



## **PROYECTO N° 4: Girasol**

Crea tu propio girasol que gire en busca de luz utilizando los componentes del Maker Control Kit y el Maker Kit 1.

La clave de este proyecto es el sensor de luz analógico, que varía su resistencia eléctrica de acuerdo a la luz que detecta. Gracias a él, el girasol detectará la falta de luz y girará para encontrarla. ¡Míralo girar como un girasol real!

**NIVEL DE DIFICULTAD:** Principiante.

**DURACIÓN DEL EJERCICIO:** 60 min.

## MATERIALES:

- 1 Sensor de luz
- 1 Servomotor
- 1 Placa controladora Build&Code 4in1
- 1 Cable USB - Micro USB
- Ordenador
- Plantilla del girasol
- Cartón u otro material para hacer la estructura
- Adhesivo
- Vaso de plástico oscuro

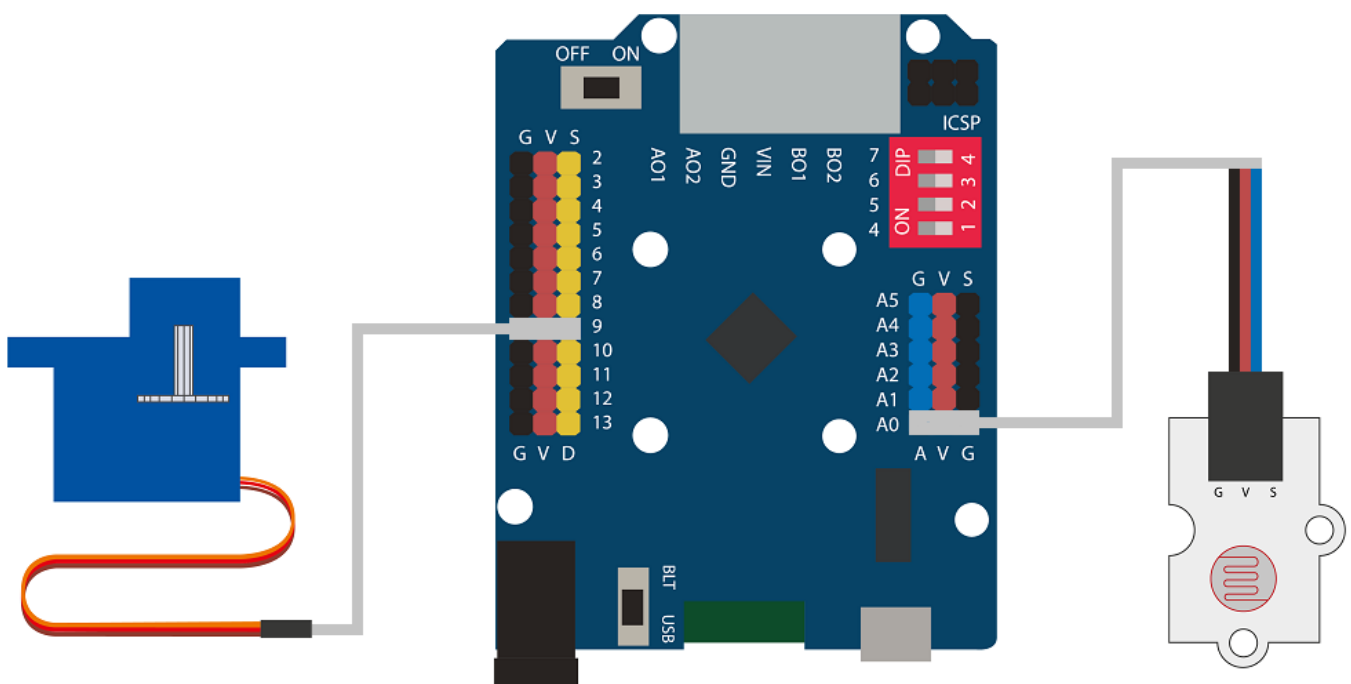
## ¿Qué es el sensor de luz?

Un sensor de luz analógico es un resistor que varía su valor de resistencia eléctrica en función de la cantidad de luz que incide sobre él. También es llamado fotoresistor o fotoresistencia.

## CONEXIONES:

1. Conecta el sensor de luz al puerto analógico A0 de la placa controladora Build&Code 4in1.
2. Conecta el servomotor al puerto digital 9 de la placa controladora Build&Code 4in1.

Para guiarte, mira los colores de los cables y los colores de los terminales de la placa controladora Build&Code 4in1. Cada cable debe ir conectado a su color:



## CONSTRUCCIÓN DE LA ESTRUCTURA:

Los materiales que usarás son principalmente cartón y un vaso de plástico. Si tienes otros materiales que se adaptan a la realización del proyecto, puedes incorporarlos también. Para construir la estructura [descarga la plantilla del girasol](#) y [descarga la guía rápida de montaje](#). Sigue los pasos indicados.

Introduce el sensor de luz en el interior de un vaso de plástico de color oscuro y pégalo en la base. Haz un orificio en el lateral del vaso, en la parte de abajo, para poder pasar el cable del sensor de luz hacia el exterior.

Luego recorta la imagen del girasol, pégala sobre un cartón y recórtala. Utiliza otro trozo de cartón para dibujar y recortar las distintas partes que forman la base del girasol. Monta el girasol, incluyendo las partes de la base.

A continuación, dibuja y recorta la base a la que irá sujeto el servomotor y la base de unión entre el girasol y el servomotor. Monta la base del servomotor y únela al girasol.

## CÓDIGO DE PROGRAMACIÓN:

El programa se basa en las lecturas del sensor de luz. Hemos fijado una relación de valores de 0 a 800. En base a esta relación del sensor de luz, el servomotor girará en un ángulo de 0º a 180º como máximo, haciendo coincidir el valor 0 del sensor de luz con el 0º del servomotor. De igual manera, los valores superiores a 800 del sensor de luz coincidirán con los 180º del servomotor.

Puedes realizar esta actividad utilizando los *software* Arduino y Bitbloq, además de otros *software* de programación por bloques compatibles. A continuación encontrarás el código de programación necesario para cada *software*.

### Código Arduino

1. [Descarga el software Arduino](#) y realiza el proceso de instalación.
2. Abre el programa Arduino y, una vez en él, copia el siguiente programa:

```
#include <Servo.h> // LIBRERÍA DEL SERVO MOTOR
Servo Motor; // VARIABLE DEL SERVOMOTOR
int PortLight = A0; // Puerto conectado el sensor de luz
int Degree, LightB; // Degree variable grados motor // LightB
variable valor sensor de luz

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Motor.attach (9); //Puerto conectado el servomotor
  Degree = 15;
```

```
Motor.write (Degree); // Servomotor a 15º
}

void loop() {
  // Put your main code here, to run repeatedly:
  LightB = analogRead(PortLight); // Lectura valores sensor de luz
  Serial.println(LightB); // Mostrar por pantalla los valores del
  sensor de luz
  if(LightB < 100) // Si Light es más pequeño que 100
  {
    Degree = 15; // Grados = 15
  }
  if((LightB > 100)&(LightB < 200)) // Si Light es más pequeño que
  200 y más grande que 100
  {
    Degree = 36; // Grados = 36
  }
  if((LightB > 200)&(LightB < 300)) // Si Light es más pequeño que
  300 y más grande que 200
  {
    Degree = 57; // Grados = 57
  }
  if((LightB > 300)&(LightB < 400)) // Si Light es más pequeño que
  400 y más grande que 300
  {
    Degree = 78; // Grados = 78
  }
  if((LightB > 400)&(LightB < 500)) // Si Light es más pequeño que
  500 y más grande que 400
  {
    Degree = 99; // Grados = 99
  }
  if((LightB > 500)&(LightB < 600)) // Si Light es más pequeño que
  600 y más grande que 500
  {
    Degree = 120; // Grados = 120
  }
  if((LightB > 600)&(LightB < 700)) // Si Light es más pequeño que
  700 y más grande que 600
  {
    Degree = 140; // Grados = 140
  }
  if((LightB > 700)&(LightB < 800)) // Si Light es más pequeño que
  800 y más grande que 700
  {
    Degree = 160; // Grados = 160
  }
}
```

```
}  
if(LightB > 800) // Si Light es más grande que 800  
{  
  Degree = 180; // Grados = 180  
}  
Motor.write (Degree); // Fijar el servomotor a los grados  
correspondientes.  
}
```

3. Configura y carga el código, siguiendo las instrucciones indicadas en el [documento de Primeros Pasos de la placa Build&Code 4in1](#).

### **Código para el *software* de programación por bloques compatible**

1. [Descarga el software](#) y realiza el proceso de instalación.
2. Abre el programa y, una vez en él copia el siguiente código.

```

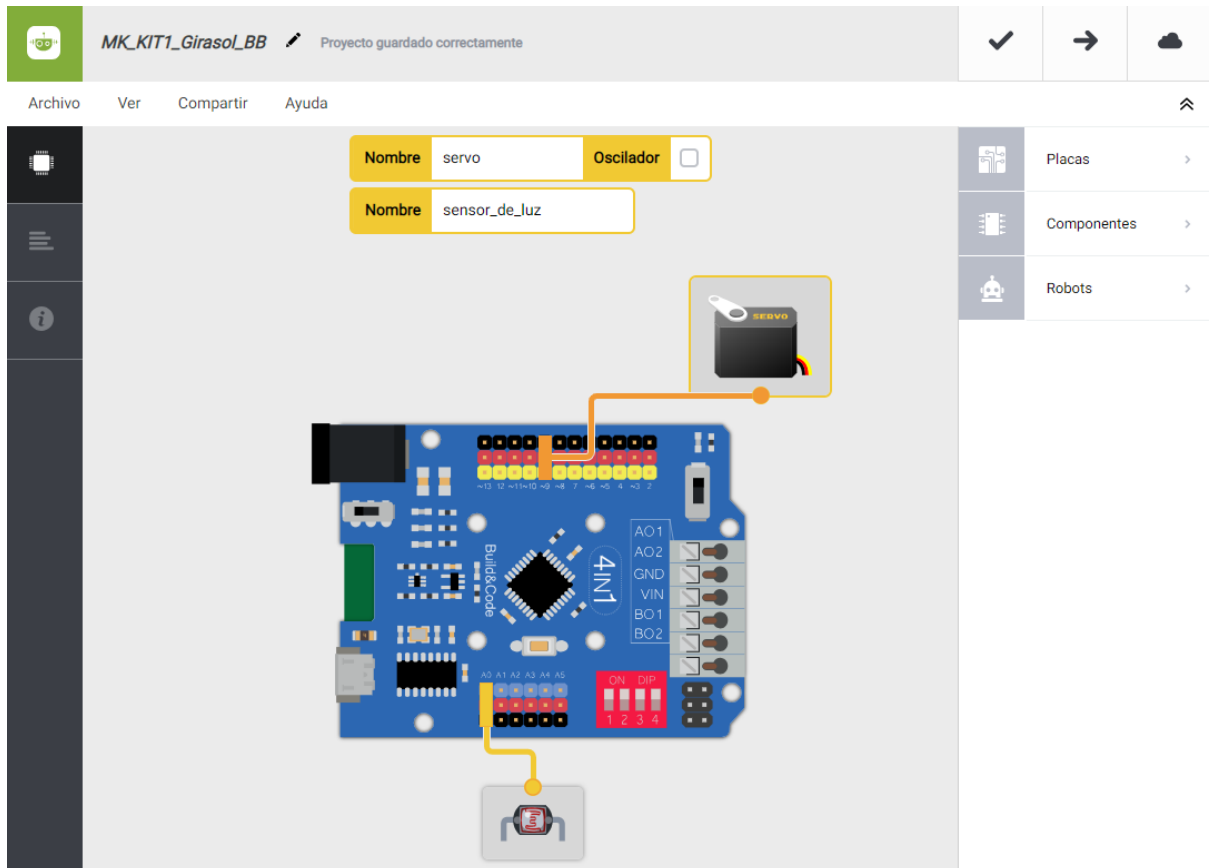
Programa de Arduino
por siempre
  fijar LightB a leer pin analógico (A) 0
  si LightB < 100 entonces
    fijar ángulo del pin 9 del servo a 15
  si LightB < 200 y LightB > 100 entonces
    fijar ángulo del pin 9 del servo a 36
  si LightB < 300 y LightB > 200 entonces
    fijar ángulo del pin 9 del servo a 57
  si LightB < 400 y LightB > 300 entonces
    fijar ángulo del pin 9 del servo a 78
  si LightB < 500 y LightB > 400 entonces
    fijar ángulo del pin 9 del servo a 99
  si LightB < 600 y LightB > 500 entonces
    fijar ángulo del pin 9 del servo a 120
  si LightB < 700 y LightB > 600 entonces
    fijar ángulo del pin 9 del servo a 140
  si LightB < 800 y LightB > 700 entonces
    fijar ángulo del pin 9 del servo a 160
  si LightB > 800 entonces
    fijar ángulo del pin 9 del servo a 180
  
```

3. Configura y carga el código, siguiendo las instrucciones indicadas en el [documento de Primeros Pasos de la placa Build&Code 4in1](#).

## Código BitBloq

1. Accede al software [Bitbloq](#).
2. Abre el programa Bitbloq y, una vez en él, copia el siguiente código:

- **Hardware**



- **Software**

MK\_KIT1\_Girasol\_BB

Archivo Editar Ver Compartir Ayuda

Bloques Código

— Variables globales y funciones

Declarar variable LightB con tipo entero = 0

— Instrucciones iniciales (Setup)

Mover servo a 15 grados

— Bucle principal (Loop)

Variable LightB = Leer sensor\_de\_luz

Si Variable LightB < 100 ejecutar:

Mover servo a 15 grados

Si Variable LightB > 100 y Variable LightB < 200 ejecutar:

Mover servo a 36 grados

Si Variable LightB > 200 y Variable LightB < 300 ejecutar:

Mover servo a 57 grados

Si Variable LightB > 300 y Variable LightB < 400 ejecutar:

Mover servo a 78 grados

Si Variable LightB > 400 y Variable LightB < 500 ejecutar:

Mover servo a 99 grados

Si Variable LightB > 500 y Variable LightB < 600 ejecutar:

Mover servo a 120 grados

Si Variable LightB > 600 y Variable LightB < 700 ejecutar:

Mover servo a 140 grados

Si Variable LightB > 700 y Variable LightB < 800 ejecutar:

Mover servo a 160 grados

Si Variable LightB > 800 ejecutar:

Mover servo a 160 grados

Componentes >  
Fun Funciones >  
Var Variables >  
Cód Código >  
Mat Matemáticas >  
Tex Texto >  
Con Control >  
Lóg Lógica >



3. Configura y carga el código, siguiendo las instrucciones indicadas en el [documento de Primeros Pasos de la placa Build&Code 4in1](#).

## **RESULTADO DEL EJERCICIO**

Como resultado del proyecto el sistema medirá la cantidad de luz que incide en el sensor de luz, y moverá el servomotor una cierta cantidad de grados en función de esos valores, imitando el movimiento de un girasol real.