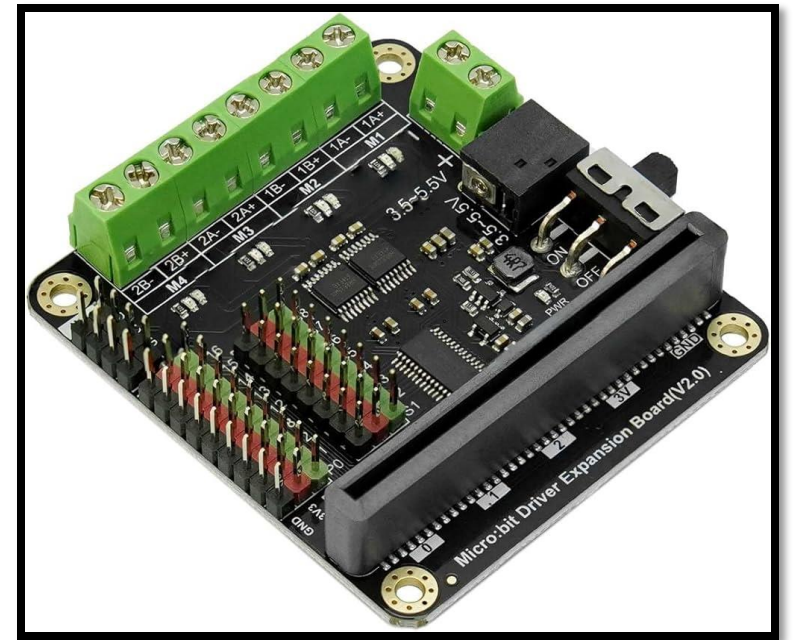
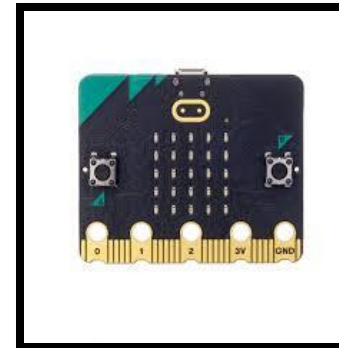


Programa Código Escuela 4.0



Kit de robótica para
Micro:Bit.



[Kit de robótica para Micro:Bit](#) © 2026 by [Jaime Temes Mendoza](#) is
licensed under [CC BY-NC-SA 4.0](#)



Índice

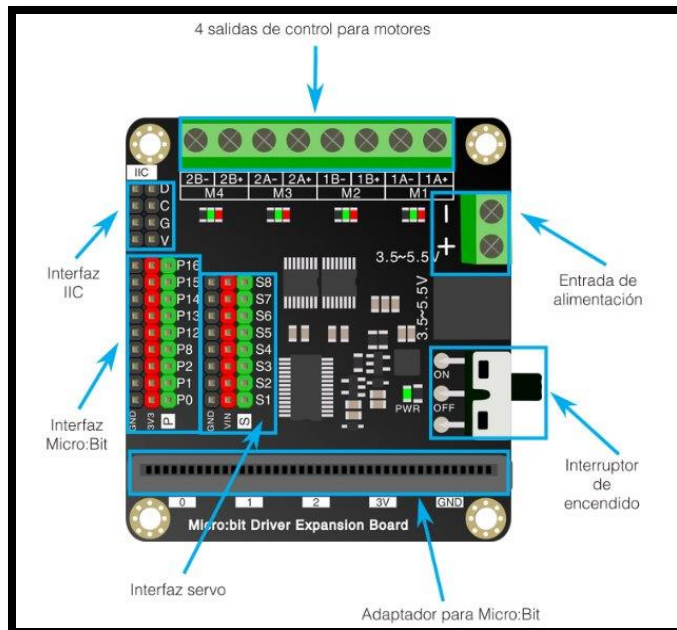
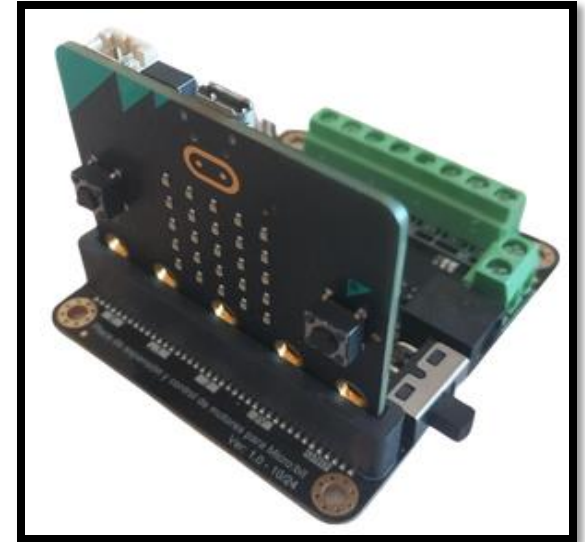


- Introducción.
- Materiales.
- LDR-comparador de umbral.
- Lectura del comparador de umbral.
- Detector de presencia con el comparador de umbral.
- Servomotor.
- Programa servomotor.
- Servomotor con comparador de umbral.
- Servomotor con comparador de umbral. Programación.
- Servomotor con comparador de umbral. Solución.
- Sensor de infrarrojos (IR).
- Servomotor con dos sensores IR.
- Servomotor con dos sensores IR. Programación
- Servomotor con dos sensores IR. Solución.
- Parking con dos sensores IR y dos servomotores.
- Parking con dos sensores IR y dos servomotores. Solución.
- Parking con dos sensores IR y dos servomotores. Avanzado.
- Parking con dos sensores IR y dos servomotores. Avanzado. Solución.
- Tira de LEDs.
- Tira de LEDs. Programación.
- Tira de LEDs. Solución.
- Parking con tira de LEDs. Avanzado II.
- Parking con tira de LEDs. Avanzado II. Solución.
- Material para impresión 3D.

Introducción

En este documento vamos a trabajar **montajes sencillos** para que el alumnado empiece a trabajar con algunos de los sensores que vienen el kit de robótica para Micro:Bit.

Para las conexiones utilizaremos la placa de expansión. A la derecha puedes verla correctamente conectada a la micro:bit.



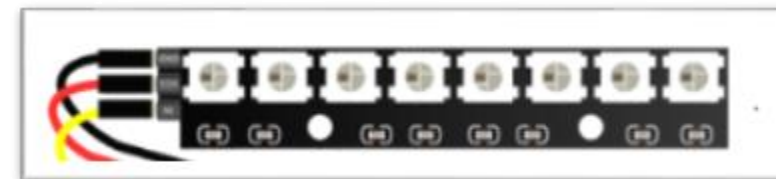
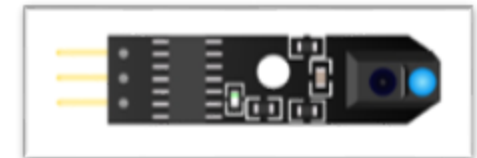
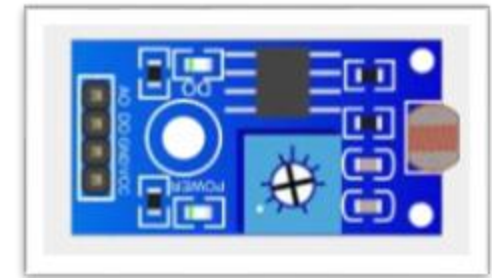
Utilizaremos los pines para sensores (columna P) y servomotores (columna S). En ambas el código de color es el mismo:

- **Conector negro: GND (ground, tierra).**
- **Conector rojo: Vcc (positivo). 3 V en columna P y 5 V en columna S.**
- **Conector verde: datos (D0). Pueden ser datos que van del elemento a la placa (sensores) o que salen de la placa hacia el componente (servomotores)**

Materiales

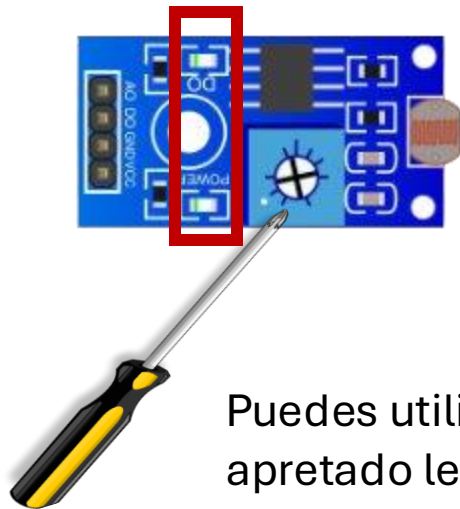
A continuación, se detallan los elementos que necesitaremos para desarrollar las prácticas.

- Micro:Bit con cable de conexión al ordenador.
- Placa de expansión para Micro:Bit.
- Portapilas con 3 pilas AA.
- Comparador de umbral (LDR).
- 2 sensores IR.
- 2 Servomotores de 180°.
- Tira de LEDs.
- 9 cables dupont hembra-hembra.



LDR-Comparador de umbral

Un **LDR** (Light Dependent Resistor) es un sensor cuya resistencia varía según la cantidad de luz que recibe. A más luz, menor resistencia; a menos luz, mayor resistencia. Normalmente se utilizan con los pines analógicos de Arduino, obteniendo un valor comprendido entre 0 y 1023.



El sensor que tenemos en nuestro kit funciona es un comparador de umbral. Esto quiere decir que enviará una señal digital (0 o 1) según detecte suficiente luz o no, respectivamente.

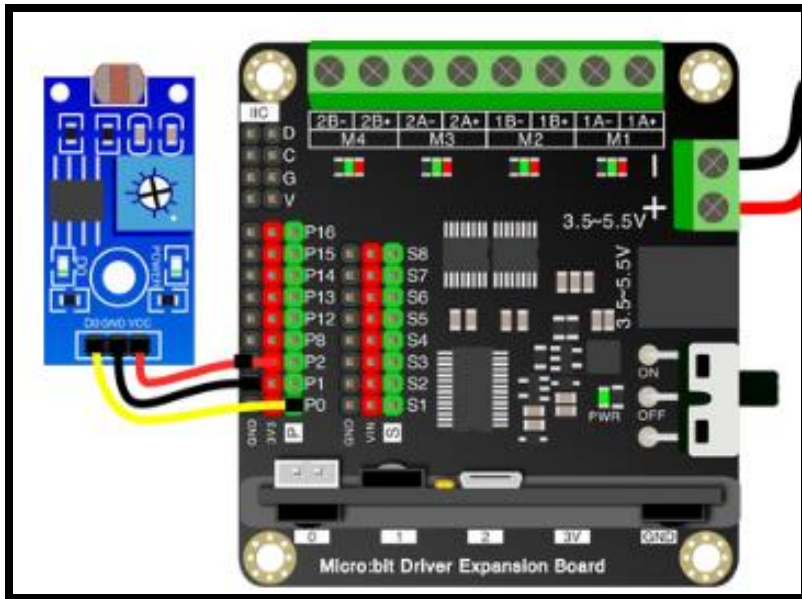
El sensor que incorpora dos LEDs, uno que se enciende cuando está conectado a la corriente y otro que se enciende cuando detecta suficiente luz.

Puedes utilizar un destornillador para calibrar el umbral del nivel de luz. Si está completamente apretado lee siempre luz, y al revés lee siempre oscuridad. Ajústalo para que lea luz justo con la iluminación ambiente de tu aula. Al pasar la mano cerca verás que se apaga el LED.

**Detecta luz – Se enciende el LED – envía 0 lógico.
No detecta luz – Se apaga el LED – envía 1 lógico.**

Lectura del comparador de umbral.

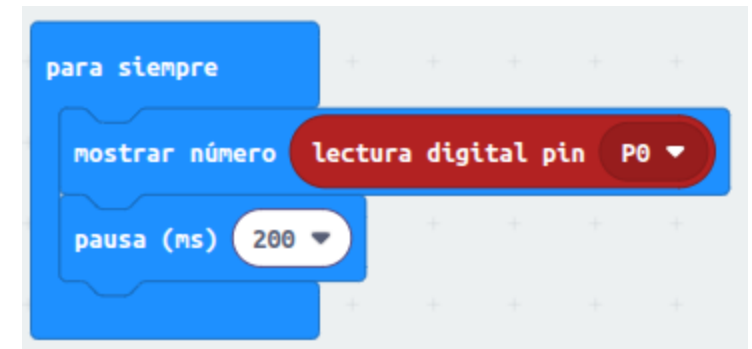
En primer lugar, debemos asegurarnos de conectar debidamente el sensor a la placa. Necesitarás **tres cables dupont hembra-hembra**: debes conectar el pin marcado como **VCC (positivo, + 3 V)** a cualquiera de los pines rojos de la columna de sensores, el **GND** a cualquiera negro (tierra o neutro) el **D0 (datos)** al pin **P0**.



En nuestra primera práctica simplemente observaremos que el sensor está funcionando correctamente. Para conseguirlo desplegamos el bloque de **Avanzado** y dentro de **pines** buscamos **lectura digital pin**:



Luego hacemos un programa sencillo para comprobar que está leyendo el sensor de manera correcta.



Detector de presencia con el comparador de umbral.

Ahora vamos a programar la micro:bit para que muestre el icono de persona cuando detecte presencia y otro icono cualquiera cuando no lo haga. Recuerda que **detecta presencia cuando no hay luz y el pin 0 lee un 1**.

Cuando hayas terminado puedes cambiar el programa para incluir distintos sonidos, animaciones,... cuando detecte presencia.

Bloques que vamos a usar:



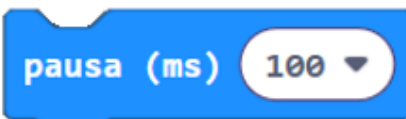
Básico



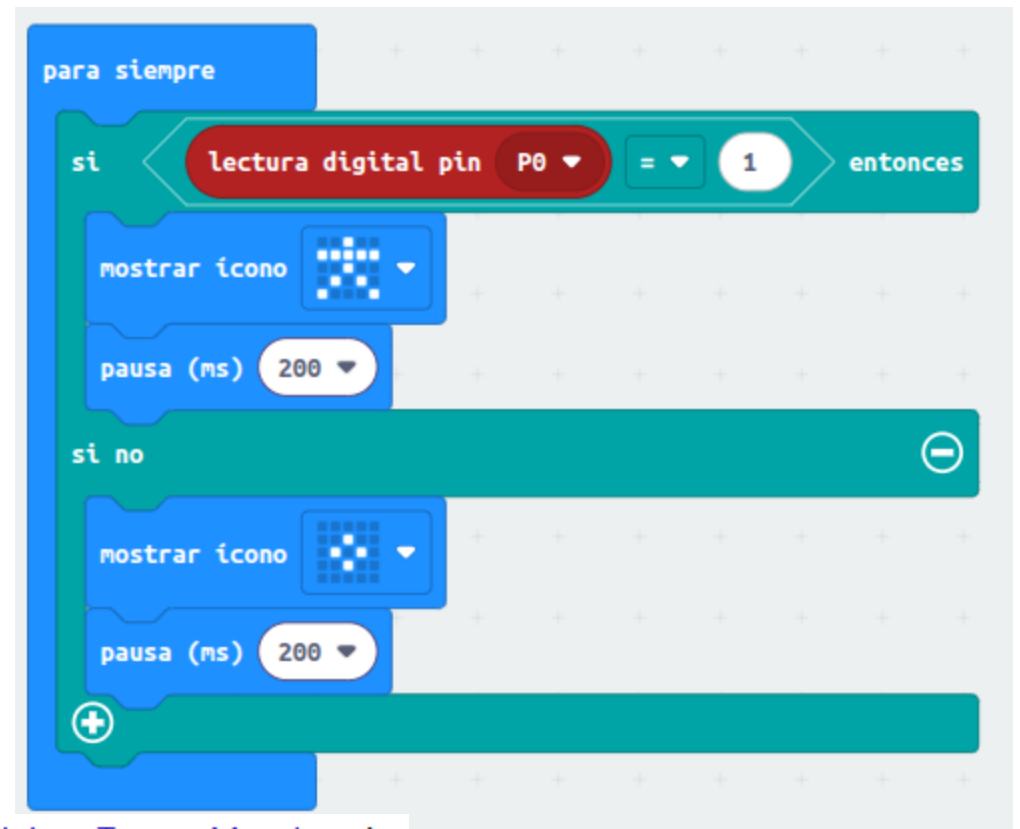
Pines



Lógica

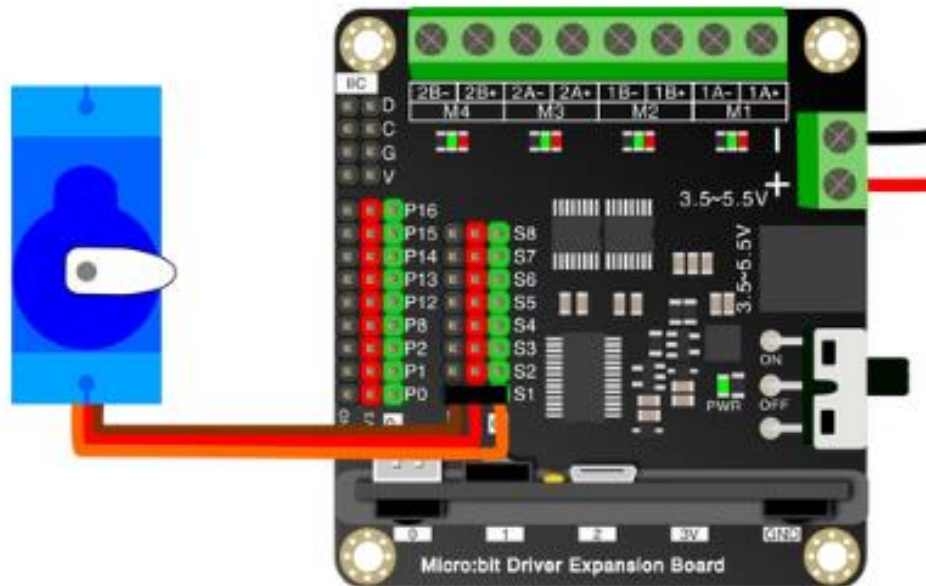


lectura digital pin P0



Servomotor

Los servomotores de 180° se sitúan en un ángulo concreto según la información que le proporcionamos a través de nuestra programación . En este caso sencillo haremos que el servomotor vaya girando de 90° en 90°.



La placa de expansión tiene una columna dedicada para servomotores. Están marcados con una S (van de S1 a S8). El código de **colores de los cables es: marrón oscuro va al GND (tierra), rojo a la alimentación (+5 V) y el naranja/amarillo transmite los datos al conector verde.**

Es muy importante utilizar la extensión propia de la placa y conectar los servos en la columna indicada. En [este enlace](#) tienes un programa en blanco con la extensión ya cargada.

Programa servomotor

Ahora vamos a programar la micro:bit para que el servomotor de 180° vaya pasando por las posiciones 0°, 90°, 180°, 90°.

Cuando hayas terminado puedes cambiar los ángulos a los que se sitúa el servomotor o utilizar una variable y un bucle para que se vaya moviendo. Puedes consultar [este ejemplo](#) si te hace falta.

Bloques que vamos a usar:

Básico

DF-Driver

pausa (ms) 100 ▼

Servo S1 ▼ degree 0



Los servos de 180° son los que tienen la etiqueta de color azul o un gomet azul en un lado.

para siempre

Servo S1 ▼ degree 0

pausa (ms) 500 ▼

Servo S1 ▼ degree 90

pausa (ms) 500 ▼

Servo S1 ▼ degree 180

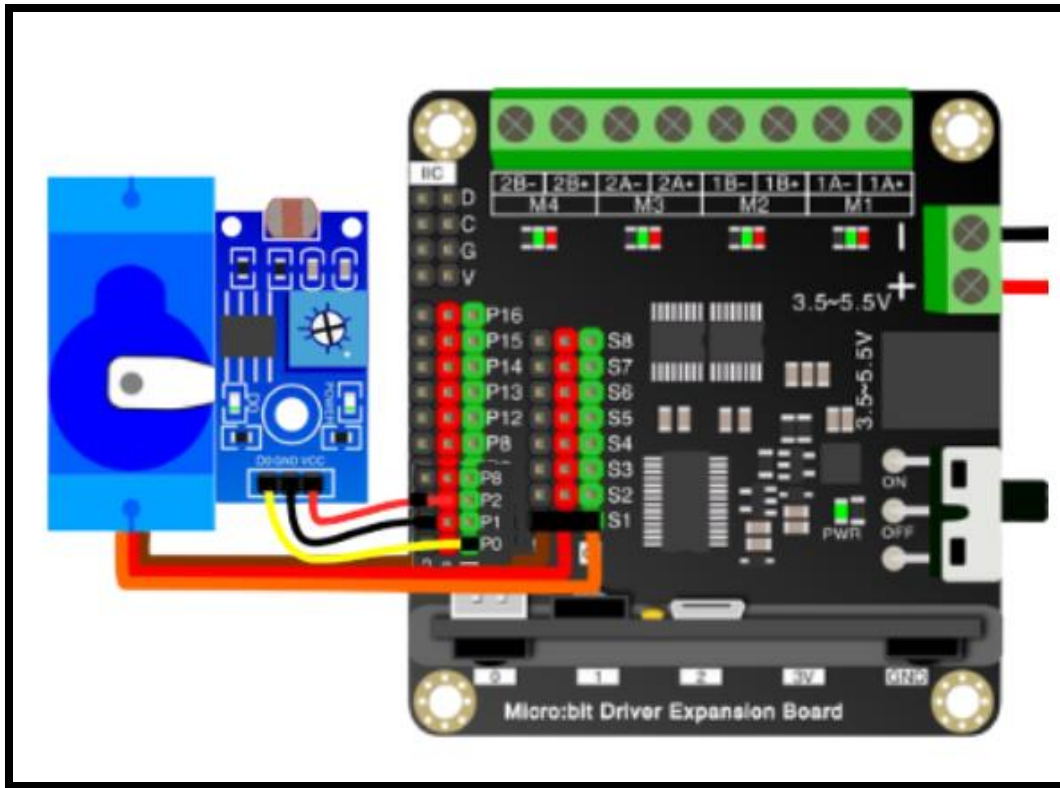
pausa (ms) 500 ▼

Servo S1 ▼ degree 90

pausa (ms) 500 ▼

Servomotor con sensor umbral (barrera)

Vamos a juntar los proyectos anteriores para hacer que cuando el sensor de umbral (LDR) detecte presencia se levante el servomotor (pase de 0° a 90°) mientras el zumbador emite pitidos.



Las **conexiones** son:

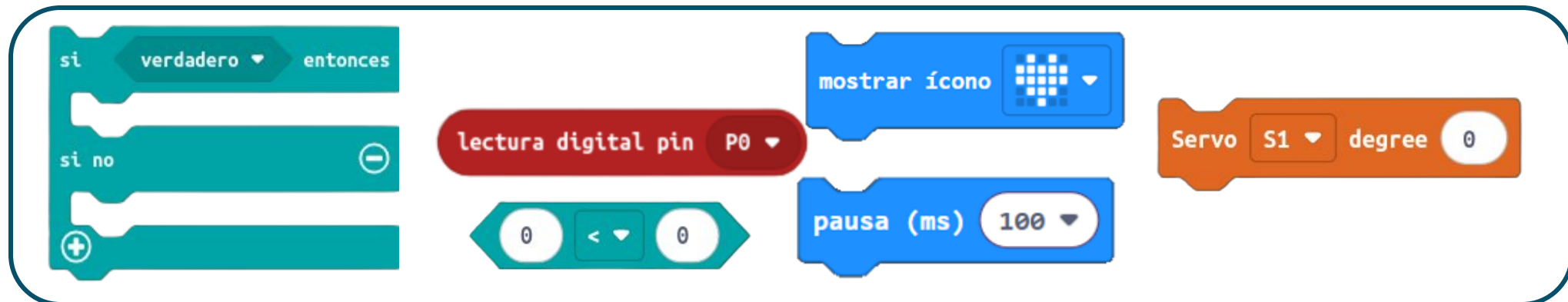
- **Servomotor:** pin S1.
- **Sensor umbral (LDR):** pin S0.

Servomotor con comparador de umbral (barrera). Programación.

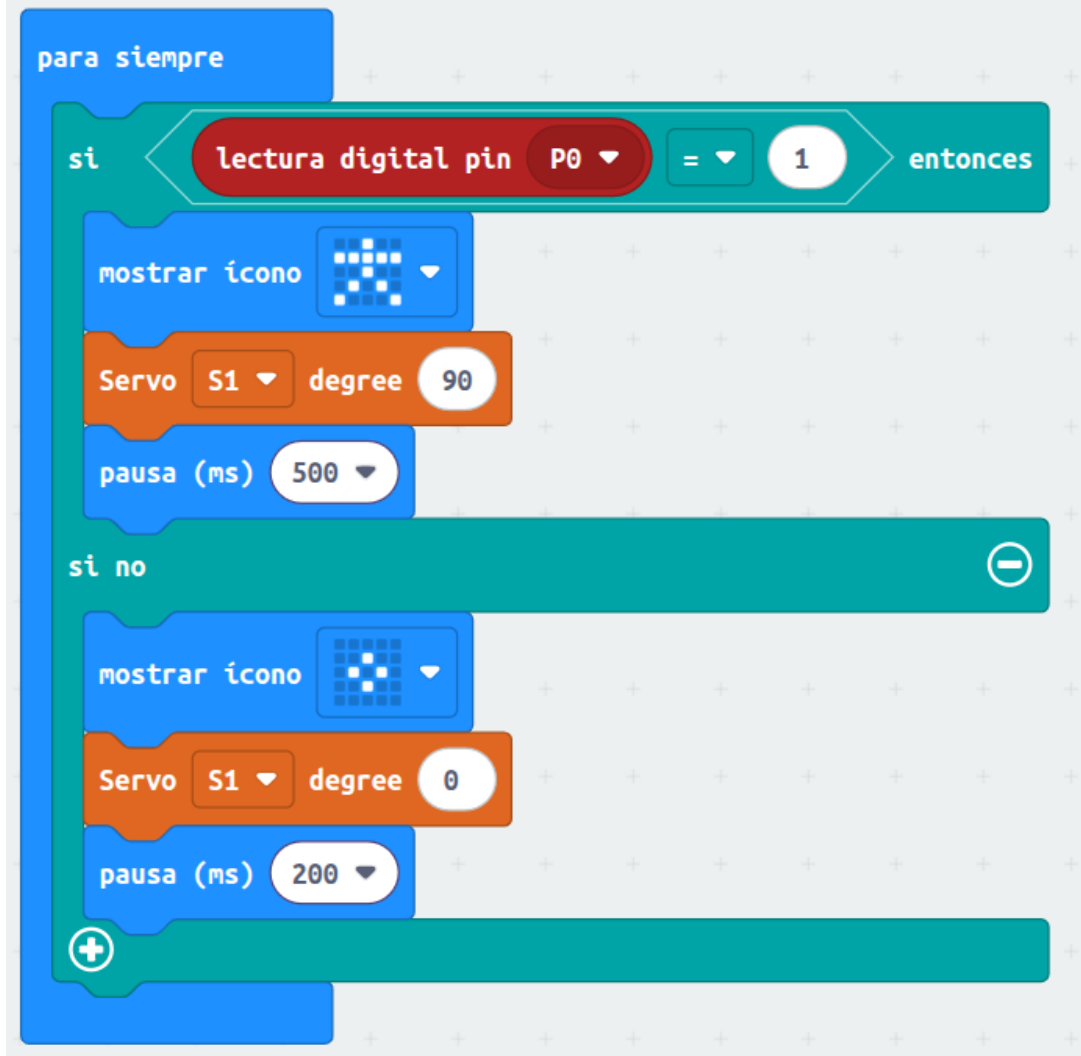
Ahora vamos a programar la micro:bit para que muestre el icono de persona y levante la barrera cuando detecte presencia y otro icono cualquiera y que baje la barrera cuando no lo haga. Recuerda que **detecta presencia cuando no hay luz y el pin 0 lee un 1**.

Cuando hayas terminado puedes cambiar el programa para incluir distintos sonidos, animaciones,... cuando detecte presencia de manera que avise de la subida/baja de la barrera.

Bloques que vamos a usar:



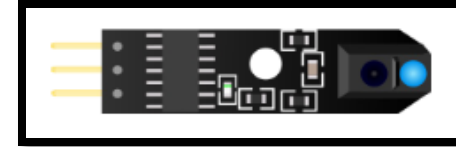
Servomotor con comparador de umbral (barrera). Solución.



[Enlace al programa](#)

Sensor de infrarrojos (IR)

El sensor de infrarrojos consta de un emisor (azul) y un receptor (negro)

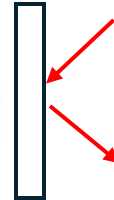
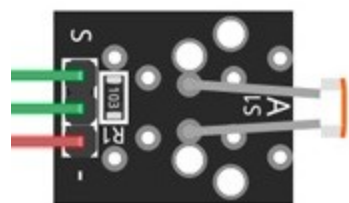


Cuando hay una superficie clara cercana (5 cm o menos) la luz se refleja con suficiente intensidad como para ser detectada.

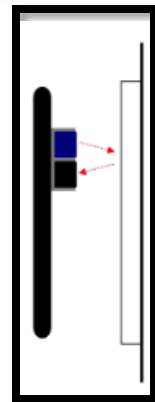
El sensor dará una **salida digital de 1 cuando detecte luz** y de 0 cuando no lo haga.

En la parte trasera dispone de un pequeño LED que se enciende cuando la salida de digital es 1, es decir, cuando detecte luz.

En cualquier de las prácticas anteriores podemos sustituir el sensor de umbral (LDR) por el sensor de IR y funcionará con la misma programación. Las conexiones son diferentes, puedes consultar la página siguiente para ver cómo hacerlo correctamente.



Cuando un objeto se sitúa frente al LDR impide el paso de la luz, el sensor detecta oscuridad y la salida digital es uno.



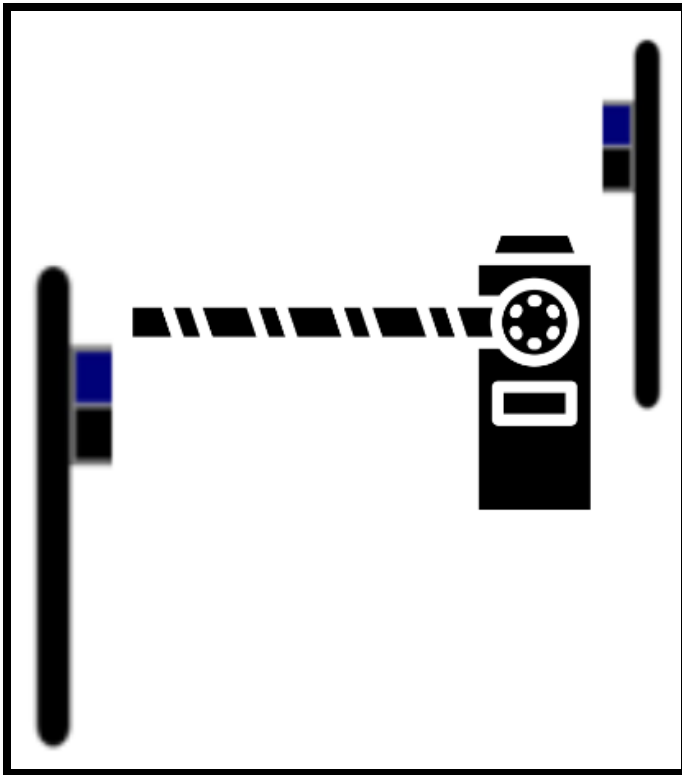
Cuando un objeto se sitúa frente al sensor IR la luz se refleja y es detectada por el receptor. La salida digital es uno.

Los dos sensores dan uno al detectar presencia, pero tienen lógicas diferentes:

- El sensor de umbral (LDR) da uno cuando no detecta luz.
- El sensor de IR da uno cuando detecte luz

Servomotor con dos sensores IR (barrera).

En esta práctica utilizaremos haremos que uno de los sensores IR sea el de entrada (levanta la barrera cuando detecta presencia) y el otro la baje (después de haber atravesado la barrera).



Cuando los LEDs traseros de los sensores están iluminados están detectando presencia.

GND
VCC (+ V)
Datos.



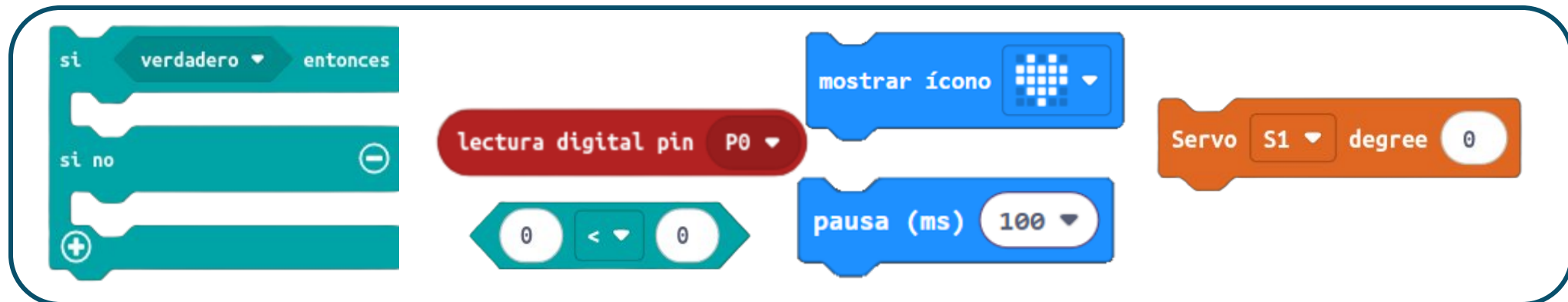
Fíjate bien en las conexiones del sensor IR, no son iguales que las del sensor de LDR.

Servomotor con dos sensores IR (barrera). Programación.

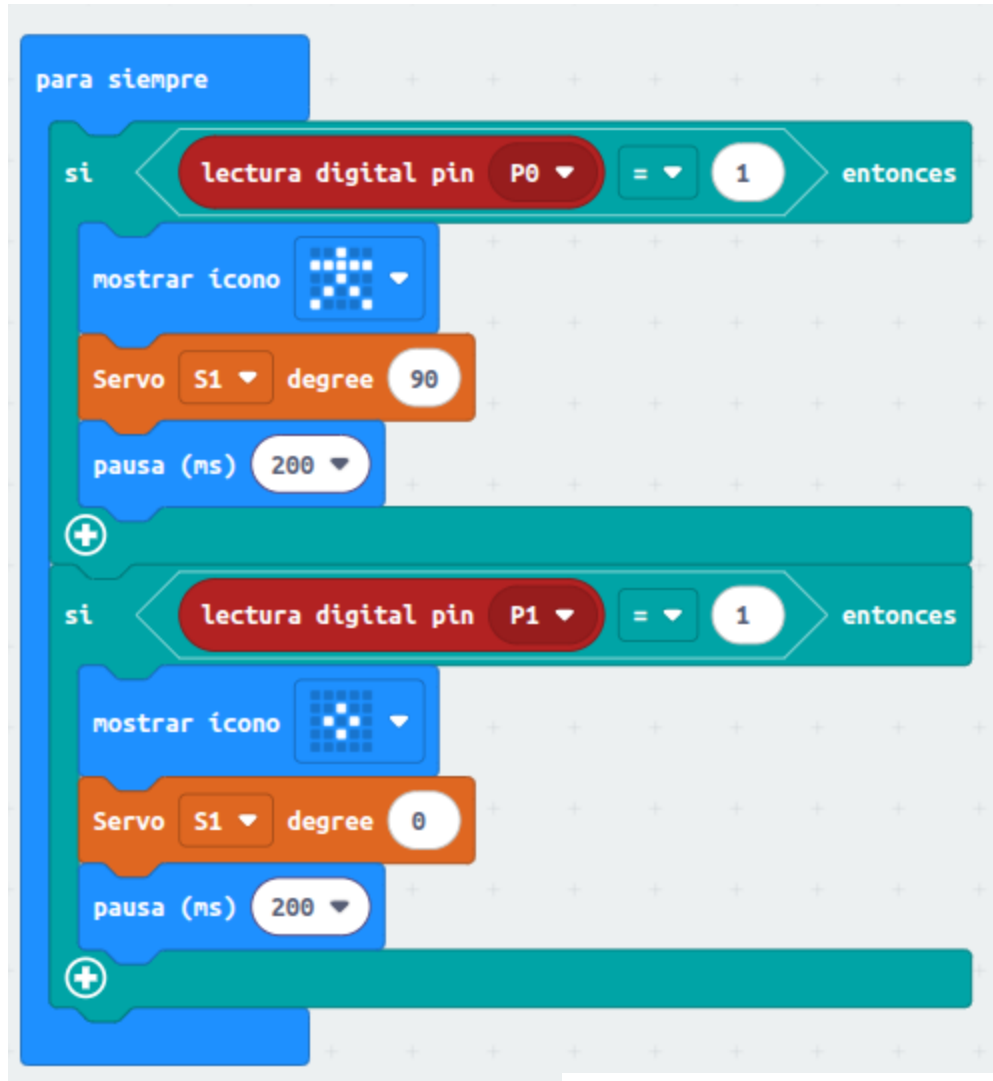
Ahora vamos a programar la micro:bit para que muestre el icono de persona y levante la barrera cuando detecte presencia en el sensor 1. Cuando sea el sensor 2 el que detecte presencia baja la barrera y muestra otro icono.

Cuando hayas terminado puedes cambiar el programa para incluir distintos sonidos, animaciones,... de manera que avise de la subida/baja de la barrera.

Bloques que vamos a usar:



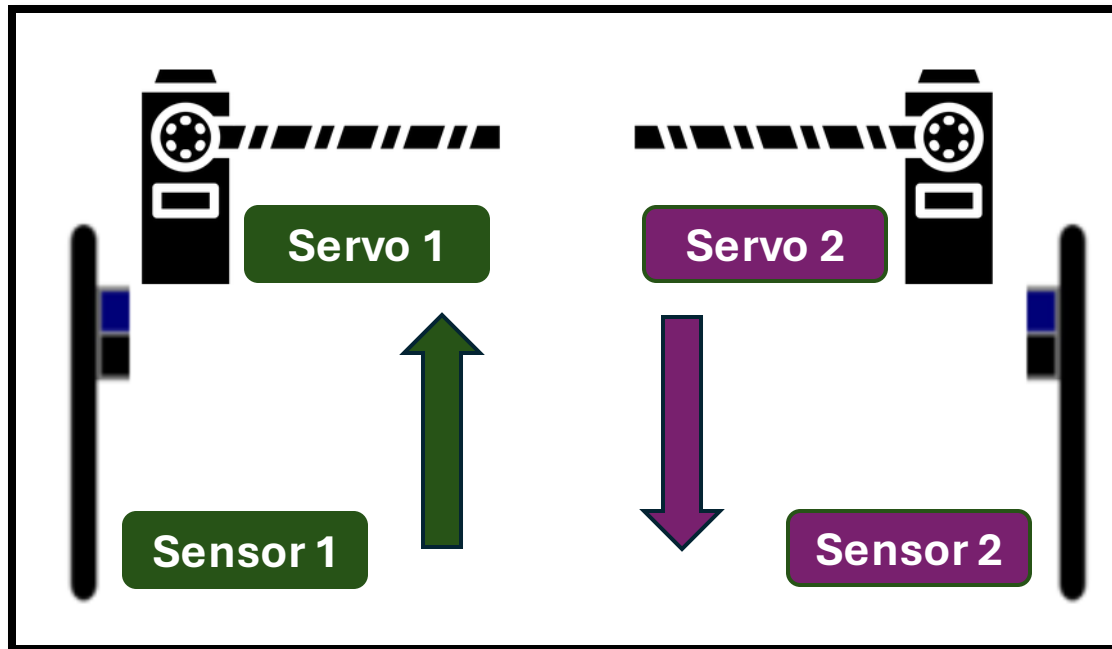
Servomotor con dos sensores IR (barrera). Solución.



[Enlace al programa](#)

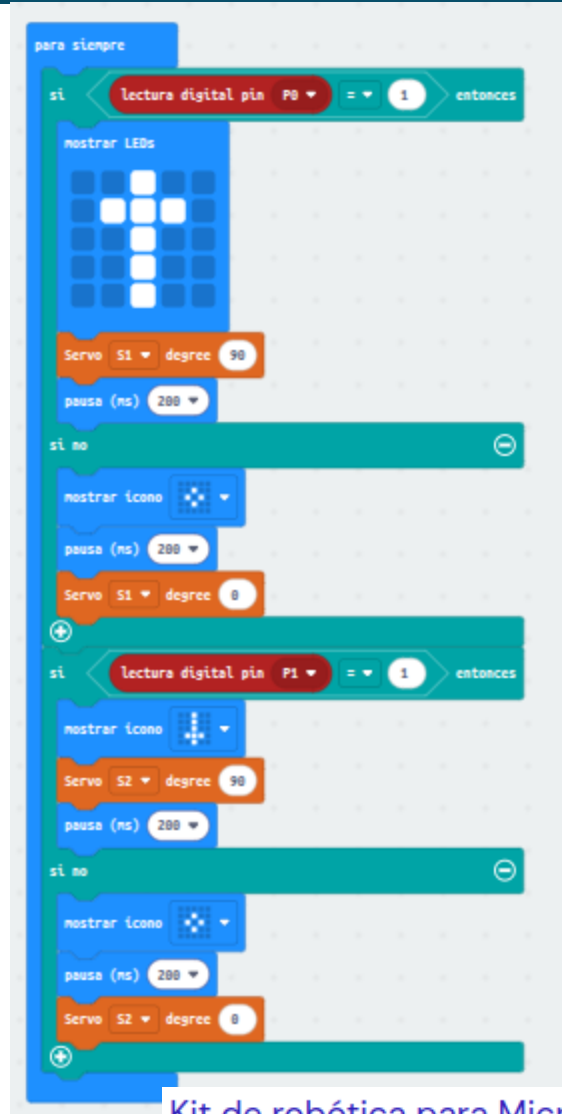
Parking con dos sensores IR y dos servomotores.

En esta práctica utilizaremos el mismo montaje que el anterior añadiendo otro servo. Con el sensor IR 1 y el Servo 1 controlaremos la entrada al parking, mientras que son el sensor IR 2 y el servo 2 controlaremos la salida.



La programación utilizará los mismos bloques que las prácticas anteriores. Es importante tener cuidado con las conexiones.

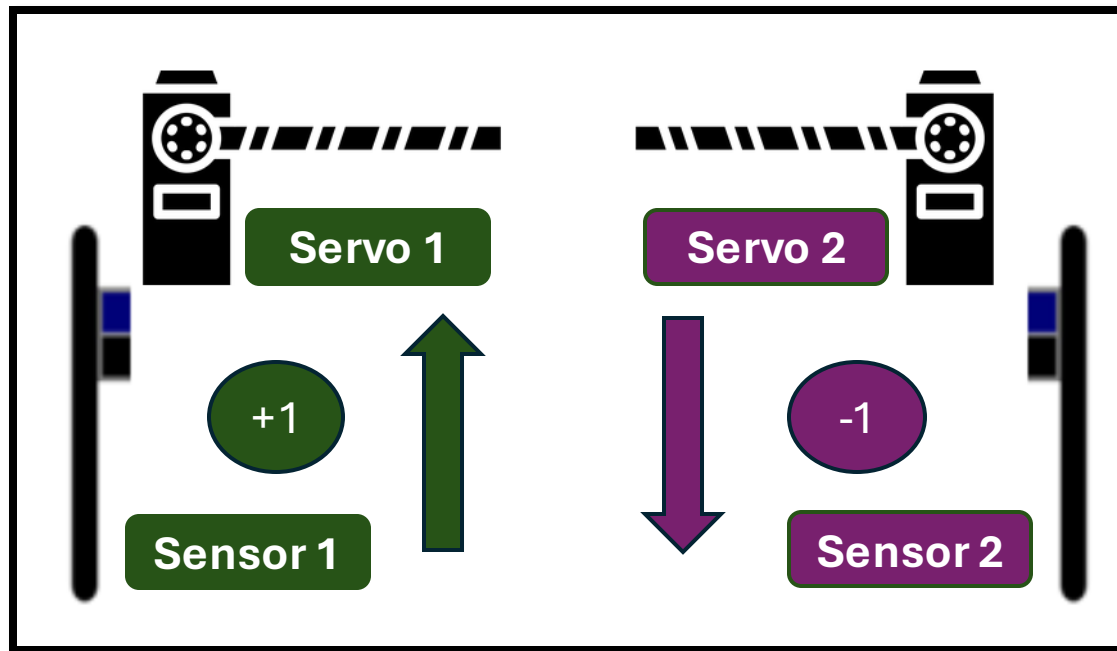
Parking con dos sensores IR y dos servomotores. Solución.



[Enlace al programa](#)

Parking con dos sensores IR y dos servomotores. Avanzado.

En esta práctica añadiremos una variable con la que contaremos la ocupación del parking. Al detectar presencia el sensor 1 añadiremos una plaza a la ocupación y cuando detectemos presencia en la 2 restaremos.



Recuerda iniciar la variable ocupación a cero.

Al incluir la variable es fácil que nos encontremos con el problema de que al detectar presencia nos suma (o resta) varias plazas a la vez. Para solucionar el problema tenemos **dos opciones**:

- **Programar un tiempo de espera** cada vez que lea un 1 en un sensor para que de tiempo al paso del coche.
- **Utilizar el bloque mientras ejecutar** de la categoría de bucles para parar el programa mientras siga detectando presencia.

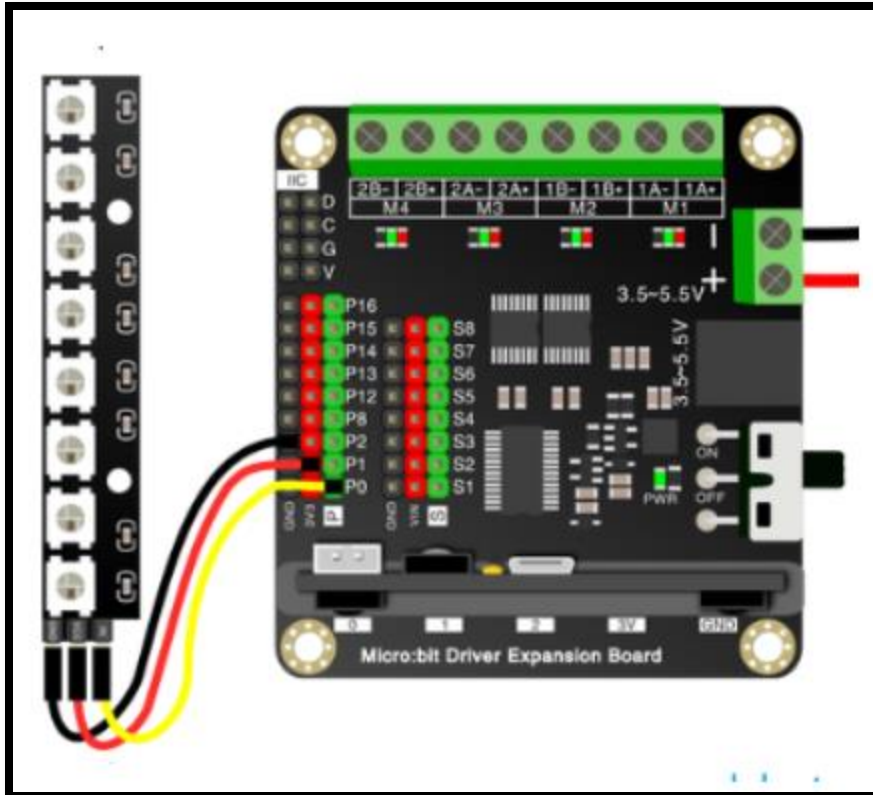
Parking con dos sensores IR y dos servomotores. Avanzado. Solución.



[Enlace al programa](#)

Tira de LEDs.

Vamos a realizar un programa sencillo que nos permita aprender a usar la tira de LEDs. Al iniciar el programa pondremos toda la tira de color verde. Al presionar el botón A encenderemos progresivamente un LED de color rojo y al presionar el botón B revertiremos y lo devolveremos a color verde.



A la izquierda tienes el esquema de conexión de la tira de LEDs. En nuestro caso **la conectaremos a P2** ya que después volveremos a utilizar el montaje de la práctica anterior..

Para poder controlar la tira de LEDs debemos cargar la extensión **Neopixel**.



Tira de LEDs. Programación.

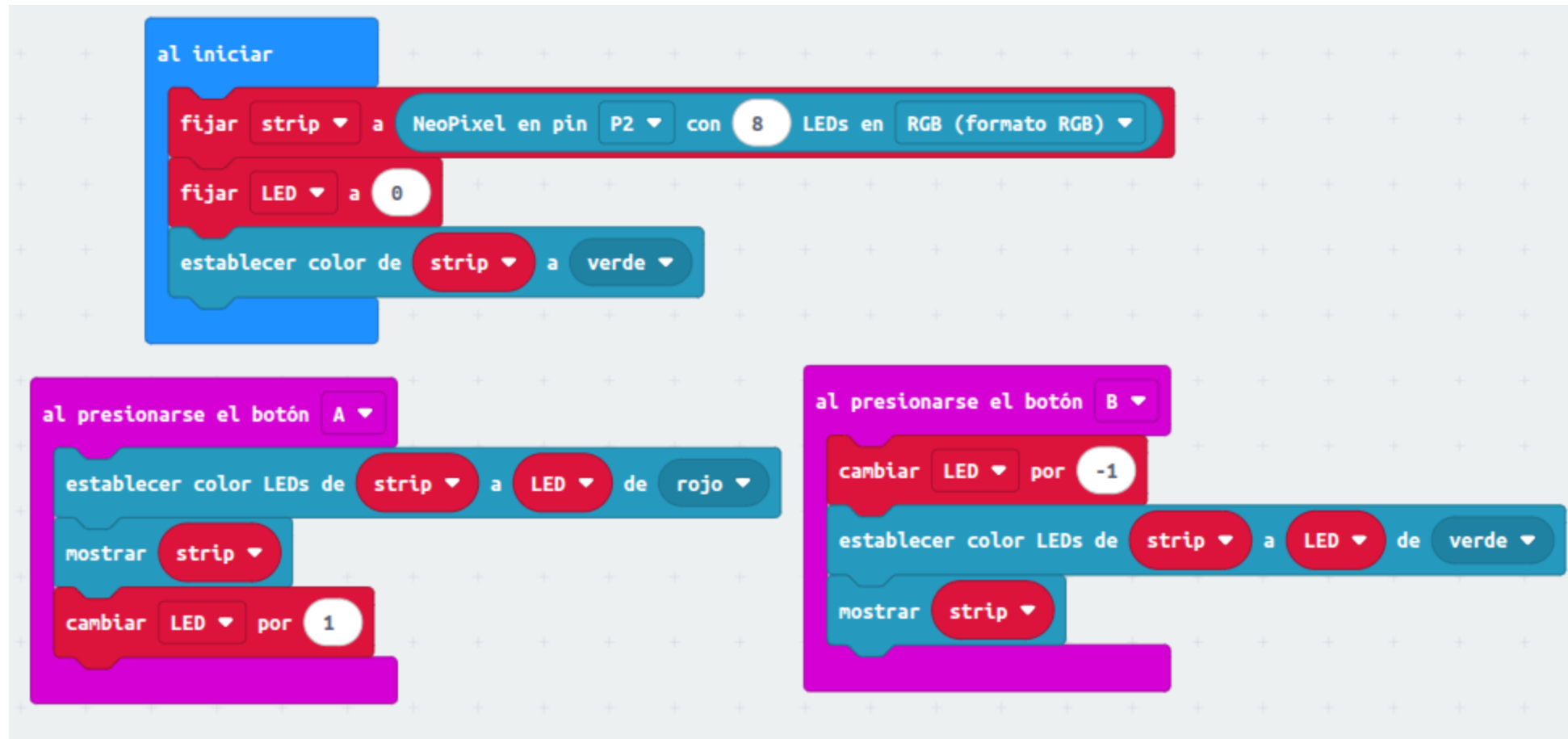
Al iniciar el programa debemos indicar en que pin está conectada la tira y de cuantos RGB dispone. También es conveniente iniciar la variable a cero. Ten en cuenta que además de decir el cambio de color del LED debemos usar el bloque mostrar para que lo podamos visualizar.

Cuando hayas terminado puedes cambiar los colores y funciones de los distintos pulsadores. Por ejemplo, puedes hacer que cuando presiones el logotipo se apague la tira de LEDs.

Bloques que vamos a usar:



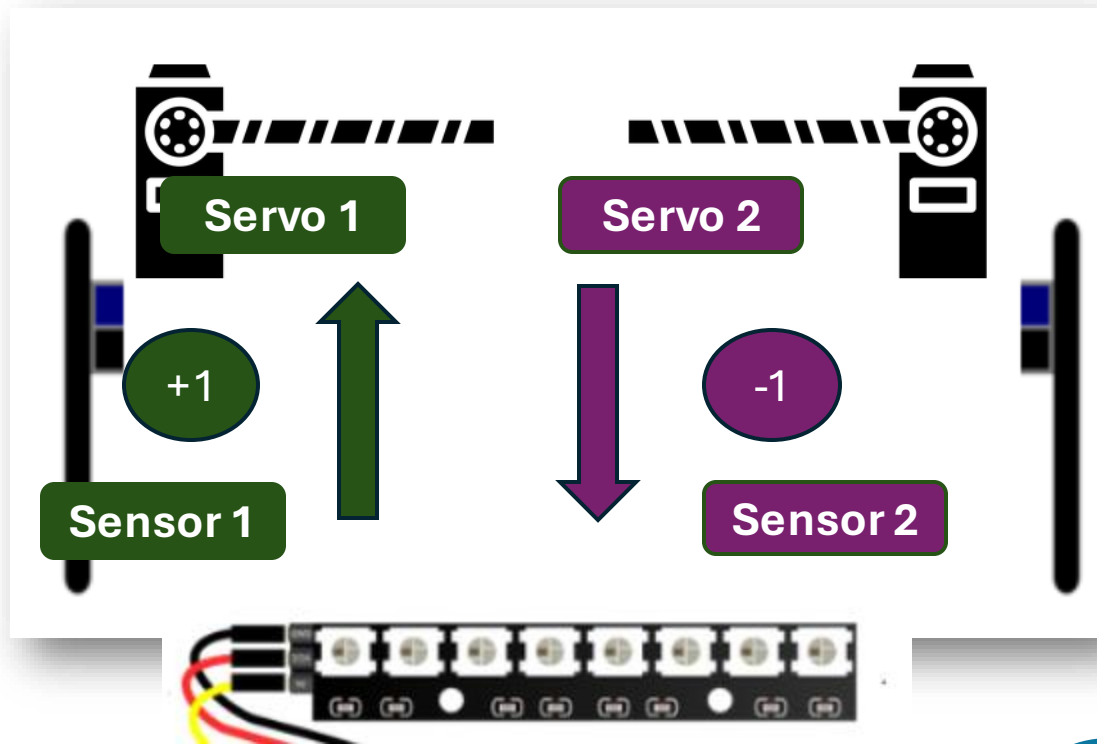
Tira de LEDs. Solución.



[Enlace al programa](#)

Parking con tira de LEDs. Avanzado II.

Vamos a añadir la tira de LEDs RGB de manera que podamos visualizar la ocupación del parking. Para facilitar la programación vamos a suponer que nuestro parking sólo tiene 8 plazas (número de LEDs de la tira). De esta manera, cada vez que entre un coche iluminaremos un LED de rojo y cuando salga de verde.

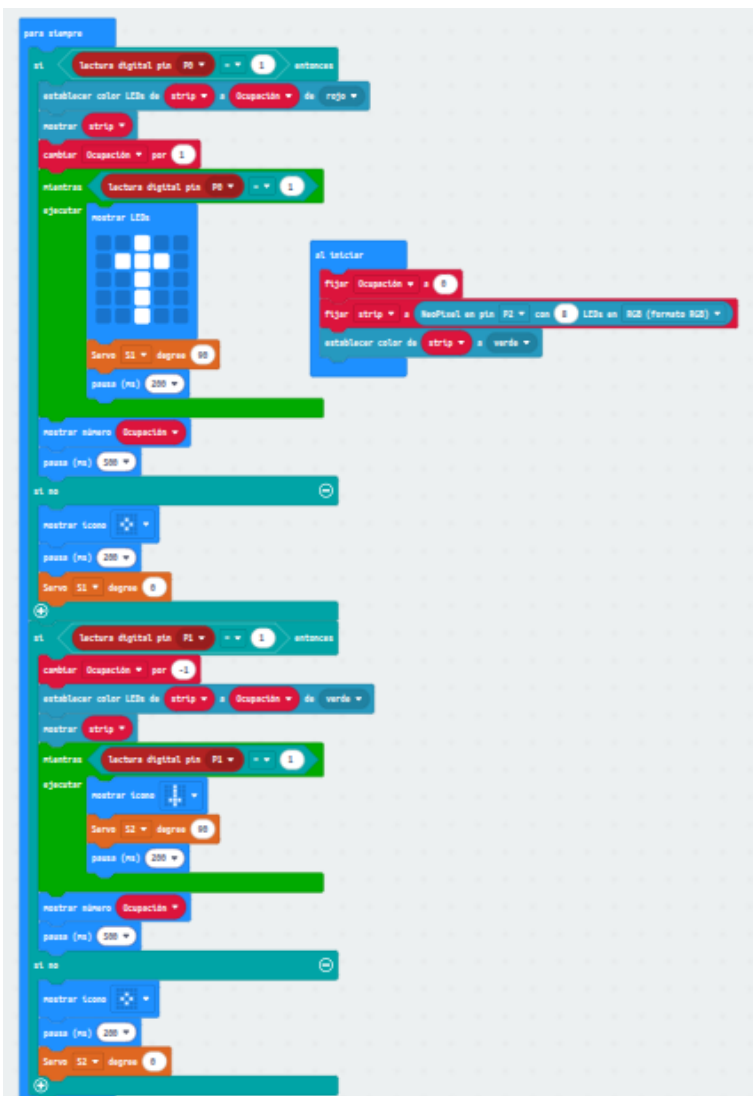


A la izquierda tienes el esquema de conexión de la tira de LEDs. En nuestro caso **la conectaremos a P2** ya que tenemos los otros pines ocupados.

Podemos usar la variable ocupación de manera análoga a cómo hemos utilizado la variable LEDs en la práctica anterior.

Cómo práctica final puedes añadir a la programación que no puedan entrar más coches cuando la variable ocupación llegue a 8.

Parking con dos sensores IR y tira de LEDs. Avanzado II.Solución.



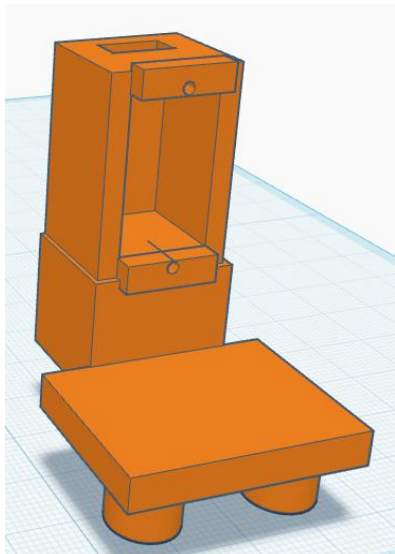
Enlace al programa

Material para imprimir 3D.

A continuación, tenéis unos enlaces para soportes, barreras,... que os pueden venir bien para realizar estas prácticas u otras parecidas.

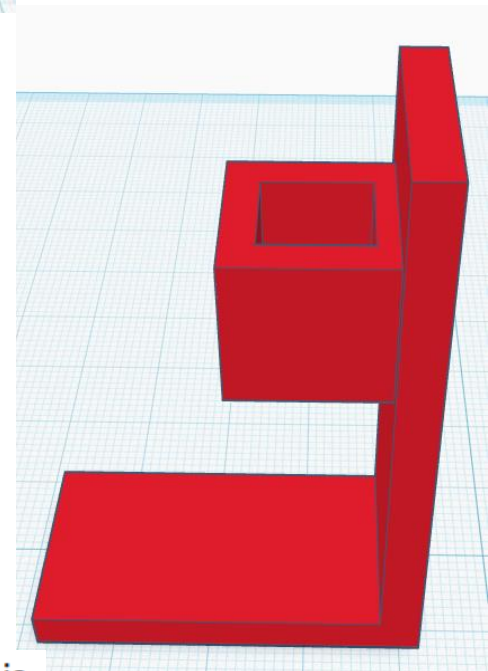


Barrera



Soporte
Servomotor

Soporte
Sensores





¡Muchas gracias!

