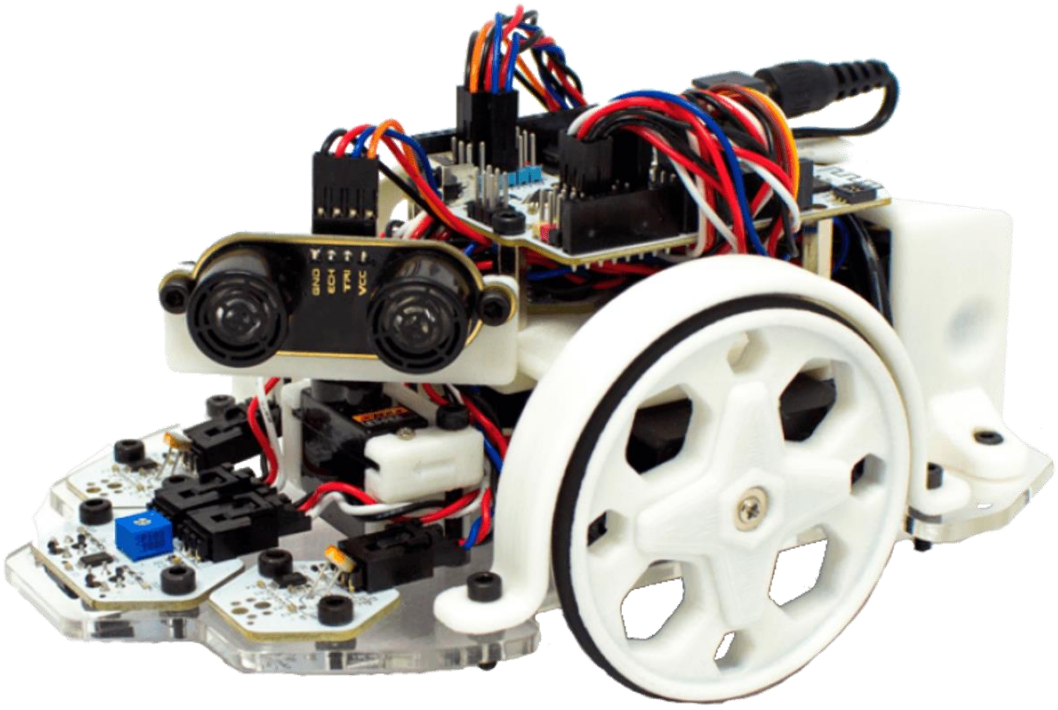


LINEFOLLOWER

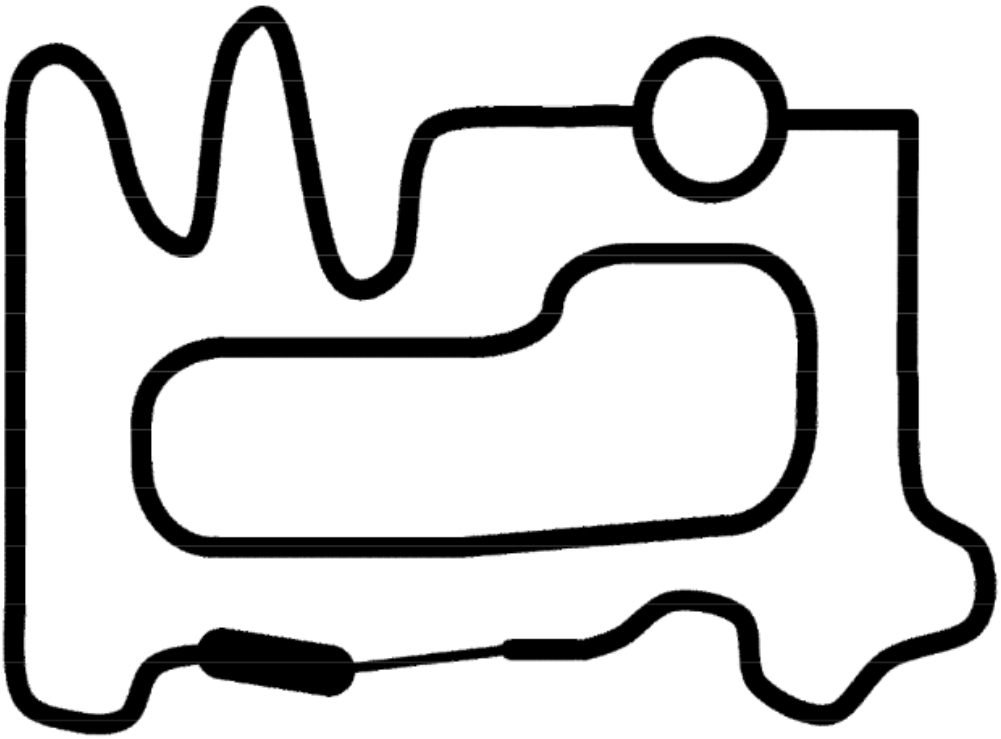
COMPUTER'S STRUCTURE PROJECT

Pablo Rodríguez Solera//Juan Mena Patón (1º A)
ESCUELA SUPERIOR DE INFORMÁTICA |

INDEX:

INTRODUCTION

The computer structure project of 2017 is based on programming a robot with an Arduino processor and different components such as infrared sensors and servos (in addition to buzzer and ultrasonic sensor) to follow a black line in a plane that emulates a circuit.



The subject department provides a base code which causes the robot to complete the inner circuit.
Our goal is for the robot to do both circuits.

STEPS TO FOLLOW

The main problem of the robot is that if you leave the path, with the base code is stopped, so we have implemented a system with if / else.

```
if (ir_derecho == NEGRO && ir_izquierdo ==NEGRO) {
   pwm.setPWM(servo_izquierdo, 0, SERVOMIN);
   pwm.setPWM(servo_derecho, 0, SERVOMAX);
  }

 else if (ir_derecho == NEGRO && ir_izquierdo != NEGRO){
   pwm.setPWM(servo_izquierdo, 0, SERVOMIN);
   pwm.setPWM(servo_derecho, 0, SERVOSTOP);
   tone(zumbador, tonos[0]);
   ultim=1;
 }
```

We also declare a variable (*ultim*) for the robot to return to the circuit where it left.
The variable saves the last position where the robot detected the black line before it exits.
This variable is changed in each of the if, so that it will serve in others if as a conditioner for the operations of the robot.

```
else if (ir_derecho != NEGRO && ir_izquierdo != NEGRO && ultim==2){
```

Thus, we get the robot does not stop, but will return to the circuit. Once the robot manages to make the whole circuit without going out, we are asked a series of secondary characteristics.

```
int ultim;
```

1. **Add sound when one of the sensors does not detect the black line:**
   For this we declare the buzzer and we introduce the *tone* function in the necessary if, in addition to an array with two numbers that will serve so that the tone is not always the same:

```
tone(zumbador, tonos[1]);
```

```
pinMode( 10 , OUTPUT);
```

```
int zumbador=10;
```

```
const int tonos[] = {261,494};
```

2. **Reduce the time the robot makes the circuit:**

For this, we increase the value of the pulse of the servos.

```
#define SERVOMIN  180 // (180) this is the 'minimum' pulse length count (out of 4096)
#define SERVOSTOP 340 // pulse length to stop de servo
#define SERVOMAX  500 // (500) this is the 'maximum' pulse length count (out of 4096)
```

3. **Detect obstacles to stop the robot or make them surround, using the ultrasonic sensor:**

This part was not able to integrate it into the main code, so we attach it in a separate file.

This program shows the distance to the object closest to the ultrasound sensor.
The program sends a pulse through the ultrasound output pin.

```
void loop(){
  digitalWrite(salida, HIGH);
  delayMicroseconds(10);
  digitalWrite(salida, LOW);
```

```
int entrada = 5;
```

Then measure the response time:

```
tiempo = (pulseIn(entrada, HIGH)/2);
```

Finally we calculate the distance multiplying the time by the speed of the sound and we print it in the monitor:

```
Serial.println(distancia);
distancia = float(tiempo * 0.0343);
```