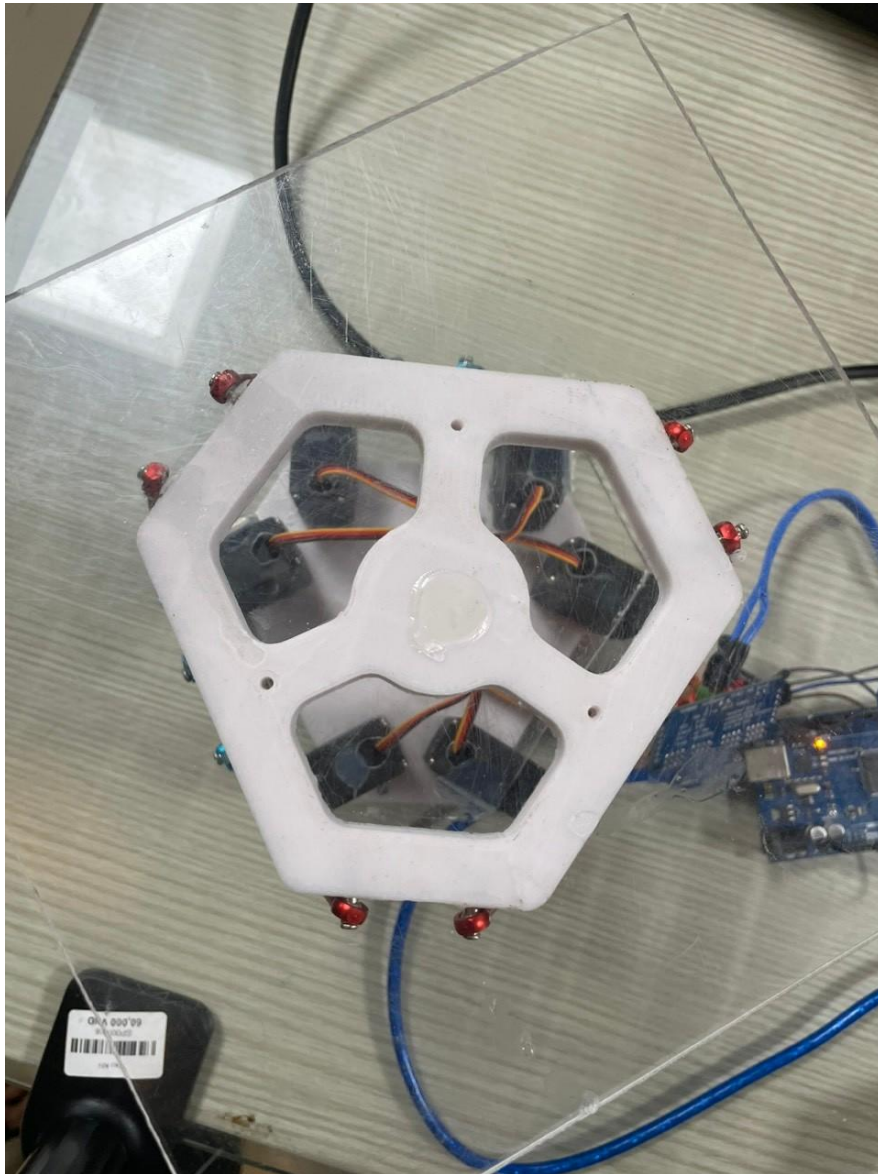
Robot sau khi hoàn thành phần cứng ( điều chỉnh các dây và camera )



**Hình 4.3:** Robot sau khi hoàn thành phần cứng.

## 1.10 Thực nghiệm hệ thống

- Khởi động:

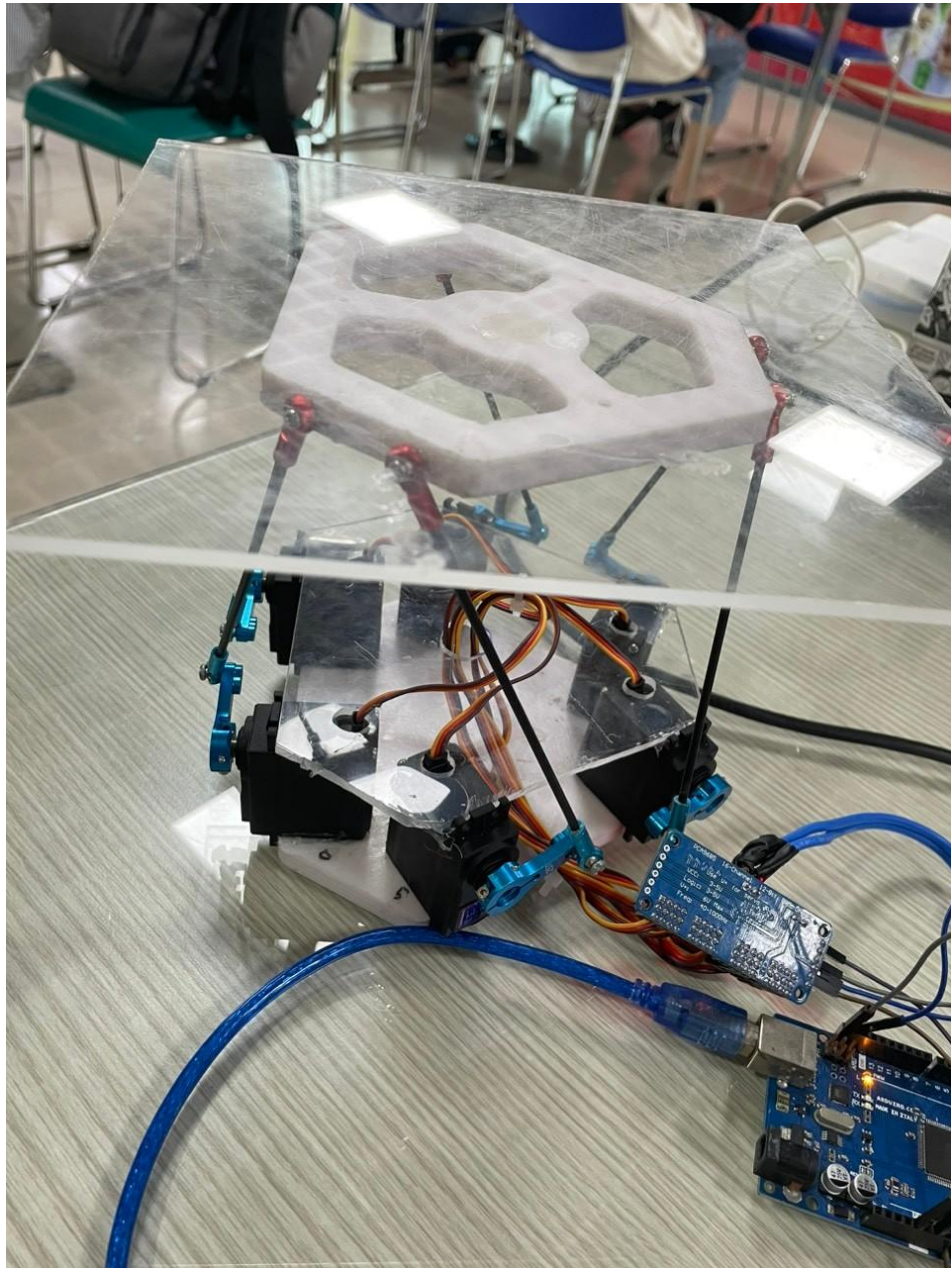


**Hình 4.4:** Khởi động Robot.

Khi hệ thống bắt đầu, robot ở trạng thái chờ và camera mở sẵn để nhận diện quả bóng trên mặt phẳng. Lúc này, các servo được điều chỉnh về vị trí trung lập. Camera sử dụng YOLOv8 để xác định tọa độ của quả bóng. Khi tọa độ được nhận diện, dữ liệu được truyền đến Arduino để bắt đầu điều khiển servo di chuyển..



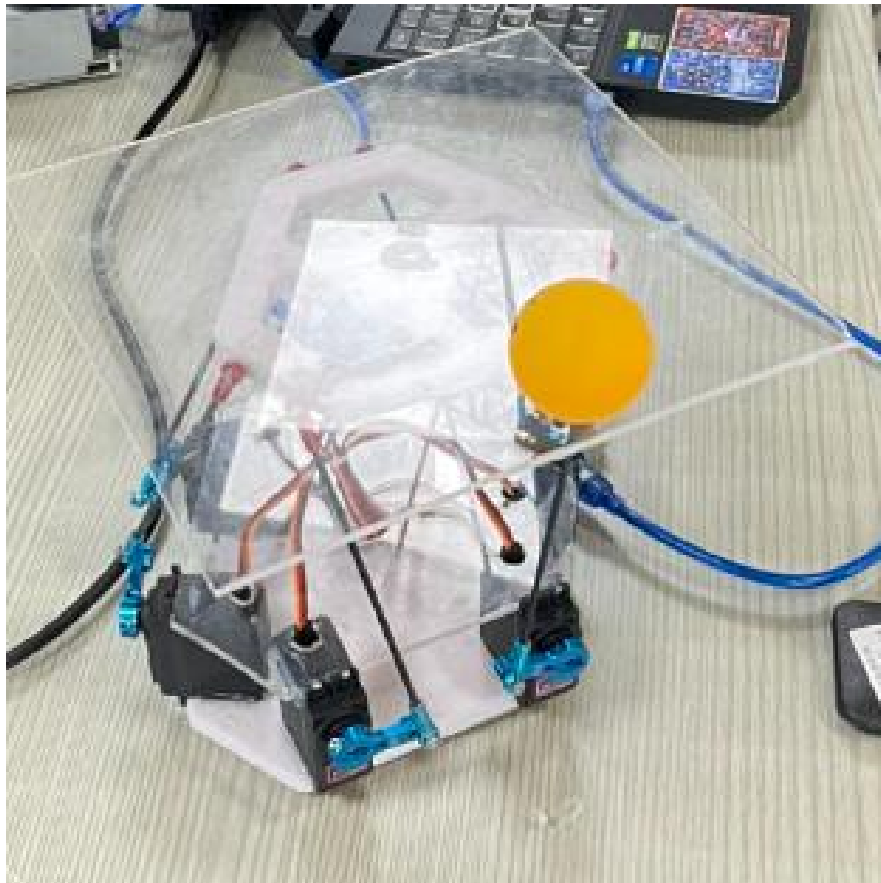
- Robot cân bằng bóng:



**Hình 4.5:** Robot cân bằng bóng trên mặt phẳng.

Sau khi nhận diện được quả bóng, robot bắt đầu điều khiển các servo để di chuyển bề mặt theo hướng cần thiết, giữ quả bóng ở vị trí trung tâm. Nếu quả bóng lệch khỏi trung tâm, thuật toán sẽ điều chỉnh góc nghiêng của bề mặt dựa trên tọa độ của bóng, sử dụng các giá trị PWM đã được tính toán và gửi tới servo. Servo sẽ điều chỉnh góc nghiêng của bề mặt sao cho quả bóng lăn về trung tâm.

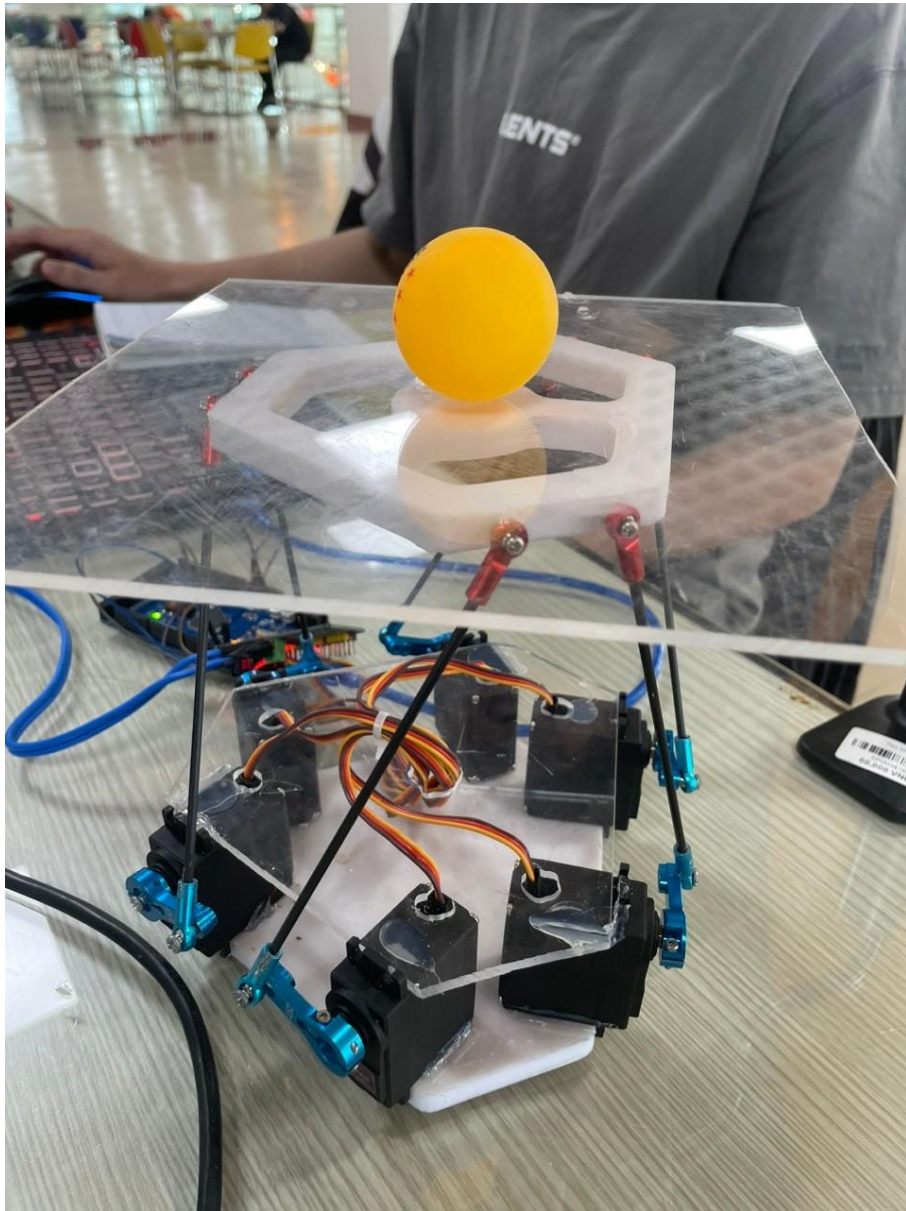
- Xử lý trường hợp bóng lệch khỏi trung tâm:



**Hình 4.6:** *Robot điều chỉnh khi bóng lệch khỏi trung tâm.*

Khi hệ thống phát hiện bóng lệch khỏi trung tâm, các servo sẽ điều chỉnh độ nghiêng của bề mặt theo hai trục x và y dựa trên tọa độ của bóng. Dữ liệu tọa độ của bóng được cập nhật liên tục và gửi về Arduino để tính toán góc cần thiết, giúp robot đưa bóng về vị trí trung tâm một cách nhanh chóng và hiệu quả.

- Kết thúc và trả về trạng thái trung lập:



**Hình 4.7:** Robot trả về trạng thái trung lập sau khi cân bằng thành công.

Khi bóng đã trở về vị trí trung tâm và duy trì trong khoảng thời gian nhất định, hệ thống sẽ đưa tất cả các servo về vị trí trung lập và tiếp tục theo dõi vị trí của quả bóng. Nếu bóng di chuyển, hệ thống sẽ lặp lại quá trình cân bằng.

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1.11 Kết quả đạt được

Trong quá trình thực hiện đề tài "Thiết kế và chế tạo robot cân bằng bóng trên mặt phẳng", em đã có thể triển khai một hệ thống robot hoạt động với độ chính xác tương đối cao và độ ổn định đáng kể, dù chỉ sử dụng những linh kiện đơn giản với giá thành phải chăng. Hệ thống không chỉ đạt được độ chính xác và ổn định mà còn đảm bảo tốc độ đáp ứng và phản hồi nhanh. Đây là một thiết bị có tiềm năng ứng dụng trong việc nghiên cứu, học tập và có thể dễ dàng tiếp cận bởi các sinh viên và những người quan tâm đến lĩnh vực điều khiển tự động.

## 1.12 Các hạn chế

Tuy nhiên, do đây là một thiết bị phục vụ cho học tập, nên hệ thống vẫn chưa được tối ưu hoàn toàn về mặt hiệu năng. Các thuật toán hiện tại còn đơn giản và chưa khai thác hết tiềm năng của phần cứng. Một số hạn chế về mặt thiết kế và hiệu suất có thể được cải thiện trong tương lai để hệ thống có thể hoạt động mượt mà và ổn định hơn trong các môi trường phức tạp hơn.

## 1.13 Phương hướng phát triển trong tương lai

Với sự phát triển không ngừng của khoa học kỹ thuật, đặc biệt là trong lĩnh vực điện tử, đề tài này còn có thể được nâng cấp thêm nhiều tính năng mới và cải thiện về mặt thiết kế để cho ra đời một sản phẩm hoàn thiện hơn, đáp ứng tốt hơn các yêu cầu trong học tập cũng như trong ứng dụng thực tiễn.

Xu hướng phát triển các sản phẩm của ngành công nghiệp vi mạch điện tử hiện nay tập trung vào các tiêu chí đơn giản hóa: thiết kế nhỏ gọn, tốc độ xử lý nhanh và chi phí hợp lý. Nhờ vào xu hướng này, ngành công nghiệp vi mạch điện tử đã có sự phát triển mạnh mẽ, đóng góp tích cực vào nhiều lĩnh vực khác nhau và đạt được doanh thu ấn tượng trên toàn cầu trong những năm qua. Đây cũng là cơ hội để tiếp tục phát triển và hoàn thiện sản phẩm robot cân bằng bóng, nhằm bắt kịp với những tiến bộ của công nghệ và mang lại nhiều giá trị hơn nữa cho người sử dụng.

# TÀI LIỆU THAM KHẢO

- [1] Ball Balancing Robot - [Ball Balancing Robot : 21 Steps \(with Pictures\) - Instructables](#)
- [2] Mô hình điều khiển sự cân bằng giữa bóng và mặt phẳng - [Mô hình điều khiển sự cân bằng giữa bóng và mặt phẳng.pdf \(slideshare.net\)](#)
- [3] NGHIÊN CỨU CHẾ TẠO MÔ HÌNH HỆ THỐNG CÂN BẰNG BÓNG - ĐĨA - [GVHD: TS. NGUYỄN QUỐC ĐỊNH HVTH: LÊ VĂN THƯỜNG \(udn.vn\)](#)
- [4] Ball & Beam: System Modeling - [Control Tutorials for MATLAB and Simulink - Ball & Beam: System Modeling \(umich.edu\)](#)
- [5] Visual Servo Control for Balancing a Ball-Plate System - [ISSN: 2315-4462 \(ijmerr.com\)](#)
- [6] BALL BALANCING - [1 \(psu.edu\)](#)