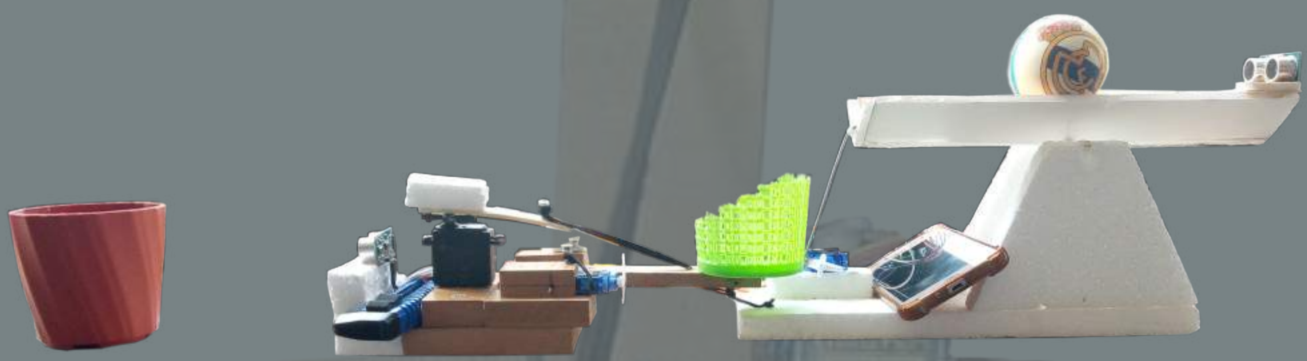
Balancing a ball and throwing it in a basket



Preparation :

Mohammed Saeed Idris
Saleh Khaled Hamandoosh
Salman Semaan Mohammed
Suleiman Redwan Al-Othman
Kenan Mohammed Raed Al-Lababidi

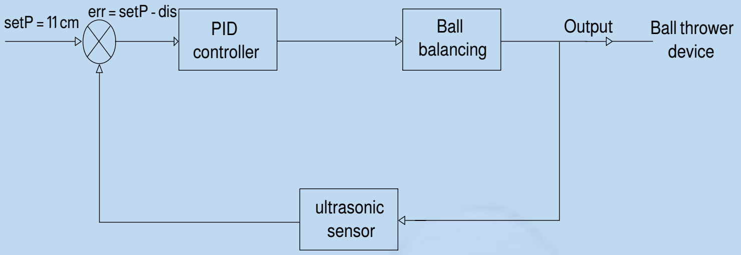
Balancing a ball and throwing it in a basket

Table of content

Introduction :

The system of balancing a ball and throwing it in a basket can balance the ball in the middle of the rudder by a servo motor attached by an iron wire with the rudder where the ball swings to the left of the rudder when the servo angle is 130 degrees and to the far right of the rudder when the servo angle is 0 degrees, where it continues to swing until balanced in the middle where it is done by an ultrasonic sensor that measures the distance of the middle of the rudder and the process is most likely carried out by the PID controller, whose reference value is the middle of the rudder The values of the servo angles according to the output of pid, after the balance will swing the ball to the far left at the angle of Servo 180 degrees to the basket of the ball thrower device consisting of the structure and two engines Servo, one of which installs the arm of the basket where the angle of the first servo is 180 degrees and the other will tighten the rubber strapped to the arm of the ejector where its angle fits after the goal basket measured by a distance sensor and then the arm of the ejector is released by making the angle of the first servo 180 degrees and the angle of the ejector is thrown into the goal basket

Block diagram for balancer ball :



setP = 11 cm

err = setP - dis

PID controller

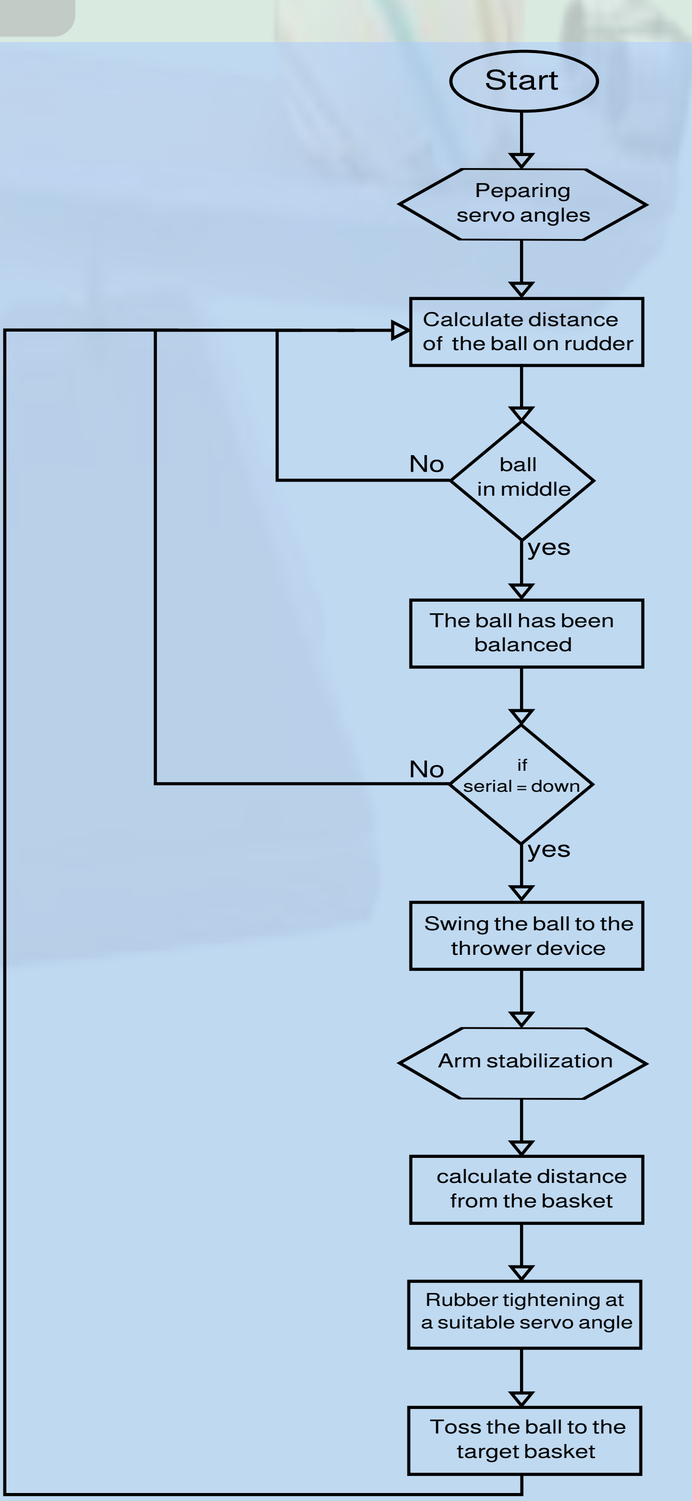
Ball balancing

Output

Ball thrower device

ultrasonic sensor

Flow chart for all project :



Start

Peparing servo angles

Calculate distance of the ball on rudder

ball in middle

No

yes

The ball has been balanced

if serial = down

No

yes

Swing the ball to the thrower device

Arm stabilization

calculate distance from the basket

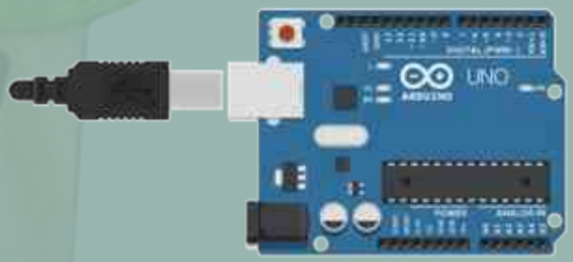
Rubber tightening at a suitable servo angle

Toss the ball to the target basket

Parts used :

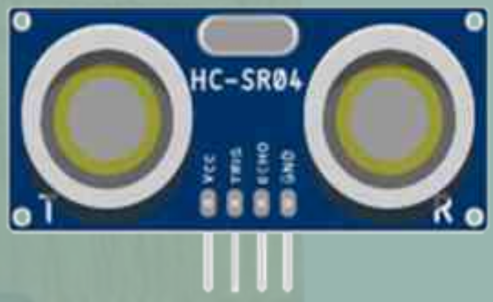
One Arduino UNO : Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino is able to get inputs, compare values and to output values in order to control an output device. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. We write the instructions using Arduino Programming Language on the IDE software [1].

Arduino UNO



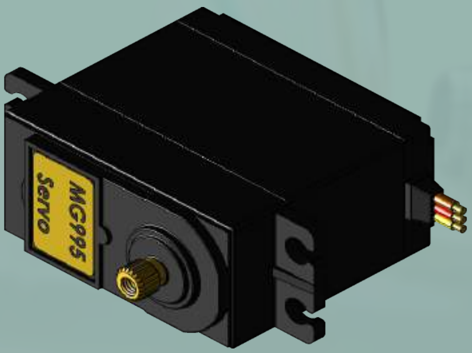
Two Ultrasonic Sensor: The ultrasonic distance sensor estimates distance by transmitting an ultrasonic soundwave and measuring the time taken to receive the wave. The ultrasonic sensor sends the wave using the Trigger and receive the wave using the Echo. The minimum and maximum measurable distances are 2cm and 4m, respectively [2].

Ultrasonic Sensor



One Servo motor : MG995 High Speed Metal Gear Dual Ball Bearing Servo
The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.
This high-speed standard servo can rotate approximately 120 degrees (60 in each direction).
You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG995 Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast! [3] .

Servo motor

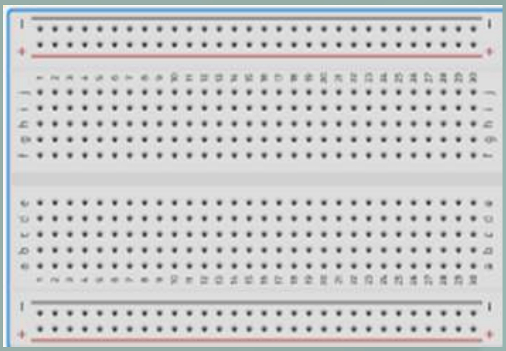


Two Servo moter : SERVO MOTOR SG90
Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware [5].

Servo motor

Breadboard

Breadboard:A breadboard is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used for slicing bread. In the 1970s the solderless breadboard became available and nowadays the term "breadboard" is commonly used to refer to these [4].

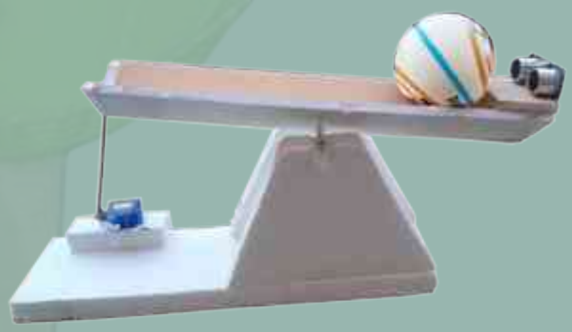


Goal Basket :

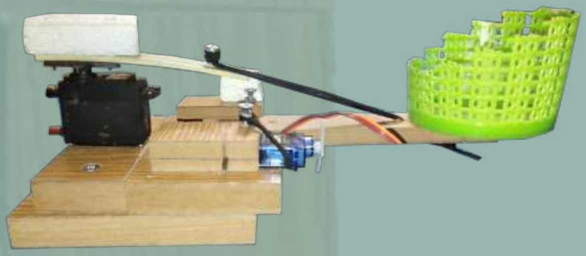
The ball is thrown inside it

Balancer ball:

Where the engine consists of a servo placed at the bottom and an iron wire attached to it and with the rudder and ultrasonic sensor placed on the rudder to measure the distance of the ball whether it is in half or not

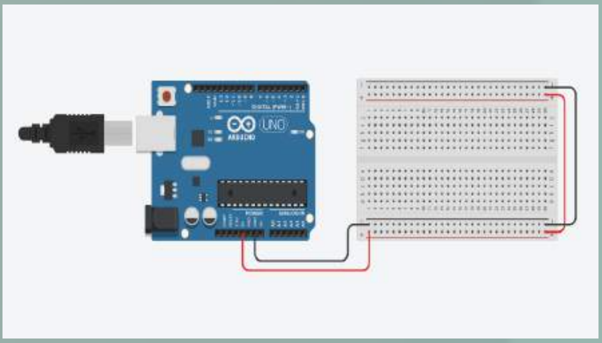


Ball thrower device: Consists of a wooden structure with two servo engines, a basket, an arm and a rubber, the first servo fastens the arm at the bottom and then the other servo tightens the rubber tied with the arm and then the first servo frees the arm and the ball is thrown into the goal basket by an ultrasonic sensor tied with the ejector

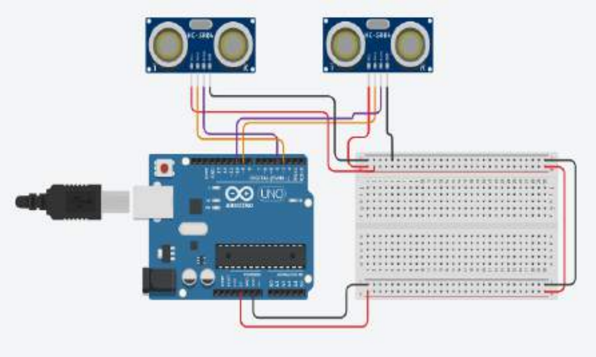


Procedure :

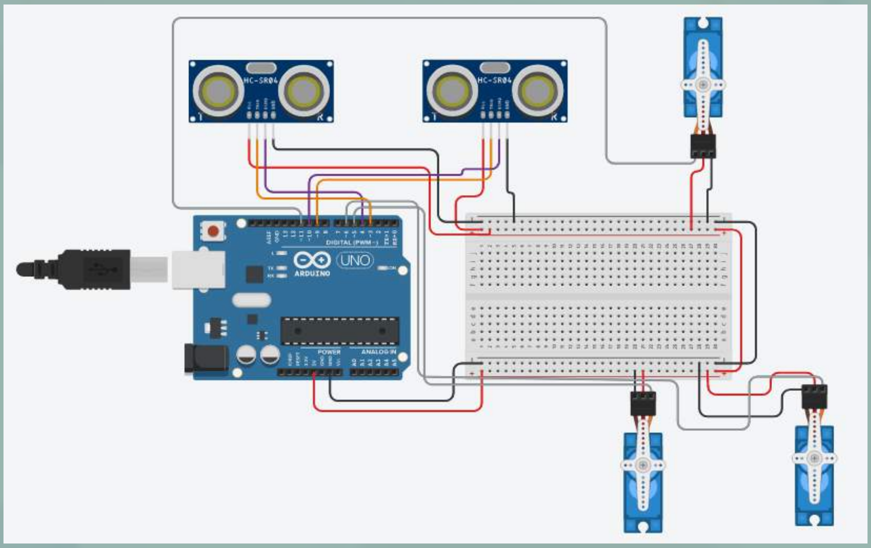
At first , we connect the positiive and the negative side of the breadboard to the 5V and ground of the Arduino respectively.



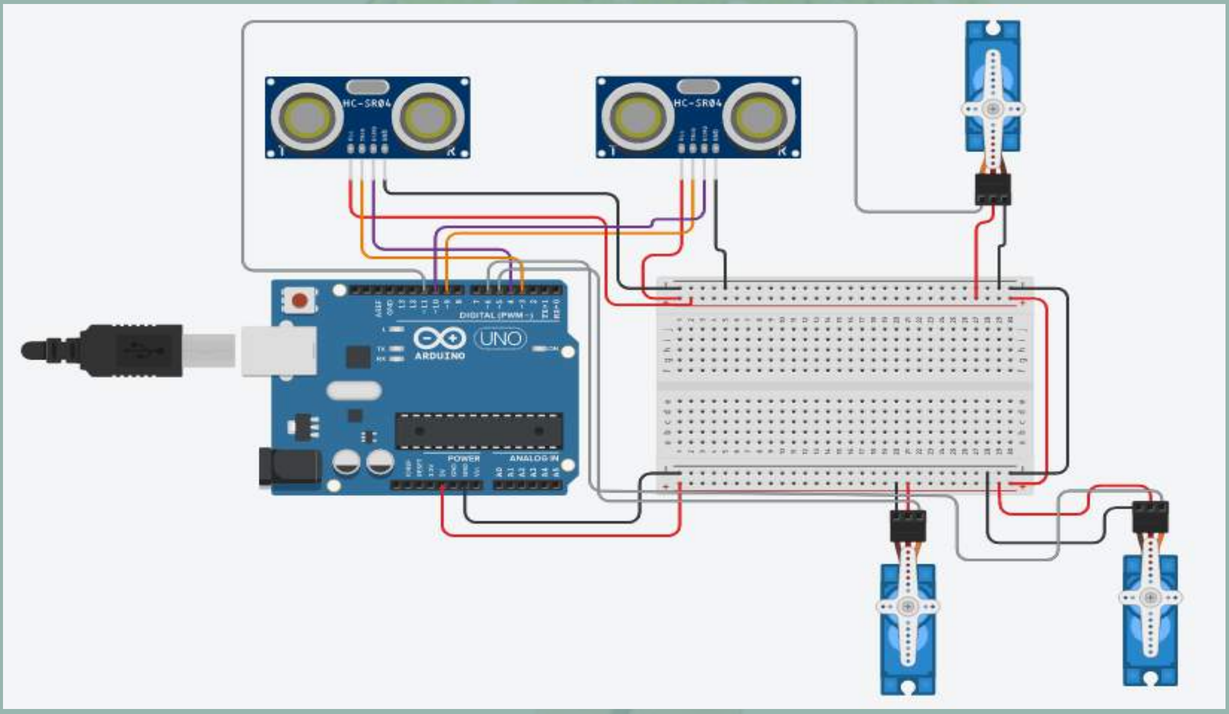
Then we connected th two ultrasinic sensor , the first sensor is connected as follows :
Trigger to digital 9 , Echo to digital 10 ,
VCC to the 5V , Ground to the ground .
The second sensor is connected as follows :
Trigger to digital 3 , Echo to digital 4,
VCC to the 5V , Ground to the ground .



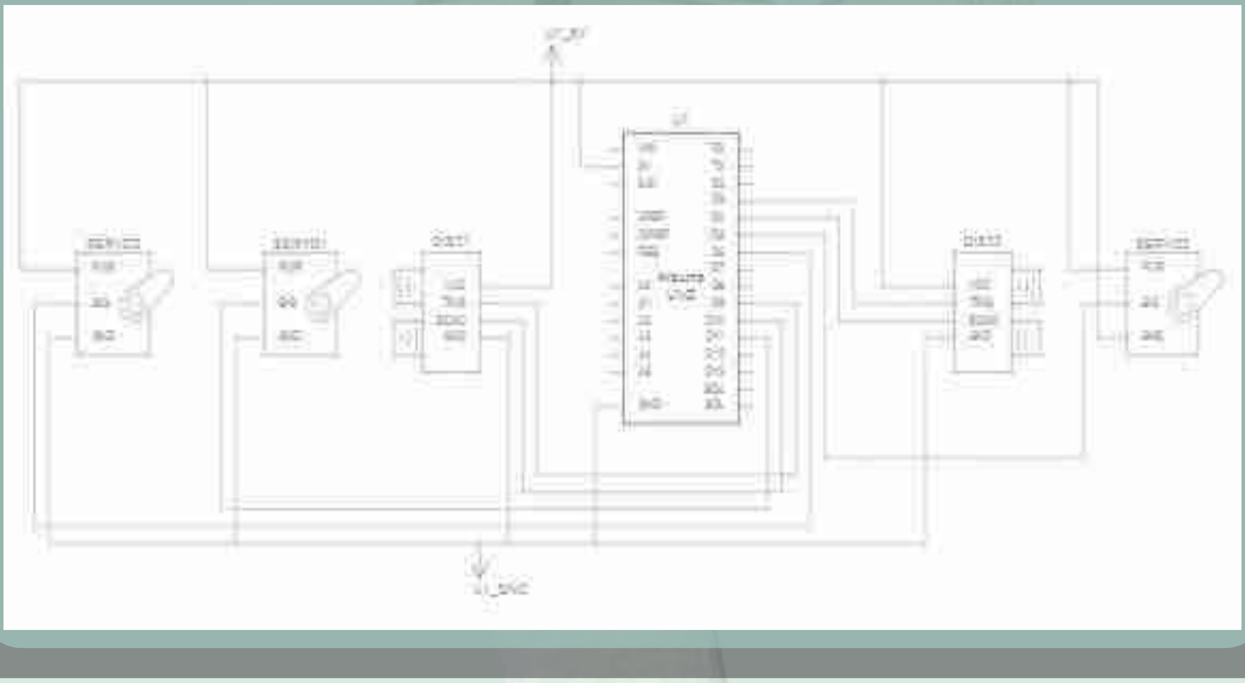
Then we have three servo motor :
The first servo is connected as follows :
ground to the GND , power to the 5V , signal to digital pin 11 .
The first servo is connected as follows :
ground to the GND , power to the 5V , signal to digital pin 5 .
The first servo is connected as follows :
ground to the GND , power to the 5V , signal to digital pin 6 .



The Circuit :



Schematic view :



Code :

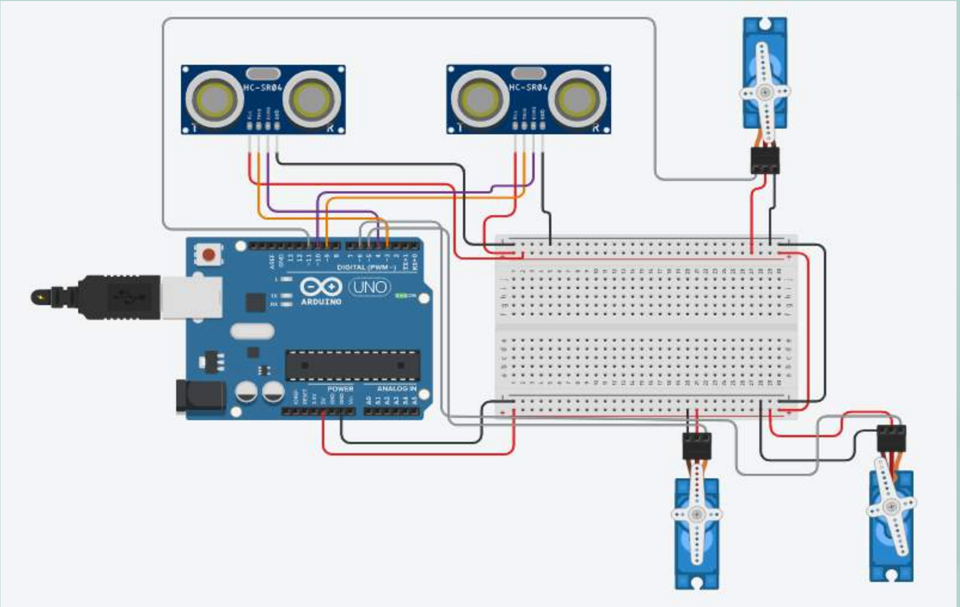
```
#include <Servo.h>
Servo servo;
Servo servo1;
Servo servo2;
#define trig 9
#define echo 10
#define trig1 3
#define echo1 4
double kp = 15;
double ki = 0.02;
double kd = 10;
double Fvalue;
double setP;
double error = 0 ;
double priError = 0;
double toError = 0;
String welcome = "";
bool s = true , l = true ;
void setup() {
 pinMode(trig, OUTPUT);
 pinMode(echo, INPUT);
 pinMode(trig1, OUTPUT);
 pinMode(echo1, INPUT);
 servo.attach(11);
 servo1.attach(5);
 servo2.attach(6);
 Serial.begin(9600);
 servo.write(65);
 servo1.write(180);
 servo2.write(10);
}
void loop() {
 if(Serial.available()>0){
  if(s&l){
   welcome = Serial.readString();
   if(welcome == "down"){
    kp = 0;
    ki = 0;
    kd = 0;
    Fvalue = 180;
    servo.write(Fvalue);
    delay(2000);
    servo1.write(50);
    servo2.write(10);
    delay(2000);
    print_Serial_2();
    double d = distance1();
    double z = map(d, 8, 20, 120, 150);
    servo1.write(50);
    servo2.write(z);
    Serial.print("( ");
    Serial.print( z );
```

```
Serial.println(" )");
    delay(3000);
    servo1.write(180);
    servo2.write(z);
    delay(2000);
    servo1.write(180);
    servo2.write(10);
    s = false , l = true;
    kp = 15;
    ki = 0.02;
    kd = 10;
    error = 0;
    priError = 0;
    toError = 0;
    PID();
    }
    else{
     kp = welcome.toDouble();
     s = true , l = false;
     }
   }
   else if((s|l)&(l==true)){
    welcome = Serial.readString();
    kp = welcome.toDouble();
    s = true , l = false;
    }
   else if((s|l)&(l==false)){
    welcome = Serial.readString();
    ki = welcome.toDouble();
    s = false , l = false;
    }
   else if(!(s&l)){
    welcome = Serial.readString();
    kd = welcome.toDouble();
    s = true , l = true;
    }
  }
 if(Fvalue == 180){}
 else{
  PID();
  print_Serial_1();
}
}
double distance1 () {
 digitalWrite(trig1, LOW);
 delayMicroseconds(4);
 digitalWrite(trig1, HIGH);
 delayMicroseconds(10);
 digitalWrite(trig1, LOW);
 double t = pulseIn(echo1, HIGH);
 double cm = t / 29 / 2;
 return cm;
}
```

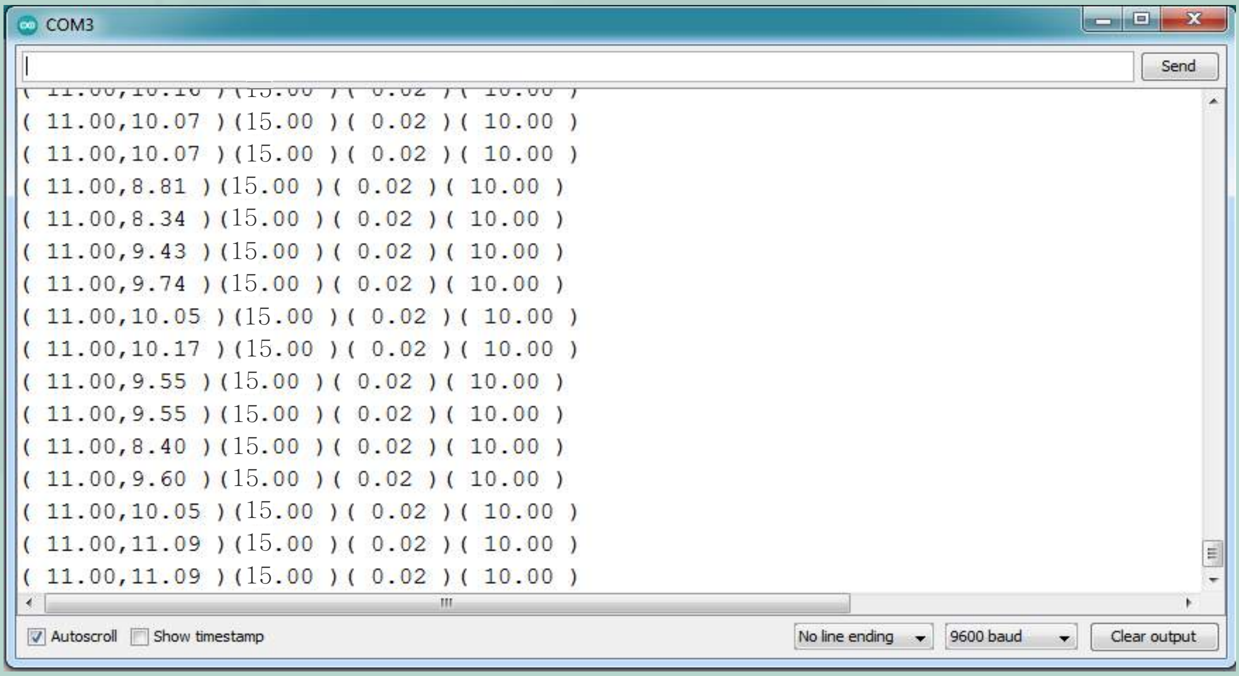
```
double distance () {
 digitalWrite(trig, LOW);
 delayMicroseconds(4);
 digitalWrite(trig, HIGH);
 delayMicroseconds(10);
 digitalWrite(trig, LOW);
 double t = pulseIn(echo, HIGH);
 double cm = t / 29 / 2;
 return cm;
}
void PID() {
 double dis = distance ();
  setP = 11;
 error = setP - dis;
 double Pvalue = error * kp;
 double Ivalue = toError * ki;
 double Dvalue = (error - priError) * kd;
 double PIDvalue = Pvalue + Ivalue + Dvalue;
 priError = error;
 toError += error;
 Fvalue = PIDvalue;
 Fvalue = map(Fvalue, -130, 130, 0, 126);
 if (Fvalue < 0) {
  Fvalue = 0;
 }
 if (Fvalue > 100) {
  Fvalue = 126;
 }
 servo.write(Fvalue);
}
void print_Serial_1(){
  double a = distance();
  Serial.print("( ");
  Serial.print( setP );
  Serial.print( "," );
  Serial.print( a );
  Serial.print(" )");
  Serial.print("( ");
  Serial.print( kp );
  Serial.print(" )");
  Serial.print("( ");
  Serial.print( ki );
  Serial.print(" )");
  Serial.print("( ");
  Serial.print( kd );
  Serial.println(" )");
 }
void print_Serial_2(){
 double b = distance1();
 Serial.print("( ");
 Serial.print( b );
 Serial.print(" )");
 }
```

Simulation :

On tinkerCad :



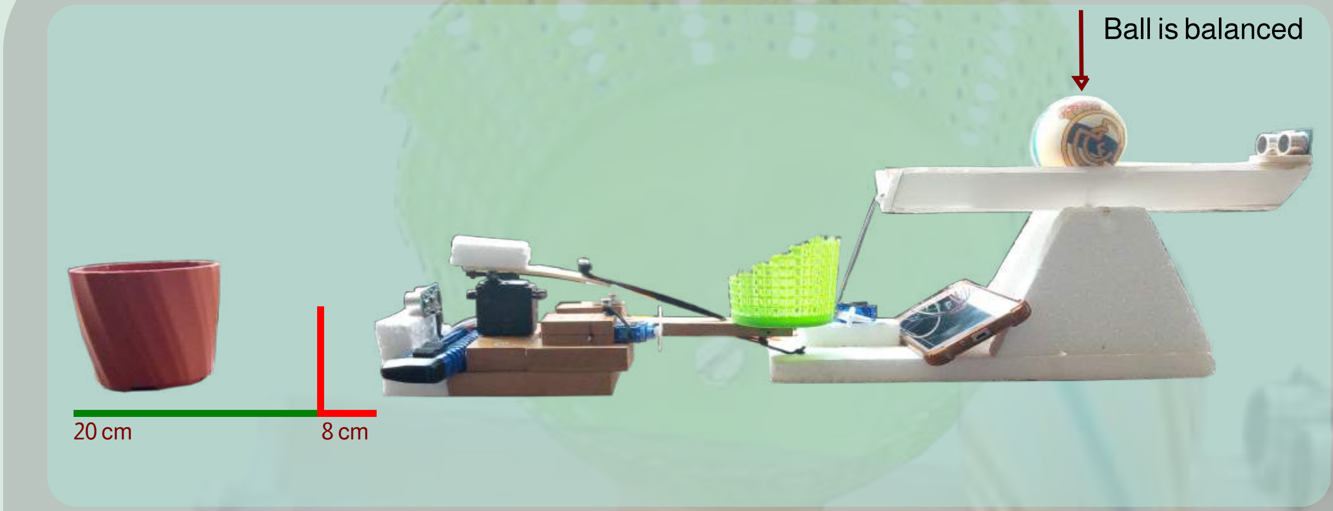
The serial :

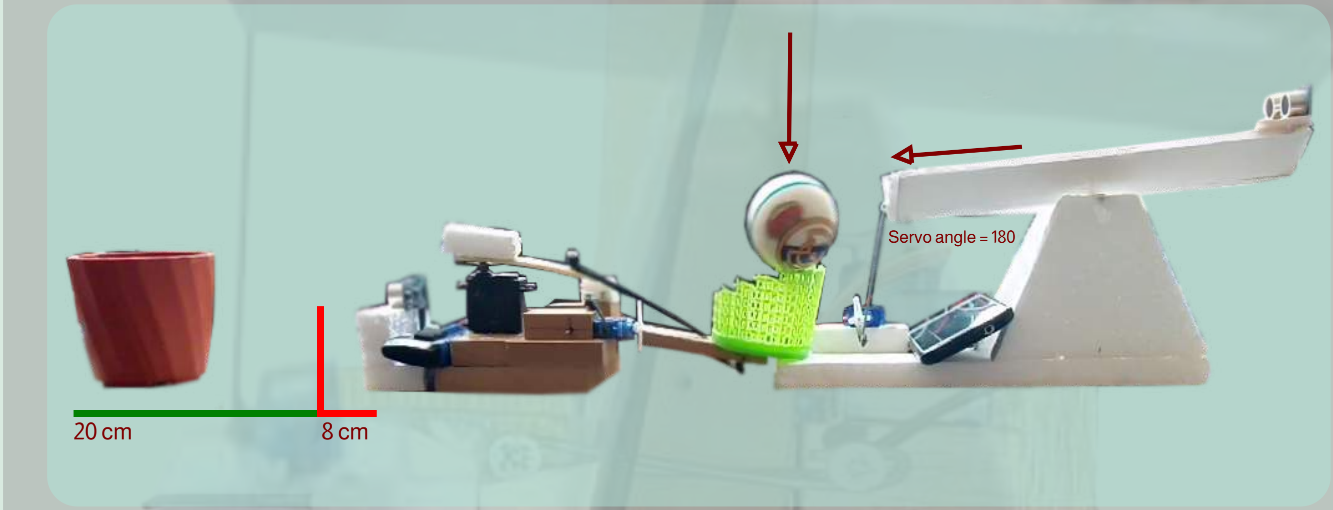


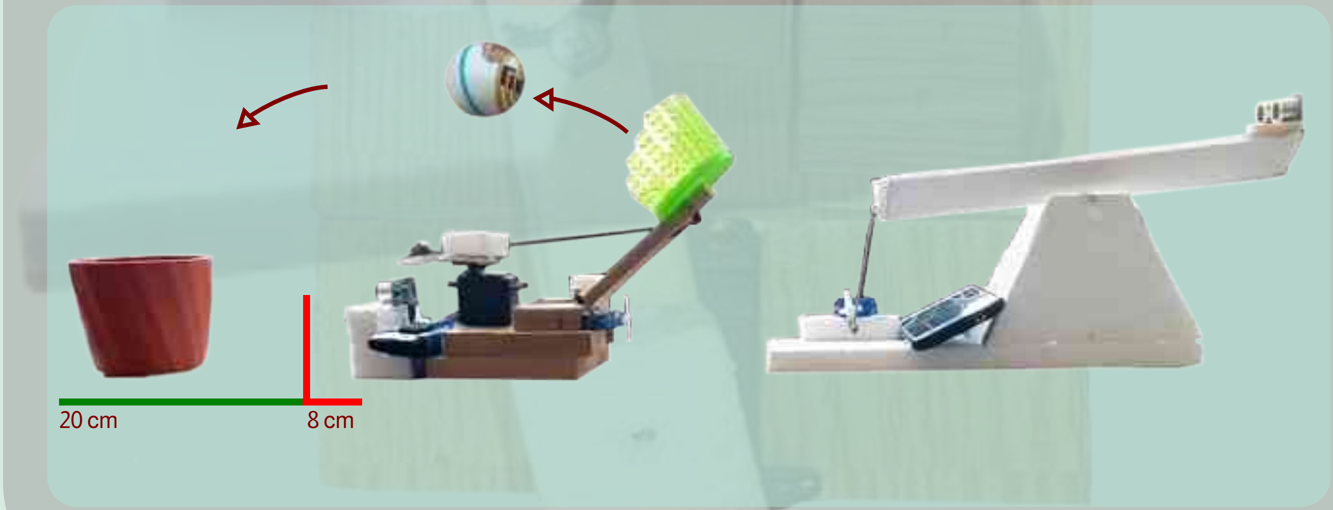
```
( 11.00,10.10 )(15.00 )( 0.02 )( 10.00 )
( 11.00,10.07 )(15.00 )( 0.02 )( 10.00 )
( 11.00,10.07 )(15.00 )( 0.02 )( 10.00 )
( 11.00,8.81 )(15.00 )( 0.02 )( 10.00 )
( 11.00,8.34 )(15.00 )( 0.02 )( 10.00 )
( 11.00,9.43 )(15.00 )( 0.02 )( 10.00 )
( 11.00,9.74 )(15.00 )( 0.02 )( 10.00 )
( 11.00,10.05 )(15.00 )( 0.02 )( 10.00 )
( 11.00,10.17 )(15.00 )( 0.02 )( 10.00 )
( 11.00,9.55 )(15.00 )( 0.02 )( 10.00 )
( 11.00,9.55 )(15.00 )( 0.02 )( 10.00 )
( 11.00,8.40 )(15.00 )( 0.02 )( 10.00 )
( 11.00,9.60 )(15.00 )( 0.02 )( 10.00 )
( 11.00,10.05 )(15.00 )( 0.02 )( 10.00 )
( 11.00,11.09 )(15.00 )( 0.02 )( 10.00 )
( 11.00,11.09 )(15.00 )( 0.02 )( 10.00 )
```

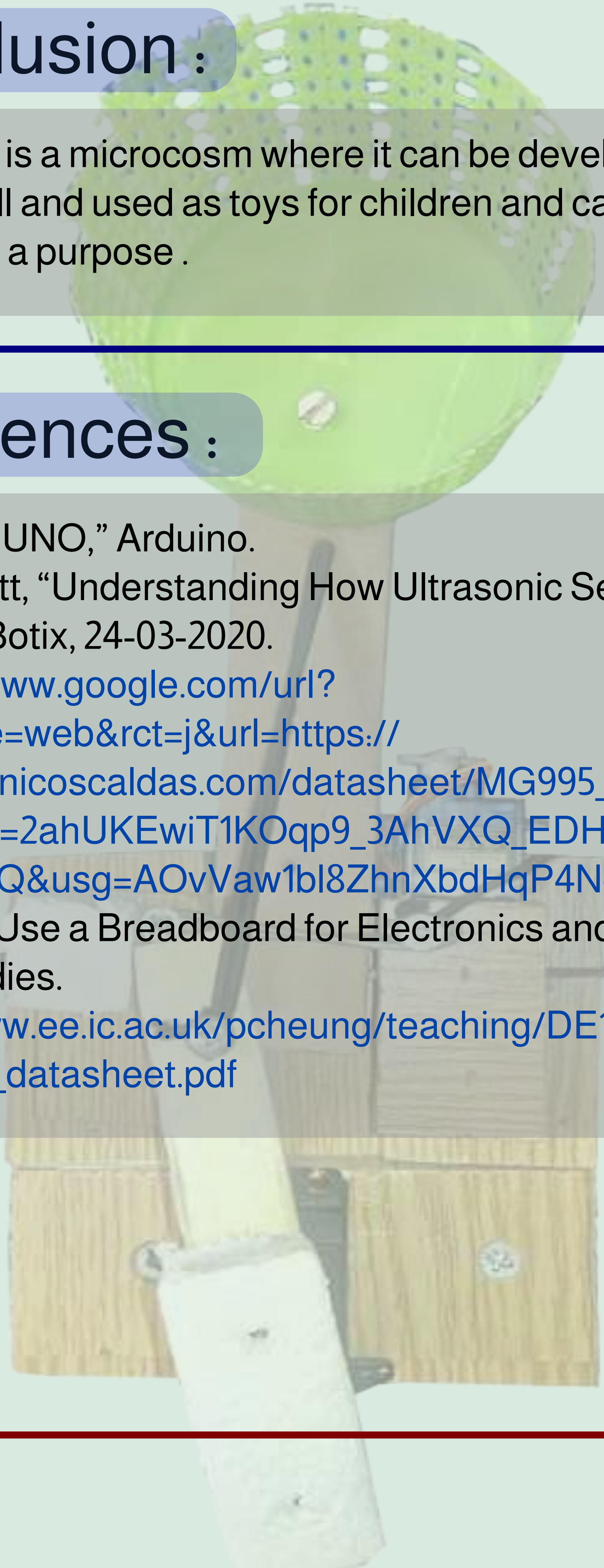
☑ Autoscroll □ Show timestamp No line ending 9600 baud Clear output

Practically :









Conclusion :

This project is a microcosm where it can be developed for use in basketball and used as toys for children and can be used in factories for a purpose .

References :

[1] “Arduino UNO,” Arduino.
[2] R. Burnett, “Understanding How Ultrasonic Sensors Work,” MaxBotix, 24-03-2020.
[3] https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf&ved=2ahUKEwiT1KOqp9_3AhVXQ_EDHX14AkEQFnoECAUQAQ&usg=AOvVaw1bl8ZhnXbdHqP4Ng61AgtN
[4] “How to Use a Breadboard for Electronics and Circuits,” sciencebuddies.
[5] http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf