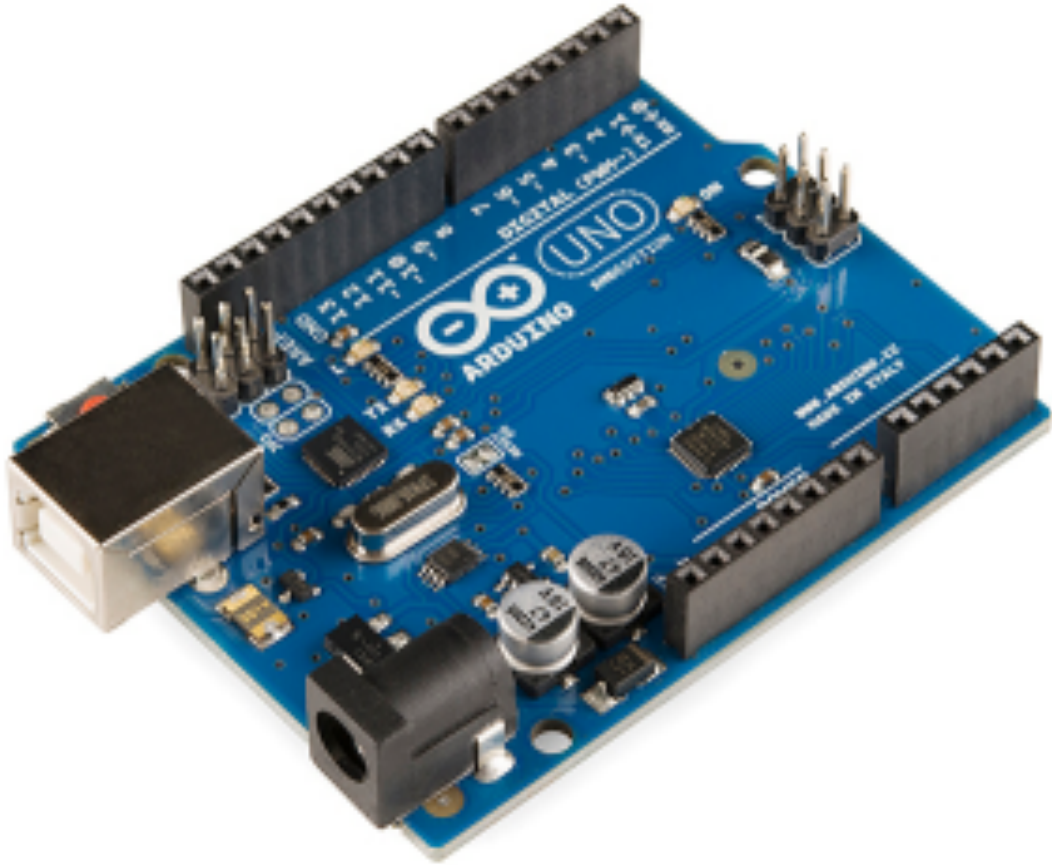


LINE FOLLOWER

Eco_LAB



Abraham López de Lerma
Mario Sanz González
Enrique Rodríguez Villafranca

INDEX

DESCRIPTION.....	3
ROBOT	3
CIRCUIT	3
ADDING FUNCIONALITIES.....	4
BUZZER	5
ULTRASOUND	5
LIGHT FOLLOW.....	6

DESCRIPTION

ROBOT

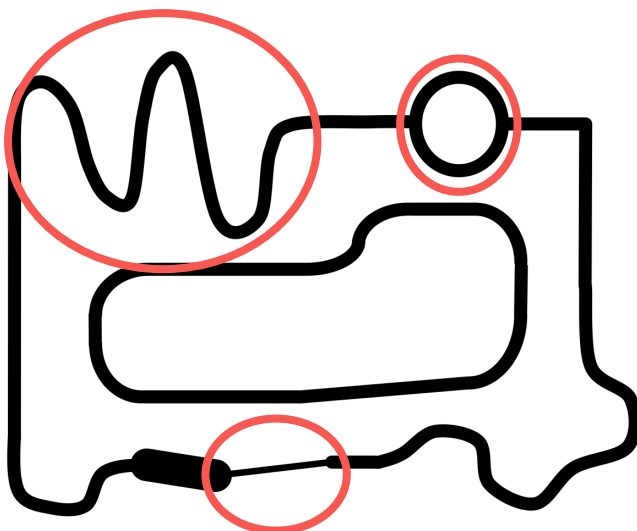
The line follower is a project that consists on programming a robot with the Arduino Board and the BQ-based robot. The tools we need to do this is the Arduino IDE software, where we implement the code in assembler language for the robot to follow the line and make other actions, and the Adafruit PWM Servo Driver Library.

The program INPUTS we used were the two digital inputs in pins 2 and 3, to connect IR sensors. The pin 4 for the ultrasound sensor (echo).

The program OUTPUTS we used were channels 0 and 1 of the PWM shield board, to manage the two servos. The pin 4 for the ultrasound sensor (trigger). The pin 8 for the buzzer included in the robot.



CIRCUIT



This is the circuit that the robot has to follow. There are two parts: an internal and an external. The internal one is the basic. It's used for the first parts of the project. The external one is for

upgrading the robot. There are some difficult parts, that are shown on the picture.

The robot has to follow the black line of the circuit and if the robot goes out of the line, he has to find the line again and continue following the line. In the red zones we have the following problems:

- In the circle, the robot sometimes doesn't recognize the line and goes around every time, or sometimes come back by the entrance.
- In the continues closed curves, sometimes the robot doesn't make the entire curve and go back, or just enter on the internal circuit.
- In the thin line, which is the hardest part of the circuit because the sensors have to be good calibrated. If the sensors aren't good calibrated, the robot sometimes detect the line but can't go ahead and sometimes go back.

ADDING FUNCTIONALITIES

Once we have solved the problems and the robot follows the entire circuit, we tried to introduce some extra functionalities like buzzer or ultrasound to avoid obstacles.

BUZZER

We tried to program the buzzer to sound when the robot couldn't find the line. We tried to implement it with the Pirates of the Caribbean's theme, but when we upload the program with the melody implemented, the robot doesn't move and only plays the melody, so we decided to implement the original sound of the buzzer.

```
#define NOTE_C4 262 //Defining note frequency
#define NOTE_D4 294
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_G4 392
#define NOTE_A4 440
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_D5 587
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_G5 784
#define NOTE_A5 880
#define NOTE_B5 988
```

ULTRASOUND

We have tried to implement the ultrasound to the robot to avoid possible obstacles that can be on the circuit. We defined the pins of the ultrasound: pin 4 for the trigger and pin 5 for the echo.

```
digitalWrite(TRI_Pin, LOW); // We secure the trigger is active
delayMicroseconds(10); // Trigger is LOW
digitalWrite(TRI_Pin, HIGH); // Activate the output pulse
delayMicroseconds(10); // We wait 10 microseconds, the pulse is still active
digitalWrite(TRI_Pin, LOW); // We cut the pulse and wait for the echo
duracion = pulseIn (ECH_Pin, HIGH);
distancia = duracion/58;
delay(500);
```

- "digitalWrite(TRI_Pin, LOW);" is to make sure that the trigger is activated.
- "delayMicroseconds(10);" is to make sure that the trigger is LOW.
- "digitalWrite(TRI_Pin, HIGH);" is to activate the output pulse.
- "delayMicroseconds(10);" is for wait 10 ms and the pulse is still active.

- “digitalWrite(TRI_Pin, LOW);” is for cut the pulse and wait for the echo.

When we upload the code to the arduino’s robot, we have problems with the “servo 180”.

LIGHT FOLLOW

We tried to implement a program that makes the robot follow a light instead of the line. We added to the code two new variables: “int Light_left” and “int Light_right”.

```
void loop() {  
  int ir_derecho=digitalRead(3);  
  int ir_izquierdo=digitalRead(2);  
  Light_left=analogRead(A0);  
  Light_right=analogRead(A1);
```

The “analogRead()” is used to know where is the light come from, and and “analogWrite()” to follow it.

But like in the ultrasound, we have problem with the “servo 180” when we tried to upload the code.