

PRIMERA VERSIÓN OPERATIVA DEL RADAR

CÓDIGO EN ARDUINO:

```
#include <Servo.h>

// Definimos los pines del sensor ultrasonidos.

const int trigPin = 10;

const int echoPin = 11;

// variables

long duracion;

int distancia;

Servo servoMotor; // creamos un servo objeto para controlar el servomotor

void setup() {

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  Serial.begin(9600);

  servoMotor.attach(12); // Definimos pin de entrada
}

void loop() {

  // el servo rota de 0 a 180 grados

  for(int ang=0;ang<=180;ang++){

    servoMotor.write(ang);

    delay(30);

    distancia = calculaDistancia();// funcion que calcula la distancia


    // Estos puertos lo que hacen es mandar informacion al processing para que pueda leer los
    // datos que llegan desde el arduino.

    Serial.print(ang); //Envia el angulo.

    Serial.print(","); //Envia una coma para diferenciar el angulo de la distancia.

    Serial.print(distancia); //Envia la distancia (aproximada).

    Serial.print("."); //Envia un punto para finalizar los elementos de la serie.(Esto lo usaremos
    // para diferenciar datos en el processing).
```

```

}

//Repetimos lo anterior teniendo en cuenta que nuestro radar vuelve a la posicion inicial.
for(int ang=180;ang>0;ang--){
  servoMotor.write(ang);
  delay(30);
  distancia = calculaDistancia();
  Serial.print(ang);
  Serial.print(",");
  Serial.print(distancia);
  Serial.print(".");
}
}

// Funcion que calcula la distancia
int calculaDistancia(){

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duracion = pulseIn(echoPin, HIGH);

  distancia= duracion*0.034/2; // calculamos la distancia de los objetos a partir de la duración
  al encontrar dicho objeto y la velocidad de onda en encontrarlo

  return distancia;
}

```

CÓDIGO EN PROCESSING:

```

import processing.serial.*; // libreria para comunicar con el arduino
import java.awt.event.KeyEvent; // libreria para extraer informacion del arduino
import java.io.IOException;

Serial Puerto; // definimos puerto

```

```

// definimos variables

String angulo="";
String distancia="";
String data="";
String objeto;
float Distanciapixel;
int angAngulo, angDistancia;
int indice1=0;
int indice2=0;
PFont orcFont;
void setup() {

    size (1200, 700); // tamaño aproximado de la pantalla
    smooth();

    Puerto = new Serial(this,"COM3", 9600); // Establece la comunicacion con el arduino

    Puerto.bufferUntil('.'); // lee los datos del puerto enviado desde el arduino hasta que llegue a
    un punto (angulo,distacia.)
}

void draw() {

    fill(98,245,31);

    noStroke();

    fill(0,4);
    rect(0, 0, width, height-height*0.065);

    fill(98,245,31);

    // funciones que dibujan el radar

    radar();

    lineas();

    objeto();

```

```

    texto();
}

void serialEvent (Serial Puerto) { // lee los datos del puerto

    data = Puerto.readStringUntil('.');
    data = data.substring(0,data.length()-1);

    indice1 = data.indexOf(","); //encuentra la coma (,) y lo escribe en la variable indice1
    angulo= data.substring(0, indice1); //lee la informacion desde el 0 hasta la posicion indice1
    para saber el angulo que manda la placa arduino al puerto
    distancia= data.substring(indice1+1, data.length()); //lee la distancia

    angAngulo = int(angulo);
    angDistancia = int(distancia);
}

//Dibuja el radar
void radar() {
    pushMatrix();
    translate(width/2,height-height*0.074);
    noFill();
    strokeWeight(2);
    stroke(98,245,31);
    // dibujamos los arcos del radar
    arc(0,0,(width-width*0.0625),(width-width*0.0625),PI,TWO_PI);
    arc(0,0,(width-width*0.27),(width-width*0.27),PI,TWO_PI);
    arc(0,0,(width-width*0.479),(width-width*0.479),PI,TWO_PI);
    arc(0,0,(width-width*0.687),(width-width*0.687),PI,TWO_PI);
    // dibujamos las lineas de los angulos
    line(-width/2,0,width/2,0);
    line(0,0,(-width/2)*cos(radians(30)),(-width/2)*sin(radians(30)));
}

```

```

line(0,0,(-width/2)*cos(radians(60)),(-width/2)*sin(radians(60)));
line(0,0,(-width/2)*cos(radians(90)),(-width/2)*sin(radians(90)));
line(0,0,(-width/2)*cos(radians(120)),(-width/2)*sin(radians(120)));
line(0,0,(-width/2)*cos(radians(150)),(-width/2)*sin(radians(150)));
line((-width/2)*cos(radians(30)),0,width/2,0);

popMatrix();
}

//Dibuja el objeto
void objeto() {
  pushMatrix();

  translate(width/2,height-height*0.074);

  strokeWeight(9);

  stroke(255,10,10);

  Distanciapixel = angDistancia*((height-height*0.1666)*0.025);

  //limitamos el rango a 40 cm
  if(angDistancia<50){

    //dibujamos los objetos segun la distancia y angulo

    line(Distanciapixel*cos(radians(angAngulo)), -Distanciapixel*sin(radians(angAngulo)), (width-
width*0.505)*cos(radians(angAngulo)), -(width-width*0.505)*sin(radians(angAngulo)));

  }

  popMatrix();
}

//Dibuja las lineas
void lineas() {
  pushMatrix();

  strokeWeight(3);

  stroke(30,250,60);

  translate(width/2,height-height*0.074);

  line(0,0,(height-height*0.12)*cos(radians(angAngulo)), -(height-
height*0.12)*sin(radians(angAngulo)));

  popMatrix();
}

```

```
//Introduce la informacion en pantalla
```

```
void texto() {
```

```
    pushMatrix();
```

```
    if(angDistancia>50) {
```

```
        objeto = "Fuera de rango";
```

```
    }
```

```
    else {
```

```
        objeto = "En rango";
```

```
    }
```

```
    fill(0,0,0);
```

```
    noStroke();
```

```
    rect(0, height-height*0.0648, width, height);
```

```
    fill(98,245,31);
```

```
    textSize(25);
```

```
    text("10cm",width-width*0.3854,height-height*0.0833);
```

```
    text("20cm",width-width*0.281,height-height*0.0833);
```

```
    text("30cm",width-width*0.177,height-height*0.0833);
```

```
    text("40cm",width-width*0.0729,height-height*0.0833);
```

```
    textSize(40);
```

```
    text("Radar Ultrasonido", width-width*0.875, height-height*0.0277);
```

```
    text("Angulo: " + angAngulo + " °", width-width*0.48, height-height*0.0277);
```

```
    text("Dist:", width-width*0.26, height-height*0.0277);
```

```
    if(angDistancia<50) {
```

```
        text("    " + angDistancia + " cm", width-width*0.225, height-height*0.0277);
```

```
    }
```

```
    textSize(25);
```

```
    fill(98,245,60);
```

```
    translate((width-width*0.4994)+width/2*cos(radians(30)),(height-height*0.0907)-width/2*sin(radians(30)));
```

```

rotate(-radians(-60));

text("30°",0,0);

resetMatrix();

translate((width-width*0.503)+width/2*cos(radians(60)),(height-height*0.0888)-
width/2*sin(radians(60)));

rotate(-radians(-30));

text("60°",0,0);

resetMatrix();

translate((width-width*0.507)+width/2*cos(radians(90)),(height-height*0.0833)-
width/2*sin(radians(90)));

rotate(radians(0));

text("90°",0,0);

resetMatrix();

translate(width-width*0.513+width/2*cos(radians(120)),(height-height*0.07129)-
width/2*sin(radians(120)));

rotate(radians(-30));

text("120°",0,0);

resetMatrix();

translate((width-width*0.5104)+width/2*cos(radians(150)),(height-height*0.0574)-
width/2*sin(radians(150)));

rotate(radians(-60));

text("150°",0,0);

popMatrix();
}

```