TRABAJO DE INFORMÁTICA

(2018-2019)

RADAR DE OBJETOS

Miembros:

- Natalia Sánchez Sánchez (54206).
- Jorge Ortega Mateo (54134).

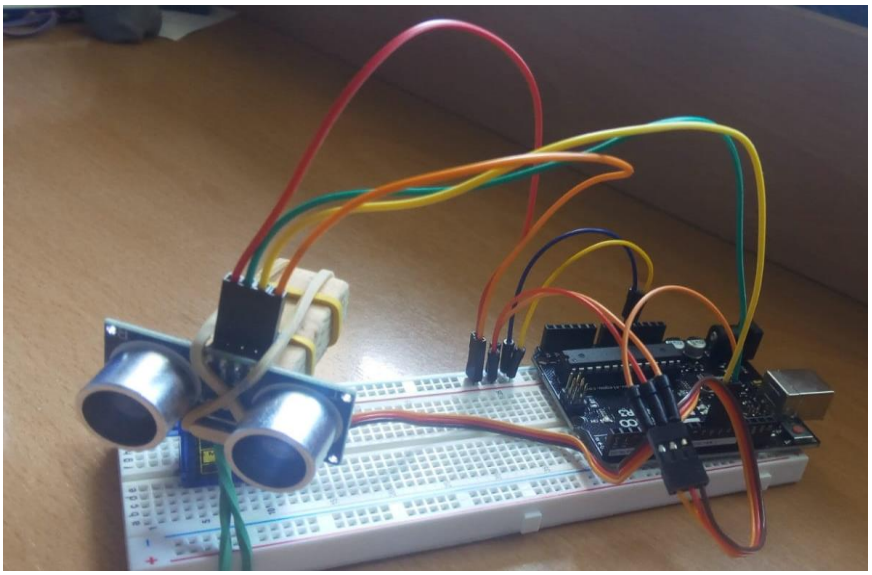
Profesor: Joaquín González Gigosos.
Grupo: A-109

Fecha de entrega: 17 / 05 / 2019

Título y Resumen

Radar de objetos.

Con un sensor ultrasonidos buscamos localizar los objetos que se encuentran a menos de 40cm y mostrarlo en pantalla en un diseño de radar imitando al de un buque marino. Para ello requeriremos de los programas Arduino y Processing.



Requisitos Funcionales

1º El radar comienza en un ángulo de 0º y rotara hasta los 180º y veremos este seguimiento en pantalla mostrado por una línea verde.
2º En esta primera pasada detectara cualquier objeto a una distancia inferior a 40cm (La distancia se puede variar en función del radar) y lo mostrara en pantalla con una marca roja.
3º Una vez llegado hasta los 180º volverá hasta su posición inicial barriendo de nuevo la zona.
4º Las marcas realizadas en el primer barrido se eliminarán y se actualizarán los datos en función a si el objeto ha variado su distancia o posición.

Hardware y Fundamentos Teóricos

Podemos definir un radar como un Sistema electrónico de localización de objetos empleado en diversas áreas (aeronáutica, navegación, astronomía, etc.), que sirve para indicar la presencia de un objeto y determinar la distancia a la que se halla y su posición desde un punto fijo, mediante la emisión de ondas de altísima frecuencia reflejadas en él.
http://buscon.rae.es/dpd/srv/search?id=Hb9zYUnF5D6GlXiZyY

SENSOR DE ULTRASONIDOS
Como su nombre lo indica, los sensores ultrasónicos miden la distancia y la posición de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde el objeto. Los sensores ultrasónicos miden la distancia al objeto contando el tiempo entre la emisión de onda y la recepción.
SERVO MOTOR
Un servomotor es un tipo especial de motor que permite controlar la posición del eje en un momento dado. Está diseñado para moverse una determinada cantidad de grados y luego mantenerse fijo en una posición.
PLACA ARDUINO UNO R3
Para conectar todos los elementos utilizamos una sencilla placa con entradas y salidas, analógicas y digitales, que nos permite comunicar el servo y el sensor con el ordenador a través de los códigos programados.

Desarrollador del software utilizado

Como ya hemos mencionado anteriormente, los desarrolladores seleccionados para llevar acabo nuestro trabajo son Arduino y Processing 3.
Arduino es una plataforma de hardware y software de código abierto, en un entorno de desarrollo que está basado en el lenguaje de programación Processing. Es decir, una plataforma de código abierto para prototipos electrónicos.

Processing es un software basado en Java y, por lo tanto, multiplataforma que fue diseñadores, para las animaciones y aplicaciones gráficas de todo tipo.

Códigos de nuestro radar

Nuestro código del radar se divide en un primer código Arduino y un segundo código hecho en Processing.
En esta primera imagen podemos observar la primera parte del código Arduino en el cual definimos las variables, establecemos la comunicación entre los distintos programas y generamos la primera barrida del radar.

```
#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 procesing).
  }
```

En esta segunda imagen realizamos la segunda barrida del radar y establecemos el cálculo de distancias.

```
 //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 las distancia de los objetos a partir de la duración al encontrar dicho objeto y la velocidad de onda en encontrarlo
  return distancia;
}
```

Esta última información es enviada al código de Processing (al igual que el ángulo de giro) para que se pueda mostrar en pantalla.

A continuación, el código desarrollado en Processing:

radar__proc_

```
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);
```

Primero definiremos todas las variables a utiliza y añadiremos las librerías necesarias; diseñamos el tamaño aproximado de la pantalla y establecemos la comunicación con el Arduino.

radar__proc_

```
  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 incice1 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);
```

Recibe los datos de la placa Arduino y los muestra en pantalla.

radar__proc_

```
  // 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);
  //limitanmos 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)));
  }
```

Diseñamos el radar, asemejándolo al utilizado por los buques, creamos la semicircunferencia y los diferentes arcos, luego diseñamos como aparecen los objetos en pantalla.

radar__proc_

```
  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);

```

radar__proc_

```
111  text("10cm",width-width*0.3854,height-height*0.0833);
112  text("20cm",width-width*0.281,height-height*0.0833);
113  text("30cm",width-width*0.177,height-height*0.0833);
114  text("40cm",width-width*0.0729,height-height*0.0833);
115  textSize(40);
116  text("Radar Ultrasonido", width-width*0.875, height-height*0.0277);
117  text("Angulo: " + angAngulo +" °", width-width*0.48, height-height*0.0277);
118  text("Dist:", width-width*0.26, height-height*0.0277);
119  if(angDistancia<50) {
120  text("        " + angDistancia +" cm", width-width*0.225, height-height*0.0277);
121  }
122  textSize(25);
123  fill(98,245,60);
124  translate((width-width*0.4994)+width/2*cos(radians(30)),(height-height*0.0907)-width/2*sin(radians(30)));
125  rotate(-radians(-60));
126  text("30°",0,0);
127  resetMatrix();
128  translate((width-width*0.503)+width/2*cos(radians(60)),(height-height*0.0888)-width/2*sin(radians(60)));
129  rotate(-radians(-30));
130  text("60°",0,0);
131  resetMatrix();
132  translate((width-width*0.507)+width/2*cos(radians(90)),(height-height*0.0833)-width/2*sin(radians(90)));
133  rotate(radians(0));
134  text("90°",0,0);
135  resetMatrix();
136  translate(width-width*0.513+width/2*cos(radians(120)),(height-height*0.07129)-width/2*sin(radians(120)));
137  rotate(radians(-30));
138  text("120°".0.0);
```

```
138    text("120°",0,0);
139    resetMatrix();
140    translate((width-width*0.5104)+width/2*cos(radians(150)),(height-height*0.0574)-width/2*sin(radians(150)));
141    rotate(radians(-60));
142    text("150°",0,0);
143    popMatrix();
144  }
145
```

Visualización de la pantalla

A continuación, mostraremos como se ven en la pantalla los objetos localizados por el radar.