

### 1.2.13. Sensor Digital: Infrarrojo

En esta actividad vamos a aprender a utilizar los sensores de infrarrojo. Hay diferentes tipos de sensores de infrarrojo y son utilizados para muchas aplicaciones, como por ejemplo detectar objetos o detectar líneas como veremos en la actividad siguiente.

Como veremos más adelante hay sensores infrarrojos digitales, que devuelven un verdadero o falso si detectan negro o no, y sensores analógicos que devuelven un valor equivalente al nivel de gris detectado (ver Figura 1.2.13-1). En esta práctica utilizaremos sensores digitales como los contenidos en el kit de robótica de BQ.

#### Componentes

- Placa ZUM BT o Arduino UNO compatible
- Sensor infrarrojos digital, por ejemplo el del kit de robótica o cualquiera que tenga un circuito de adaptación (puedes saber que es digital si encuentras una resistencia variable como la indicada en la Figura 1.2.13-1)
- Cable USB para mostrar los datos en el PC

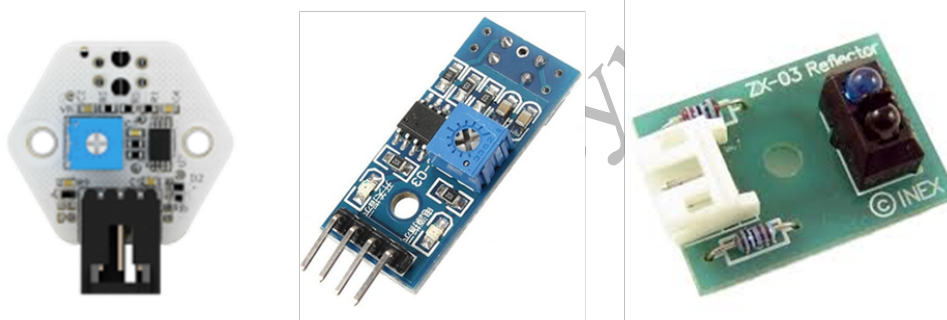


Figura 1.2.13-1 Sensores de infrarrojos digital (izquierda), analógico y digital (centro), analógico (derecha). La resistencia variable (cuadrado en azul) indica que el sensor tiene una salida digital. El sensor del centro tiene un pin para salida digital y otro para salida analógica..

#### Conexionado.

Al tener solo un componente el conexionado es muy sencillo, quedando:

- Sensor infrarrojo → Pin 4
- USB PC → Placa ZUM BT o Arduino compatible

Abriremos un nuevo proyecto en Bitbloq y añadimos del menú de la izquierda la placa ZUM BT (u otra Arduino UNO compatible), el sensor de infrarrojo y el USB como aparece en la Figura 1.2.13-2

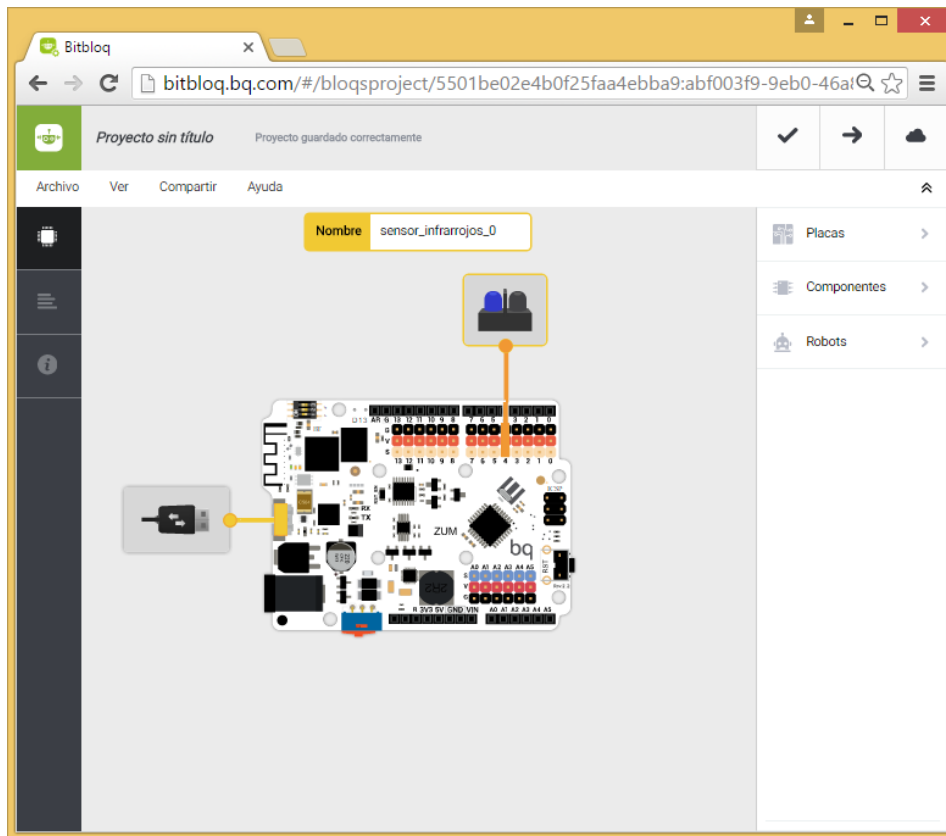


Figura 1.2.13-2 Componentes en Bitbloq para la lectura de infrarrojos

## Programación

El programa por bloques de esta actividad es muy sencillo, simplemente vamos a mandar por el puerto serie lo que leamos del sensor de infrarrojo. Para ello simplemente utilizaremos el bloque “enviar” del puerto serie y lo completaremos con la lectura del sensor de infrarrojos como aparece en la Figura 1.2.13-3. Además, añadiremos un retardo para que la información no se envíe continuamente sino cada 2 segundos como aparece en bloque siguiente.

## Libro de Actividades de Robótica Educativa

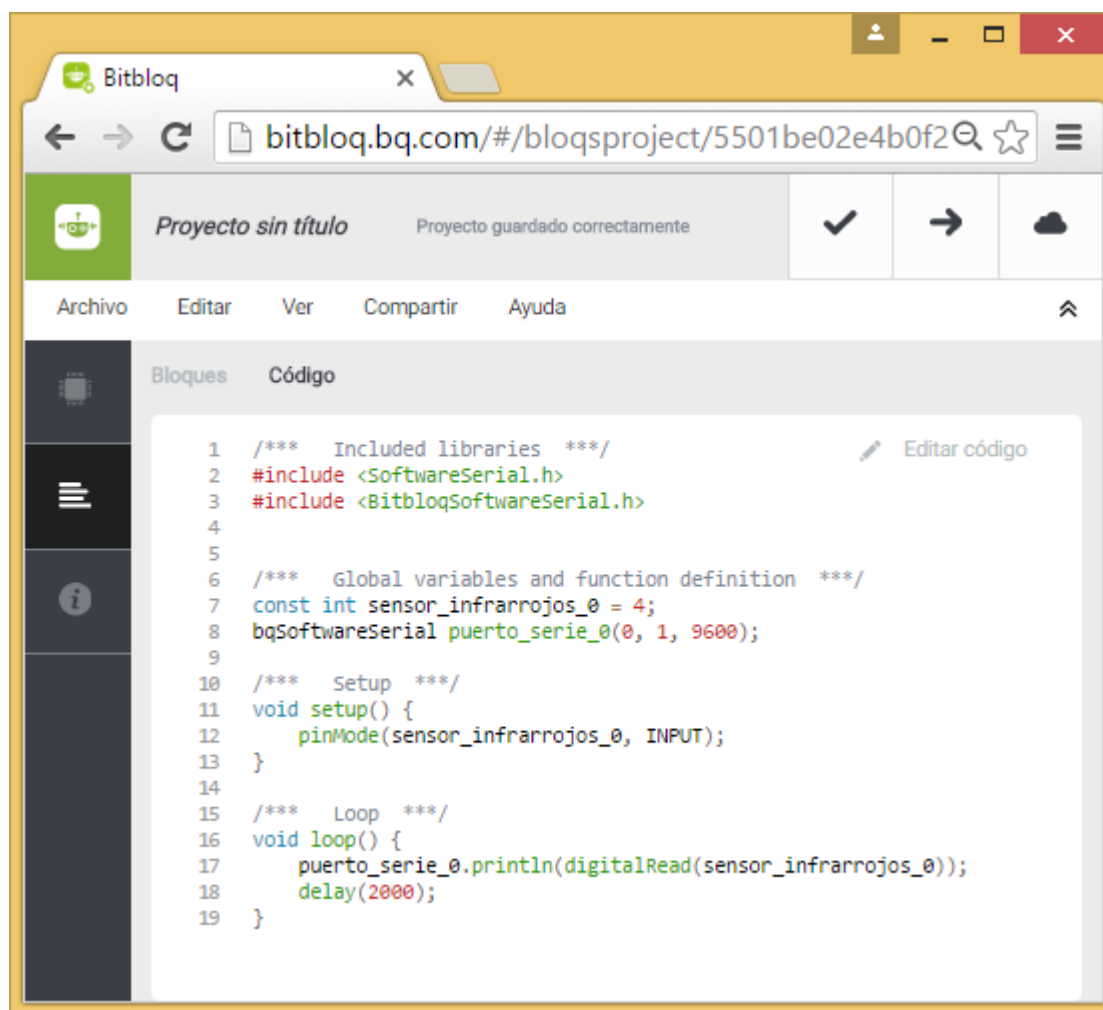
The screenshot shows the Arduino IDE's block programming environment. At the top, there are two tabs: 'Bloques' (selected) and 'Código'. The workspace is divided into three sections:

- Variables globales, funciones y clases:** Contains a dashed box with an information icon and the text 'Arrastra un bloque aquí para empezar tu programa'.
- Instrucciones iniciales (Setup):** Also contains a dashed box with an information icon and the text 'Arrastra un bloque aquí para empezar tu programa'.
- Bucle principal (Loop):** Contains a sequence of blocks:
  - A red block with four segments: 'puerto\_serie\_0' (dropdown), 'Enviar', 'Leer sensor\_infrarrojos\_0' (dropdown), and 'Con salto de línea' (dropdown).
  - A blue block: 'Esperar 2000 ms'.

Figura 1.2.13-3 Programa por bloques para envío de la lectura del sensor de infrarrojo

El código en Arduino correspondiente es el siguiente:

www.automatizapro.com



The screenshot shows the Bitbloq web interface in a browser. The address bar displays the URL `bitbloq.bq.com/#/bloqsproject/5501be02e4b0f2`. The interface includes a top navigation bar with a 'Proyecto sin título' and a 'Proyecto guardado correctamente' message. Below this is a menu with options: 'Archivo', 'Editar', 'Ver', 'Compartir', and 'Ayuda'. The main content area is split into 'Bloques' and 'Código' tabs, with the 'Código' tab selected. The code editor displays the following C++ code:

```
1  /**  Included libraries  **/  
2  #include <SoftwareSerial.h>  
3  #include <BitbloqSoftwareSerial.h>  
4  
5  
6  /**  Global variables and function definition  **/  
7  const int sensor_infrarrojos_0 = 4;  
8  bqSoftwareSerial puerto_serie_0(0, 1, 9600);  
9  
10 /**  Setup  **/  
11 void setup() {  
12     pinMode(sensor_infrarrojos_0, INPUT);  
13 }  
14  
15 /**  Loop  **/  
16 void loop() {  
17     puerto_serie_0.println(digitalRead(sensor_infrarrojos_0));  
18     delay(2000);  
19 }
```

Como vemos, en la línea 12, por medio del comando `pinMode` seleccionamos el pin 4 como entrada. Posteriormente en la línea 17 enviamos por el puerto serie (con `println` que envía añadiendo una nueva línea) lo que leemos del pin4 (con `digitalRead`).

Ya solo queda probarlo. Abriremos un monitor serie (desde el menú `ver`) y comprobaremos que estamos recibiendo unos o ceros en función de si colocamos el sensor sobre negro o sobre blanco.

## Un poco de teoría

Un sensor de infrarrojo es un sensor que puede ver la franja de luz infrarroja. Estos sensores pueden ser de diferentes tipos pero por lo general están compuestos por un emisor (led) de luz infrarroja y un receptor que detecta la reflexión de dicha luz en un objeto.

Hay sensores que son utilizados para detectar líneas negras o blancas (como los de la Figura 1.2.13-1). Básicamente lo que ocurre es que el color blanco refleja la luz infrarroja mientras que el negro absorbe la luz infrarroja (ver Figura 1.2.13-4), esto permite al receptor detectar una diferencia entre negro y blanco.

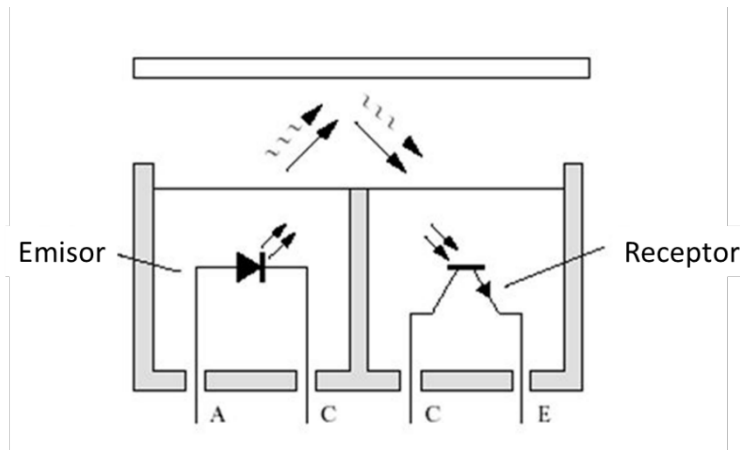


Figura 1.2.13-4 Principio de funcionamiento de un sensor infrarrojo

Hay otros sensores, como el de la figura Figura 1.2.13-5 (izquierda), los que le foco de luz es más potente y se utilizan para detectar obstáculos y medir distancias. El principio de funcionamiento es muy parecido, estos sensores detectan el reflejo de la luz, pero el receptor, como vemos en la Figura 1.2.13-5 (derecha), es lineal (es una banda que detecta el reflejo de la luz en un determinado punto de las misma) lo que permite realizar un proceso de triangulación para saber la distancia a la que se encuentra el objeto.

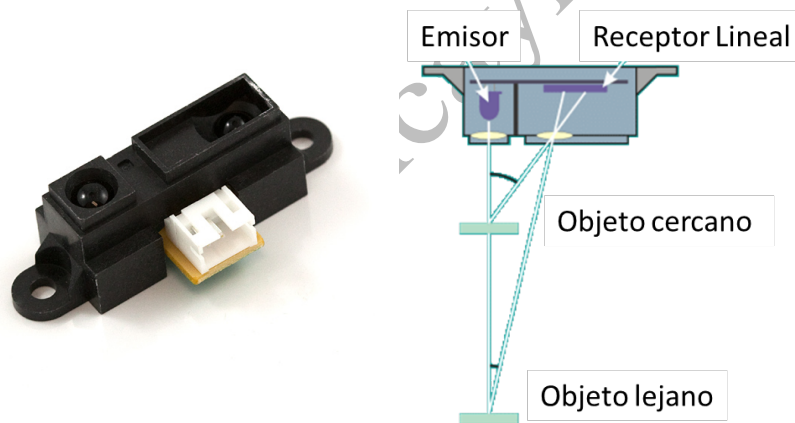


Figura 1.2.13-5 Sensor infrarrojo de distancia