

1.2.17. Sensor Analógico: LDR

En este apartado vamos a realizar dos actividades en donde utilizaremos un sensor de luz visible o LDR. Este tipo de sensores son analógicos, nos dan una medida de intensidad de la luz que reciben, por eso los conectaremos a entradas analógicas de nuestra tarjeta de control.

1.2.17.1. Lectura del nivel de Luz

La primera actividad va a consistir en una simple lectura del nivel de luz que recibe el sensor y su envío por el puerto serie para visualización en un PC.

Componentes

- Placa ZUM BT o Arduino UNO compatible
- Sensor LDR del kit de robótica de BQ u otro cualquiera
- Cable USB

Conexionado

- Sensor LDR—PIN A3
- USB PC—Arduino

Abrimos un nuevo proyecto en Bitbloq y añadimos los componentes y sus conexiones como aparece en la

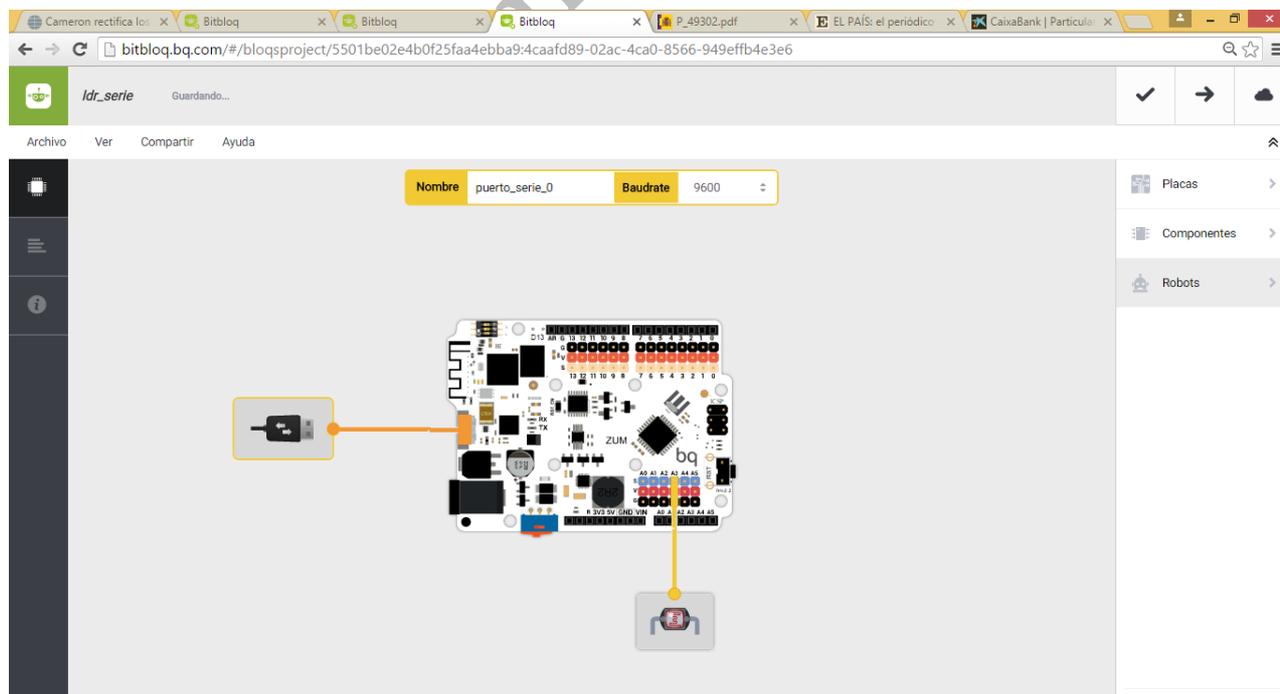


Figura 1.2.17-1 Conexionado del LDR en Bitbloq

Programación

La programación de esta actividad es muy sencilla ya que solo realizamos una lectura analógica y la enviamos al puerto serie como aparece en la Figura 1.2.17-2

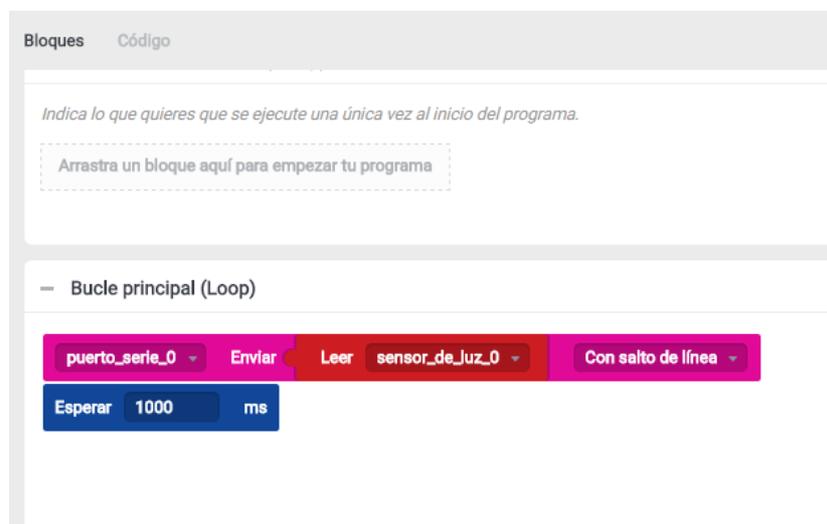


Figura 1.2.17-2 Programación por bloques de la lectura de un LDR y su envío por puerto serie.

El código en Arduino equivalente será:

```
/** Included libraries */
#include <SoftwareSerial.h>
#include <BitbloqSoftwareSerial.h>

/** Global variables and function definition */
int sensor_de_luz_der = A3;
BitbloqSoftwareSerial puerto_serie_0(0, 1, 9600);

/** Setup */
void setup() {
  pinMode(sensor_de_luz_der, INPUT);
}

/** Loop */
void loop() {
  puerto_serie_0.println(analogRead(A3));
}
```

Ya solo tenemos que subir el programa a la placa y abrir un monitor serie para ver las lecturas de luz de nuestro sensor como por ejemplo mostramos en la Figura 1.2.17-3

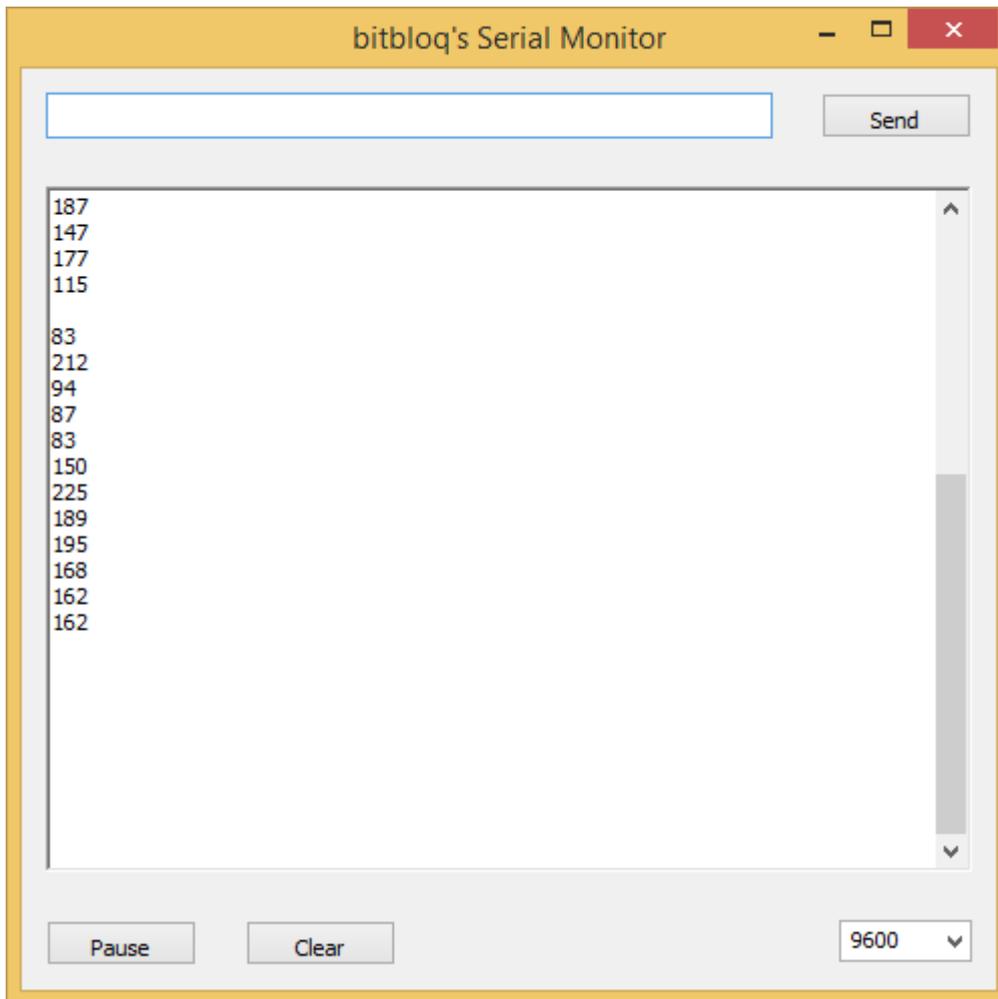


Figura 1.2.17-3 Valores mostrados del LDR por el monitor serie.

1.2.17.2. Sigue una linterna

En esta actividad haremos que el robot siga una linterna a modo de faro. Es parecido a lo que hacen algunos robots, como por ejemplo el robot limpiador Roomba con el uso de balizas activas que le indican el camino a seguir

Componentes

- Robot Printbot o cualquier con los siguientes componentes
 - o 2 Servos de rotación continua para las ruedas
 - o 2 Sensores LDR

Conexionado

- Realizaremos el conexionado siguiente:
- Servo rueda izquierda → Pin 10
 - Servo rueda derecha → Pin 12

Libro de Actividades de Robótica Educativa

- LDR Izquierdo → Pin A2
- LDR Derecho → Pin A3

En la Figura 1.2.17-4 izquierda tenemos como quedaría el Printbot con la conexión de los LDR a la tarjeta ZUM BT. De hecho en el chasis del PrintBot ya existe unos huecos para colocar los LDR, pero recuerda que puede utilizar cualquier otro robot basado en Arduino siempre que pongas un sensor en el lado derecho del robot y otro al lado izquierdo, como por el ejemplo el de la Figura 1.2.17-4 derecha.

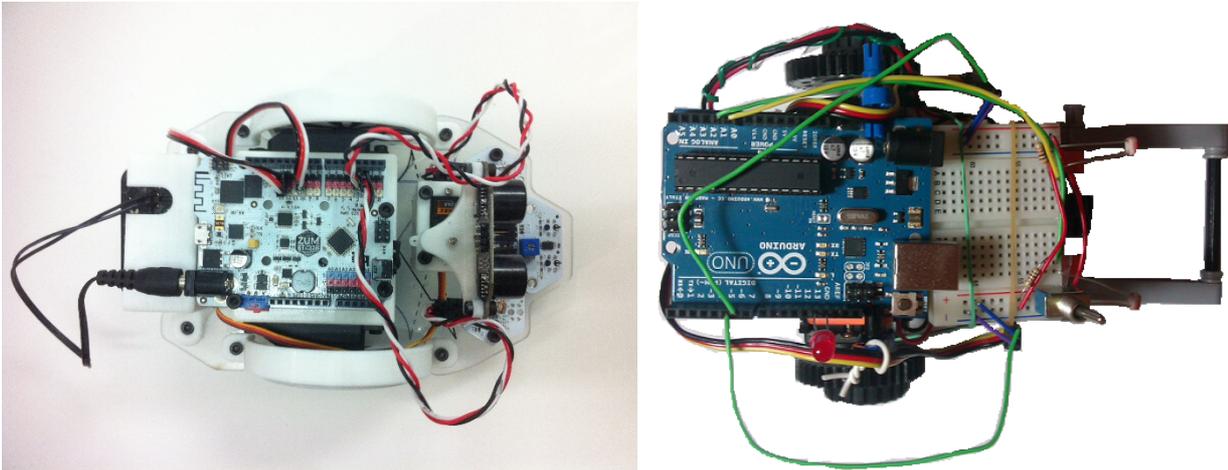


Figura 1.2.17-4 Robots con 2 LDR. (Izquierda) PrintBot. (Derecho) Robot basado en Arduino UNO

Procedemos a abrir un proyecto en Bitbloq y añadir los componentes y el conexionado como se indicaba anteriormente.

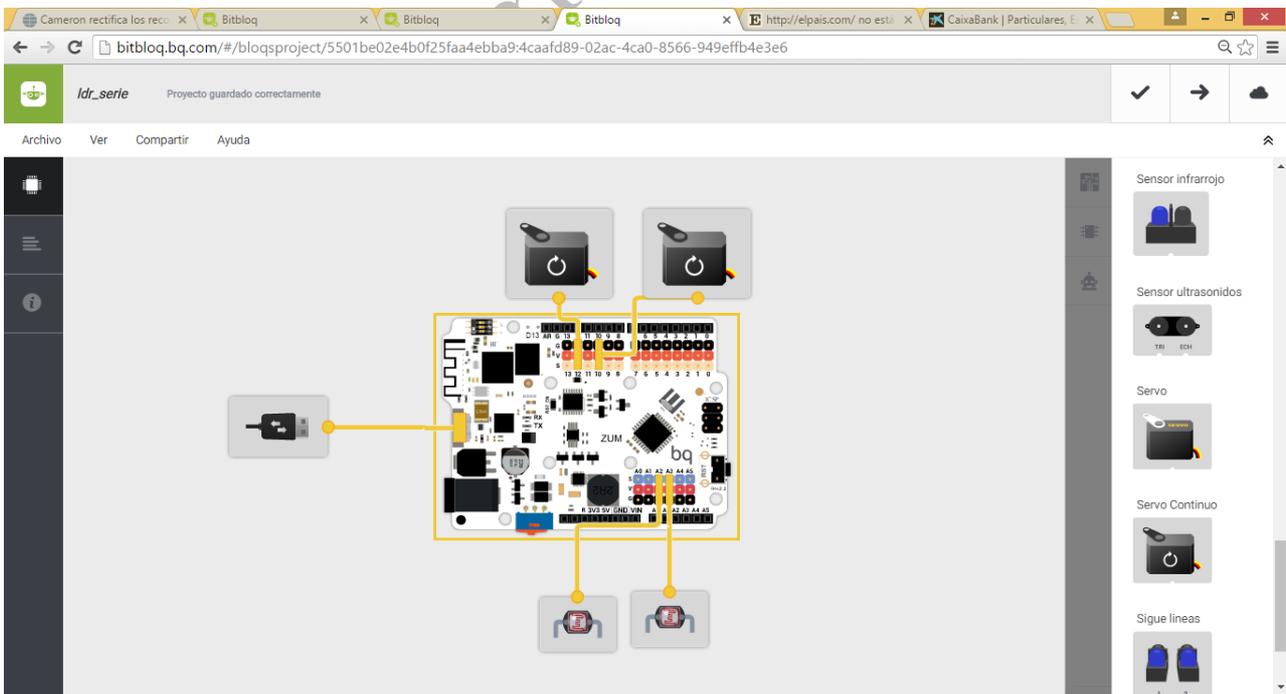


Figura 1.2.17-5 Conexionado en bitbloq de un robot seguidor de luz

Programación

Para que el robot se dirija a una baliza luminosa vamos a programar el siguiente comportamiento:

- SI la intensidad de luz recibida por el LDR derecho es bastante mayor (superior en 30 unidades) que la intensidad de luz recibida por el LDR izquierdo ENTONCES giramos el robot hacia la derecha
- SI la intensidad de luz recibida por el LDR izquierdo es bastante mayor (superior en 30 unidades) que la intensidad de luz recibida por el LDR derecho ENTONCES giramos el robot hacia la izquierda
- SI los dos sensores reciben más o menos la misma luz ENTONCES seguimos recto

Esto se traduce en los siguientes bloques de bitbloq

— Bucle principal (Loop)

The image shows a Bitbloq code editor with the following blocks:

- Declarar variable luz_der = Leer sensor_de_luz_der
- Declarar variable luz_izq = Leer sensor_de_luz_izq
- puerto_serie_0 Enviar Variable luz_der Con salto de línea
- puerto_serie_0 Enviar Variable luz_izq Sin salto de línea
- Si Variable luz_der + 30 < Variable luz_izq ejecutar:
 - Girar servo servo_continuo_der en sentido horario
 - Girar servo servo_continuo_izq en sentido horario
- en cambio, si Variable luz_izq + 30 < Variable luz_der ejecutar:
 - Girar servo servo_continuo_izq en sentido antihorario
 - Girar servo servo_continuo_der en sentido antihorario
- de lo contrario, ejecutar:
 - Girar servo servo_continuo_izq en sentido antihorario
 - Girar servo servo_continuo_der en sentido horario

Siendo su código en Arduino el siguiente:

```
/** Included libraries */  
#include <Servo.h>  
#include <SoftwareSerial.h>  
#include <BitbloqSoftwareSerial.h>
```

Libro de Actividades de Robótica Educativa

```
/** Global variables and function definition */
Servo servo_continuo_izq;
Servo servo_continuo_der;
int sensor_de_luz_der = A3;
int sensor_de_luz_izq = A2;
BqSoftwareSerial puerto_serie_0(0, 1, 9600);

/** Setup */
void setup() {
  servo_continuo_izq.attach(10);
  servo_continuo_der.attach(12);
  pinMode(sensor_de_luz_der, INPUT);
  pinMode(sensor_de_luz_izq, INPUT);
}

/** Loop */
void loop() {
  float luz_der = analogRead(A3);
  float luz_izq = analogRead(A2);
  puerto_serie_0.println(luz_der);
  puerto_serie_0.print(luz_izq);
  if ((luz_der + 30) < luz_izq) {
    servo_continuo_der.write(180);
    servo_continuo_izq.write(180);
  } else if ((luz_izq + 30) < luz_der) {
    servo_continuo_izq.write(0);
    servo_continuo_der.write(0);
  } else {
    servo_continuo_izq.write(0);
    servo_continuo_der.write(180);
  }
}
```

www.auton