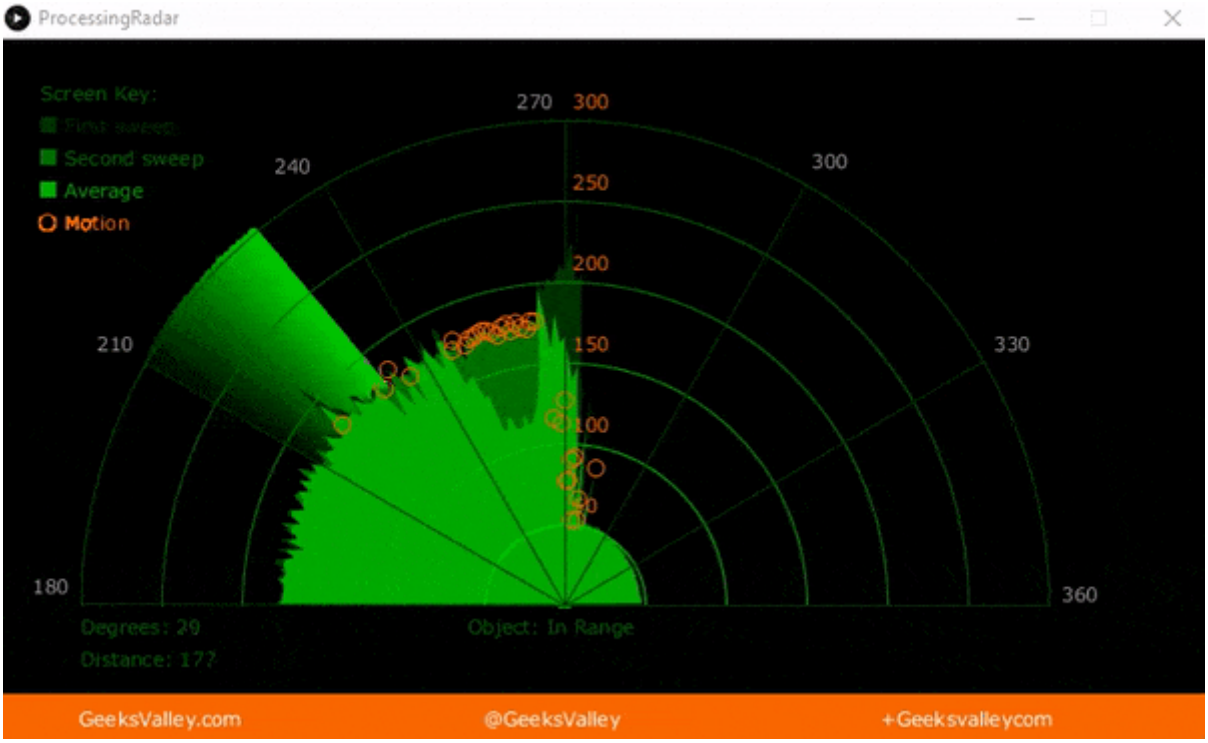
مشروع الرادار الالكتروني

سنقوم في هذا المشروع بعمل نظام بسيط لرصد إحداثيات موقع الأجسام عن طريق صناعة رادار رقمي بإستخدام جهاز مستشعر الموجات فوق الصوتية HC-SR04، ولوحة الأردوينو كمتحكم للنظام . ويتم استخدام جهاز الحاسوب لعرض الواجهة الرسومية.



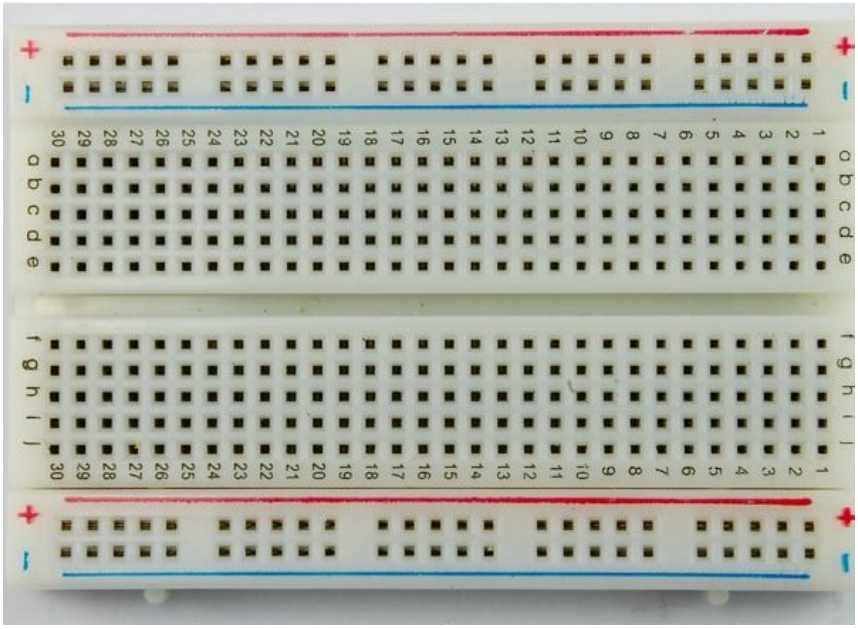
القطع المطلوبة:

الأدوات التي تحتاجها لهذا المشروع :

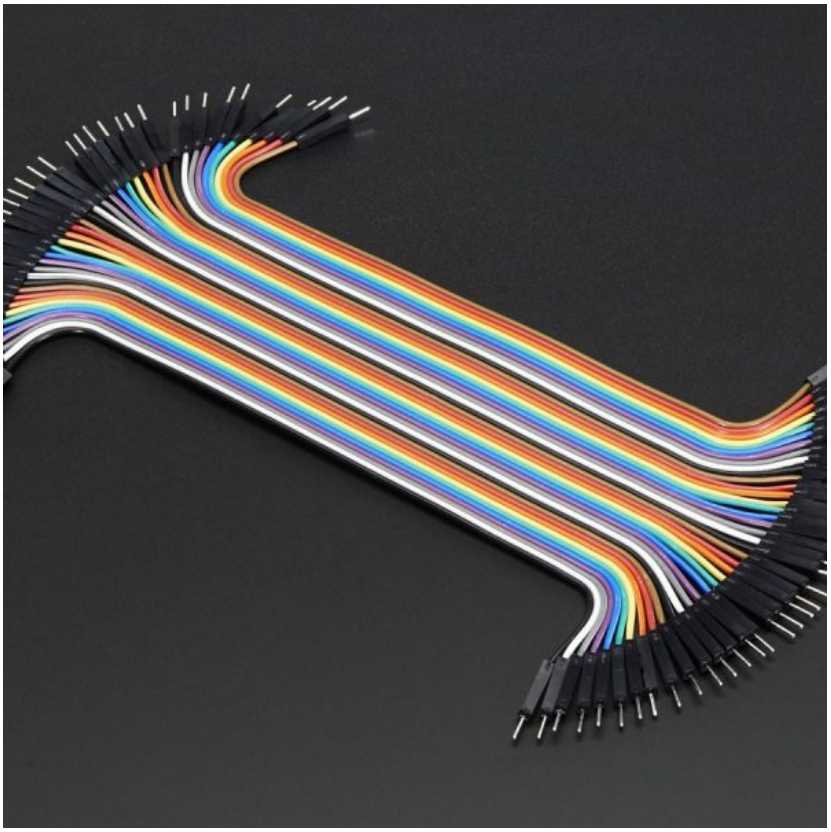
مستشعر الموجات فوق الصوتية (Ultrasonic sensor HC- SR04).

محرك سيرفو (Servo Motor).

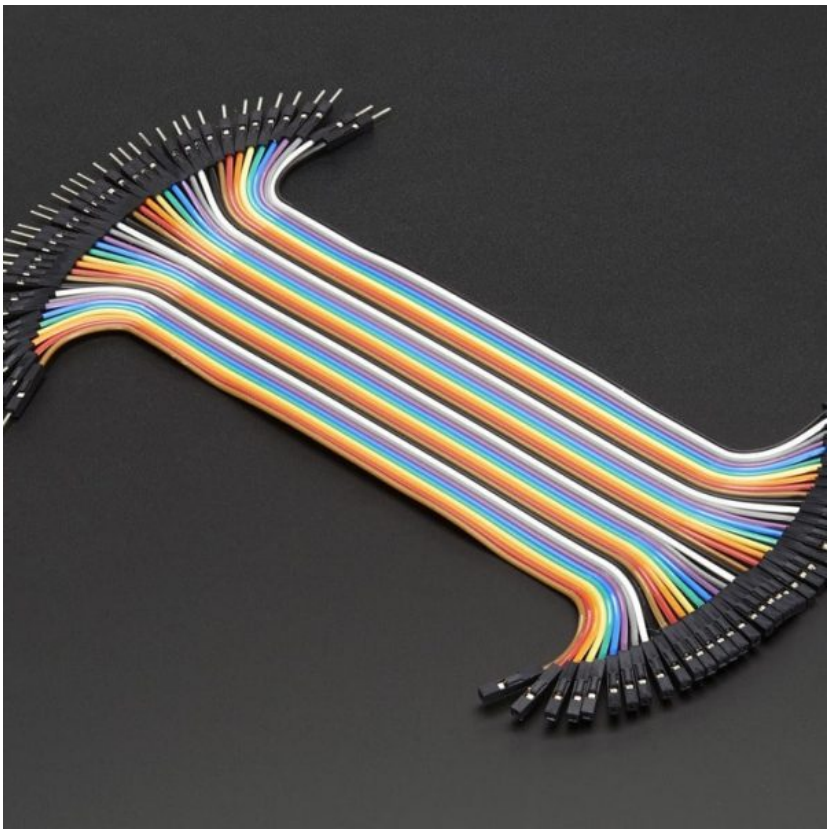
Arduino Uno R3



لوح تجارب حجم متوسط (Half size breadboard)



اسلاك توصيل ذكر/ذكر (Jumper Wires Male Male)

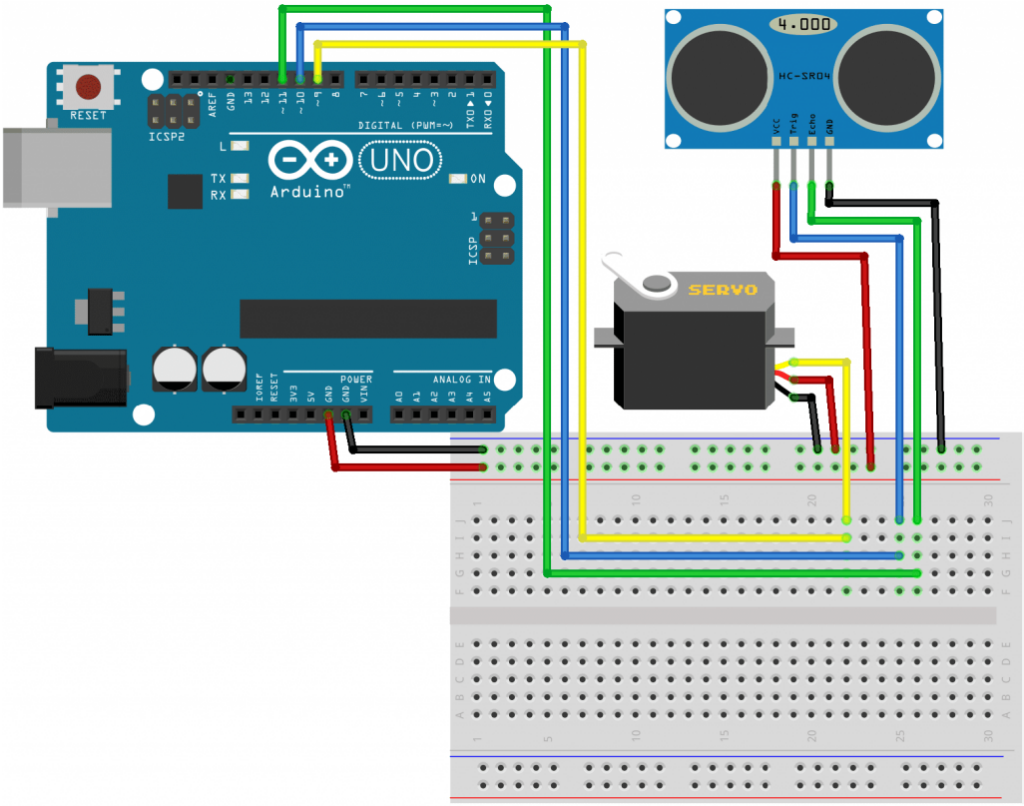


اسلاك توصيل أنثى/ذكر (Jumper Wires Female/male)

تصميم لوح التجارب

يقوم الاردوينو بالتحكم بزاوية دوران محرك السيرفو(Servo motor) ، لذلك يتم تثبيت حساس الموجات فوق الصوتية على المحرك ليتم قياس المسافة عند تلك الزاوية.

قم بتوصيل الدائرة الكهربائية على لوحة التجارب كما هو موضح بالصورة.



يتحتوي مستشعر الموجات فوق الصوتية على اربع منافذ، منفذ الطاقة Vcc و منفذ GND و منفذ Trig الذي يتم من خلاله ارسال النبضة, و منفذ Echo الذي يتم من خلاله استقبال صدى النبضة المرسلة من قبل Trig.

يتم توصيل مستشعر الموجات فوق الصوتية بالاردوينو كالتالي:

حساس الموجات فوق الصوتية	لوحة الاردوينو
Vcc	5v
GND	GND
Trig	Pin 10
Echo	Pin 11

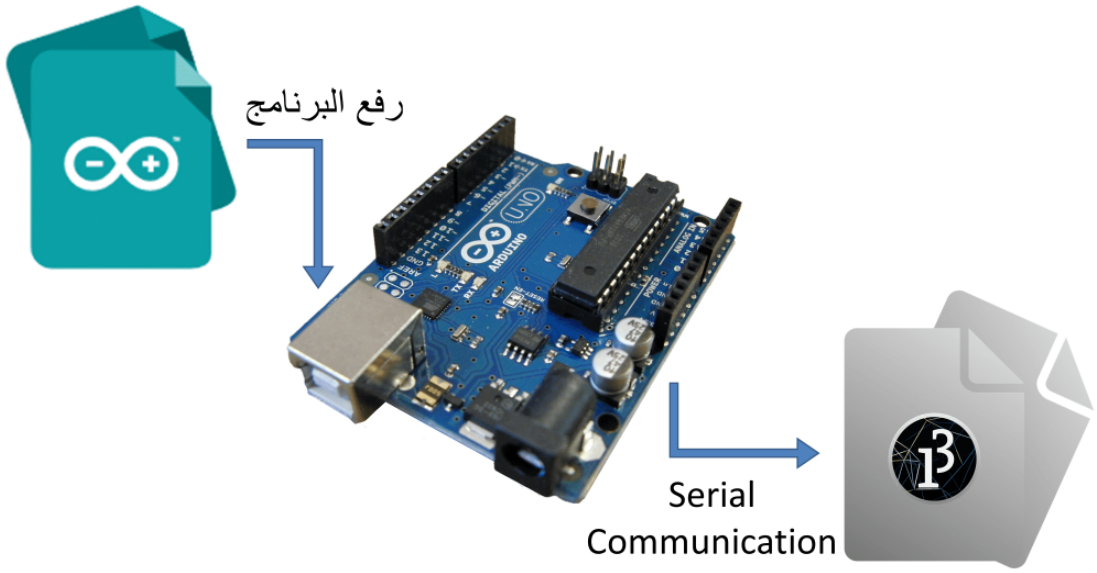
و توصيل محرك السيرفو كالتالي:

لوحة الاردوينو	محرك السيرفو
5v	السلك الأحمر
GND	السلك الأسود/البني
Pin 9	السلك الاصفر/البرتقالي

عندما يقوم الاردوينو بتحريك محرك السيرفو بدرجات معينة (و بالتالي يتحرك حساس الموجات الفوق صوتية) يتم حساب المسافة عند تلك الزاوية. سيتم ارسال هذة البيانات إلى جهاز الحاسوب عبر المنفذ التسلسلي (Serial port)، و بالتالي يقوم الحاسوب باستخدام برنامج لرسم الخريطة البيانية اعتمادا على هذة القيم .

واجهة المستخدم:

Processing IDE، هي بيئة تطوير متكاملة تم بناؤها من أجل الأشخاص المهتمين بالتصميم المرئي والفنون الإلكترونية الأخرى. حيث تمكنك من برمجة الرسوم و الصور المتحركة و تقوم بتحديث الرسومات المعروضه على شاشة الكمبيوتر و الاستجابة لتفاعل المستخدم.



الان نحن بحتاج لعمل برنامج يتم رفعه على الاردوينو الذي من شأنه تمكين التفاعل بين Arduino IDE و IDE Processing.

*يمكنك تنزيل برنامج Processing من هنا.

الكود البرمجي للأردوينو

قم برفع البرنامج التالي على الاردوينو:

```
#include <Servo.h>

Servo leftRightServo; // set a variable to map the servo
int leftRightPos = 0; // set a variable to store the servo position
const int numReadings = 10; // set a variable for the number of readings to take
int index = 0; // the index of the current reading
int total = 0; // the total of all readings
int average = 0; // the average
int echoPin = 11; // the SRF05's echo pin
int initPin = 10; // the SRF05's init pin
unsigned long pulseTime = 0; // variable for reading the pulse
unsigned long distance = 0; // variable for storing distance

/* setup the pins, servo and serial port */
void setup() {
```

```
  leftRightServo.attach(9);
// make the init pin an output:
  pinMode(initPin, OUTPUT);
// make the echo pin an input:
  pinMode(echoPin, INPUT);
// initialize the serial port:
  Serial.begin(9600);
}

void loop() {
  for(leftRightPos = 0; leftRightPos < 180; leftRightPos++) { // going left to
right.
    leftRightServo.write(leftRightPos);
    average=Avg();
    Serial.print("X"); // print leading X to mark the following value as degrees
    Serial.print(leftRightPos); // current servo position
    Serial.print("V"); // preceeding character to separate values
    Serial.println(average); // average of sensor readings
  }
/*
start going right to left after we got to 180 degrees
*/
  for(leftRightPos = 180; leftRightPos > 0; leftRightPos--) { // going right to left
    leftRightServo.write(leftRightPos);
    average=Avg();
    Serial.print("X");
    Serial.print(leftRightPos);
    Serial.print("V");
    Serial.println(average);
  }
}

long Avg()
{
  for (index = 0; index<=numReadings;index++) {
  digitalWrite(initPin, LOW);
  delayMicroseconds(50);
  digitalWrite(initPin, HIGH);
  delayMicroseconds(50);
  digitalWrite(initPin, LOW);
  pulseTime = pulseIn(echoPin, HIGH);
  distance = pulseTime/58;
  total = total + distance;
  delay(10);
}
average = total/numReadings; // create average reading

if (index >= numReadings) { // reset the counts when at the last item of the array
index = 0;
total = 0;
}
return average;
}
```

لمحة عن الكود :

يقوم محرك السيرفو بالدوران من 0 إلى 180 درجة و العكس ، ومن خلاله يقوم مستشعر الموجات فوق الصوتية بالمسح الأرضي و الجوي لما يجري داخل منطقة محدودة وهي ابعد ما يستطيع المستشعر التعامل معه.

دالة for تقوم بتغير قيمة زاوية الدوران للمحرك :

```
for(leftRightPos = 0; leftRightPos < 180; leftRightPos++) {
}
```

```
for(leftRightPos = 180; leftRightPos > 0; leftRightPos--){
}
```

تقوم الدالة ()AVG اعتمادا على القيم التي يتلقاها حساس الموجات فوق الصوتية بحساب المسافة التي يبعد عنها الجسم عن الحساس :

```
long Avg()
{
  for (index = 0; index<=numReadings;index++) {
  digitalWrite(initPin, LOW);
  delayMicroseconds(50);
  digitalWrite(initPin, HIGH);
  delayMicroseconds(50);
  digitalWrite(initPin, LOW);
  pulseTime = pulseIn(echoPin, HIGH);
  distance = pulseTime/58;
  total = total + distance;
  delay(10);
}
average = total/numReadings; // create average reading

if (index >= numReadings) { // reset the counts when at the last item of the array
index = 0;
total = 0;
}
return average;
}
```

* يمكنك الاطلاع على مشروع نظام الحسّاسات لاصطفاف السيارة لفهم الكود المتعلق بحساس الموجات فوق الصوتية و حساب المسافة , و الدرس الخامس عشر للاطلاع على شرح الكود المتعلق بمحرك السيرفو.

يتم ارسال قيمة المسافة و الزاوية عند هذة القيمة إلى الحاسوب فيقوم البرنامج (Processing IDE) برسم خطوط الرادار باستعمال الدوال المثلثة طبقا للمعلومات المستقبله.

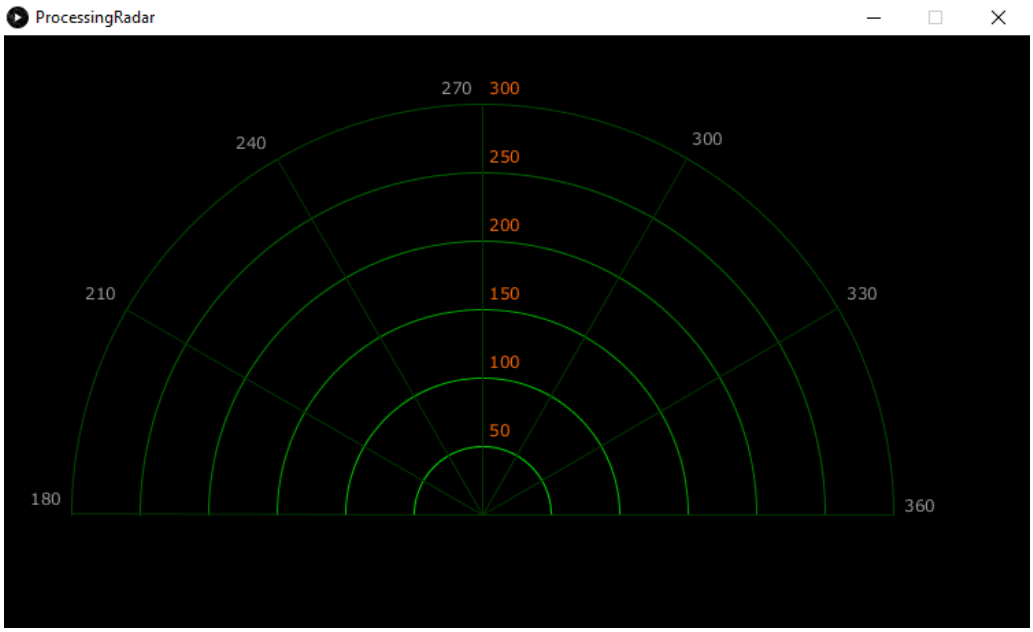
```
Serial.print("X");
Serial.print(leftRightPos);
Serial.print("V");
Serial.println(average);
```

برمجة الواجهة الرسومية

الحاسوب بعد ارسال قيم الزاوية و المسافة من الاردوينو الى (processing IDE) ، يتم استقبالها بإستخدام دالة SerialEvent() .
تقوم SerialEvent() بقراءة البيانات من المنفذ التسلسلي (serial port) ثم نقوم بوضع قيم الزاوية و المسافة في متغيرات (degree,value). هذة المتغيرات سيتم استخدامها لرسم الرادار، و الخطوط، و الكشف عن الأجسام ، وتغير النصوص .

لعرض البيانات على الشاشة يتم انشاء Arrays لتخزين القيم الحديثة على newValue Array و عمل تحديث للقيم القديمة على oldValue Array . لأن الموقع اتباعا لحركة محرك السيرفو يتغير بإستمرار ، فسنفقد البيانات القديمة التي سيتم عرضها على الشاشة.

```
int[] newValue = new int[181];
int[] oldValue = new int[181];
```



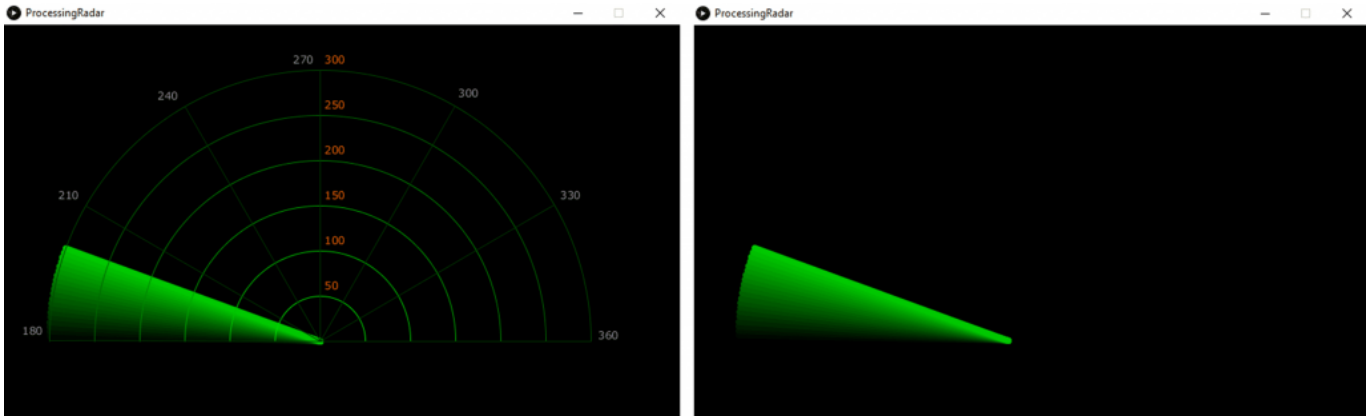
لرسم الرادار سيتم كتابة الداله drawRadar() التي تتكون من دوال arc() و line().

```
void drawRadar() {

 for (int i = 0; i <=6; i++){
  noFill();
  strokeWeight(1);
  stroke(0, 255-(30*i), 0);
  arc(radius, radius, (100*i), (100*i),PI,TWO_PI);
  fill(250, 103, 0);
  noStroke();
  text(Integer.toString(radarDist+50), 380, (305-radarDist), 50, 50);
  radarDist+=50;
 }

 radarDist = 0;
 for (int i = 0; i <= 6; i++) {
   strokeWeight(1);
   stroke(0, 55, 0);
   line(radius, radius, radius + cos(radians(180+(30*i)))*w, radius +
sin(radians(180+(30*i)))*w);
```

```
   fill(153, 153, 153);
   noStroke();
   if (180+(30*i) >= 300) {
    text(Integer.toString(180+(30*i)), (radius+10) +
cos(radians(180+(30*i)))*(w+10), (radius+10) + sin(radians(180+(30*i)))*(w+10),
25,50);
   }
  else {
   text(Integer.toString(180+(30*i)), radius + cos(radians(180+(30*i)))*w, radius +
sin(radians(180+(30*i)))*w, 60,40);
  }
 }
}
```



ليتم عمل مسح للرادار يتم رسم الخط الذي يتحرك جنب إلى جنب اتباعا لحركة محرك السيرفو بإستخدام الدالة setupSweep() .

تستخدم الدالة setupSweep() الدالة line() التي تستخدم المتغير degree لإعادة رسم الخط لكل درجة.

```
void setupSweep(){
 strokeWeight(7);
 if (motion == 0) {
 for (int i = 0; i <= 20; i++) {
 stroke(0, (10*i), 0);
 line(radius, radius, radius + cos(radians(degree+(180+i)))*w, radius +
sin(radians(degree+(180+i)))*w);
 }
 } else { // if going right to left
 for (int i = 20; i >= 0; i--) {
 stroke(0,200-(10*i), 0);
 line(radius, radius, radius + cos(radians(degree+(180+i)))*w, radius +
sin(radians(degree+(180+i)))*w);
 }
 }
}
```

لإعداد تحديث الأشكال يتم عمل الدالة SetupShapes() . سنقوم بإستخدام الدالة For loop للتحرك بين القيم التي تم تخزينها بالـ Arrays (newValue, oldValue)

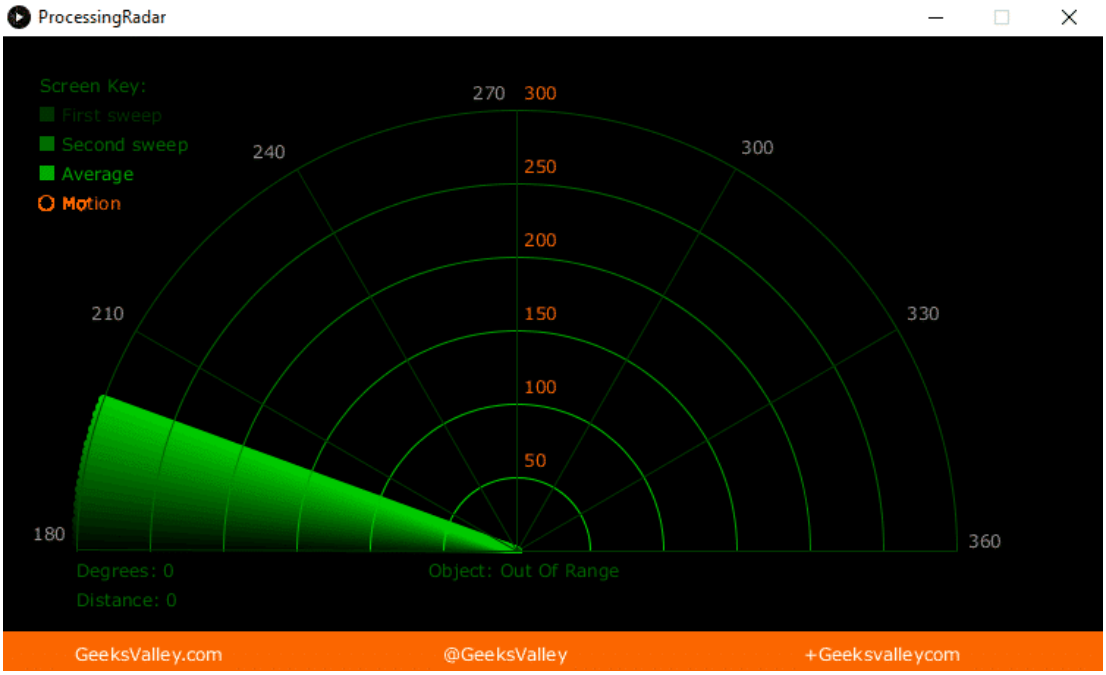
```
void SetupShapes(){
  noStroke();
```

```
  fill(0,50,0);
  beginShape();
  for (int i = 0; i < 180; i++) {
  x = radius + cos(radians((180+i)))*((oldValue[i]));
  y = radius + sin(radians((180+i)))*((oldValue[i]));
  vertex(x, y);
  }
  endShape();
  fill(0,110,0);
  beginShape();
  for (int i = 0; i < 180; i++) {
  x = radius + cos(radians((180+i)))*(newValue[i]);
  y = radius + sin(radians((180+i)))*(newValue[i]);
  vertex(x, y);
  }
  endShape();
  fill(0,170,0);
  beginShape();
  for (int i = 0; i < 180; i++) {
  x = radius + cos(radians((180+i)))*((newValue[i]+oldValue[i])/2);
  y = radius + sin(radians((180+i)))*((newValue[i]+oldValue[i])/2);
  vertex(x, y);
  }
  endShape();
}
```

لرسم موقع الاجسام التي تم رصدها ، نقوم بعمل الدالة drawObject() .التي تستخدم المسافة الملتقطة من مستشعر الموجات فوق الصوتية و بالاشتراك مع الزاوية لرسم الجسم على الرادار.

```
void drawObject() {
  if (firstRun >= 360) {
  stroke(250,103,0);
  strokeWeight(1);
  noFill();
  for (int i = 0; i < 180; i++) { if (oldValue[i] - newValue[i] > 35 || newValue[i]
- oldValue[i] > 35) {
  x = radius + cos(radians((180+i)))*(newValue[i]);
  y = radius + sin(radians((180+i)))*(newValue[i]);
  ellipse(x, y, 10, 10);
  }
  }
  }
}
```

وللإطلاع على القيم التي تم استقبالها و النصوص تم إنشاء الدالة drawText() .



يمكنك تنزيل كود الـ (Processing) المتعلق بالواجهة الرسومية للرادار من هنا.