

# CONTROL DE PARKING



## STEMJAM Teaching Guide

Developing make spaces to promote creativity  
around STEM in schools

Acronym: STEMJAM

Project no. 2016-1-ES01-KA201-025470

[www.stemjam.eu](http://www.stemjam.eu)



Co-funded by the  
Erasmus+ Programme  
of the European Union

# CONTROL DE PARKING

## RESUMEN

La actividad consiste en crear una barrera como el aparcamiento. Combinando arduino con mBot.

## OBJETIVOS DIDÁCTICOS

### TECNOLOGÍA

- ❖ Desarrollar el código para la barrera.

### INGENIERÍA

- ❖ Construir el sistema con Arduino y la casa de la barrera.

Materia STEM:

Ciencia

Tecnología

Ingeniería

Matemáticas

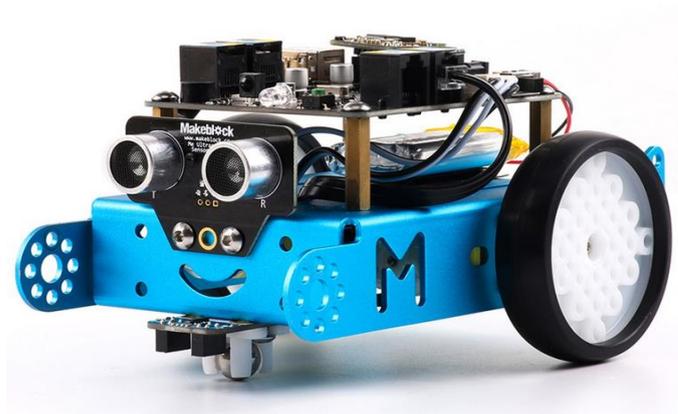
Nivel educativo:

12-14 años

14-16 años

## LISTADO DE MATERIALES

- ❖ mBot



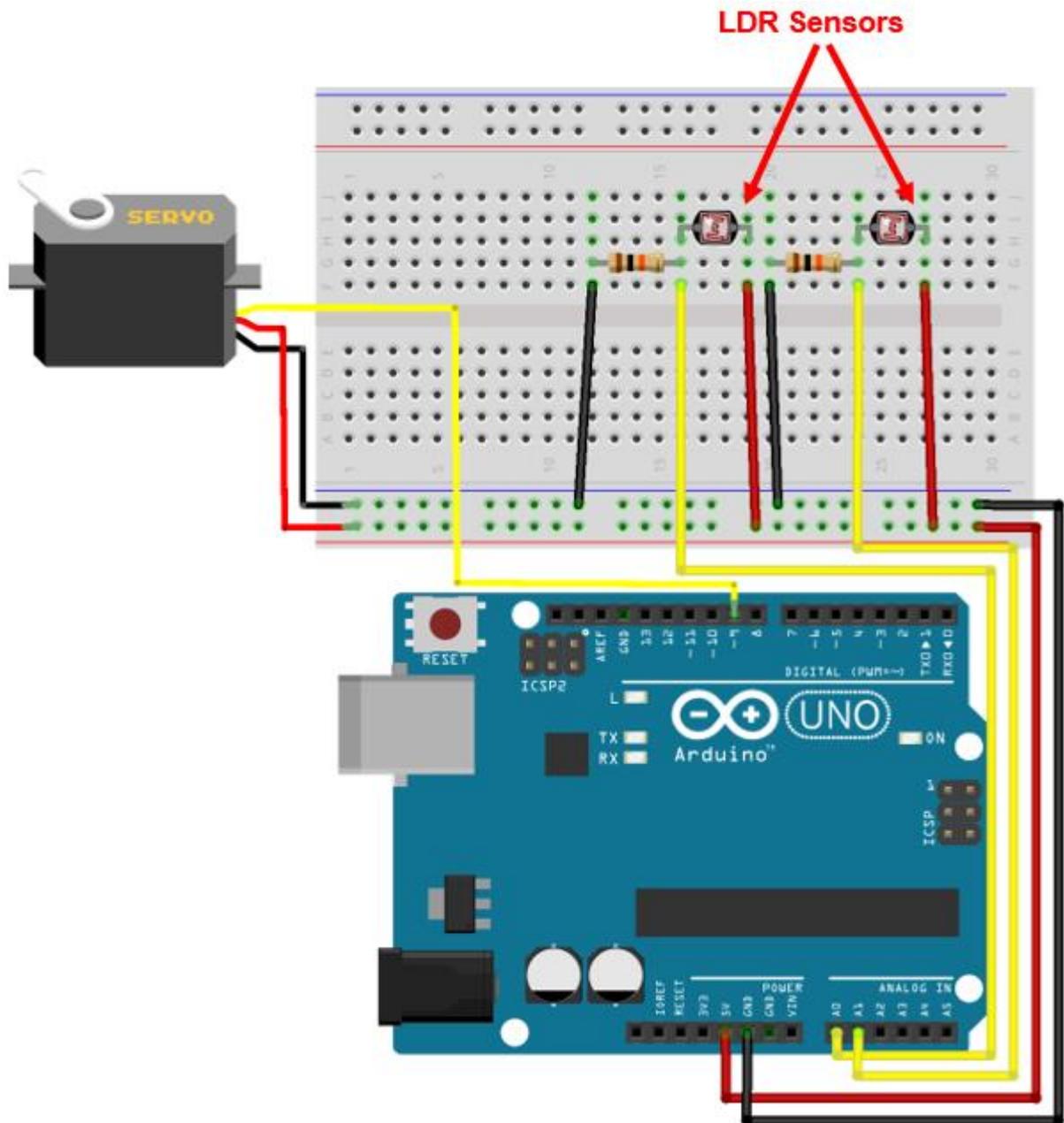
- ❖ Arduino Uno Shield.
- ❖ Protoboard.
- ❖ (2x) Sensores LDR.
- ❖ (2x) Resistencias de 10k ohms.
- ❖ Servo motor.
- ❖ Material para construir la estructura, por ejemplo, piezas de Lego.



## DESCRIPCIÓN DE LA ACTIVIDAD

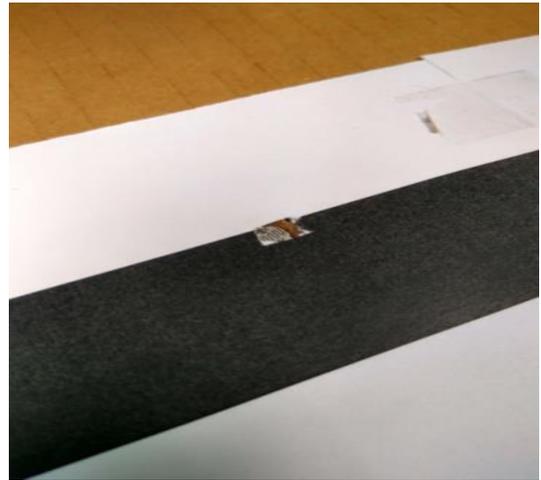
Este proyecto podría combinarse con el sigue líneas de mBot. El mBot sigue la línea negra, y cuando llega a la barrera, éste se detendrá y la barrera se abrirá. El mBot podría seguir la línea negra.

Para visualizar mejor los trabajos del Sistema Arduino, podemos ver un ejemplo de la estructura:

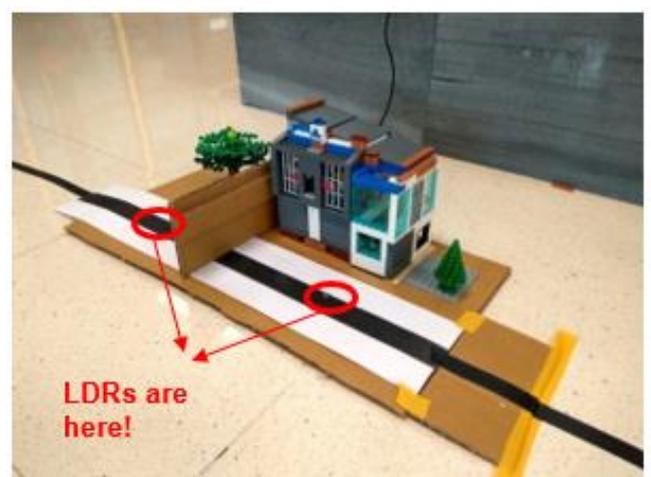
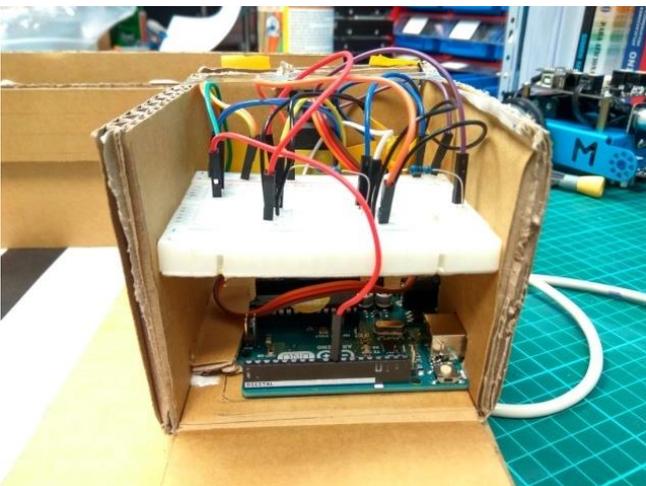
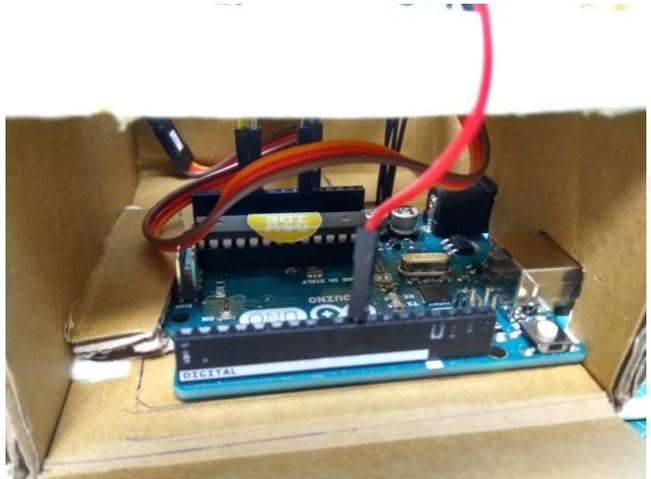
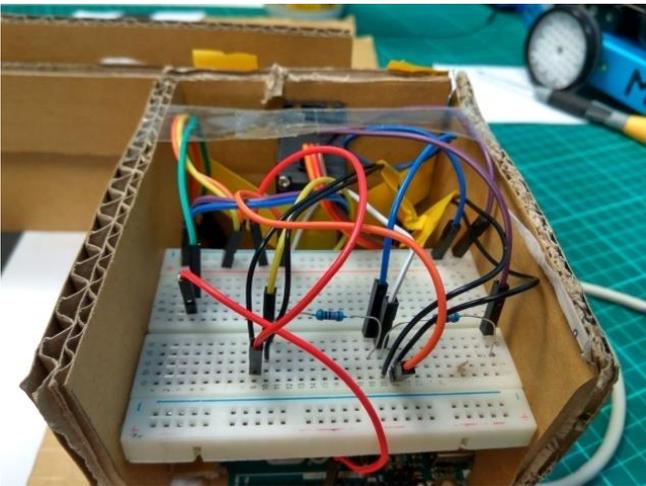




1 Línea negra

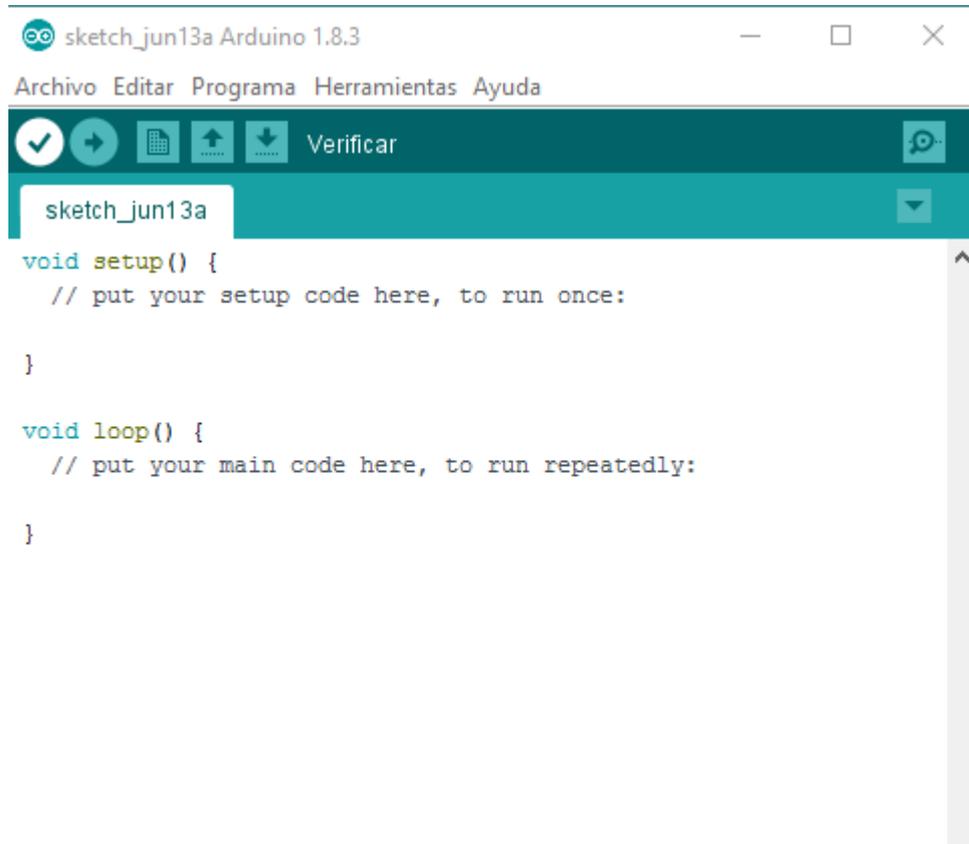


2 Sensores LDR



Una vez que hayamos establecido la Estructura de Arduino, desarrollaremos el código de la Barrera.

Para hacer esto, utilizaremos el software Arduino IDE (<https://www.arduino.cc/en/main/software>)



```
sketch_jun13a Arduino 1.8.3
Archivo Editar Programa Herramientas Ayuda
Verificar
sketch_jun13a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Procedemos a realizar la programación:

1. Importe la biblioteca del servo:

```
#include <Servo.h>
```

2. Crear e iniciar las variables:

```
Servo myservo;
int pos = 1;
int LDR1 = A0;
int LDR2 = A1;
int value1;
int value2;
```



3. Dentro de la función "setup ()" configuramos la configuración de la placa:

```
void setup() {  
  Serial.begin(9600);  
  pinMode(LDR1, INPUT);  
  pinMode(LDR2, INPUT);  
  myservo.attach(9);  
  myservo.write(1);  
}
```

4. En la función "loop()", el código repetirá las instrucciones todo el tiempo:

```
void loop() {  
  value1 = analogRead(LDR1);  
  value2 = analogRead(LDR2);  
  Serial.print(value1);  
  Serial.print("|");  
  Serial.println(value2);  
  
  if (value1 < 780 && pos<=5 && value2 >900) {  
    for (pos = 1; pos <= 90; pos += 1) { // goes from 0 degrees to 180 degrees  
      // in steps of 1 degree  
      myservo.write(pos);           // tell servo to go to position in variable 'pos'  
      delay(15);                    // waits 15ms for the servo to reach the position  
    }  
  }  
  
  if (value2 < 820 && pos>=85 && value1 >870) {  
    for (pos = 90; pos >= 1; pos -= 1) { // goes from 180 degrees to 0 degrees  
      myservo.write(pos);           // tell servo to go to position in variable 'pos'  
      delay(15);                    // waits 15ms for the servo to reach the position  
    }  
  }  
}
```

Programaremos el mBot, para que cuando encuentre un obstáculo a menos de 10 centímetros que se detenga, y cuando no detecte nada, continuará avanzando.

