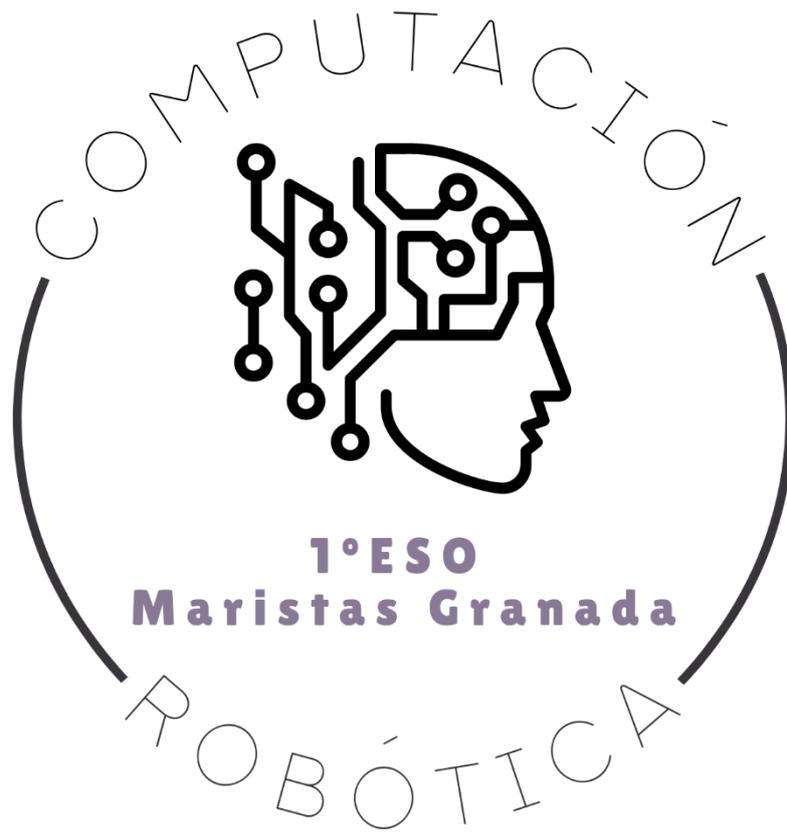


CURSO 2023-2024



RETO 17: SENSORES DE MICROBIT

COMPUTACIÓN Y ROBÓTICA 1ºESO

COLEGIO MARISTA LA INMACULADA

CALLE SÓCRATES, 8
18002 - GRANADA

Índice

0. Ubicación en la programación.....	2
1. Ubicación en la programación.....	3
2. Aprendemos practicando	4
2.1 Sensor de luz	4
2.2 Sensor de sonido.....	6

0. Ubicación en la programación

Título: Reto 17. Sensores de micro:bit

Evaluación: Tercera.

Temporalidad: 1 sesión (2 horas consecutivas)

Índice de contenidos:

Breve resumen de la situación:

1. Ubicación en la programación

...

2. Aprendemos practicando

2.1 Sensor de luz

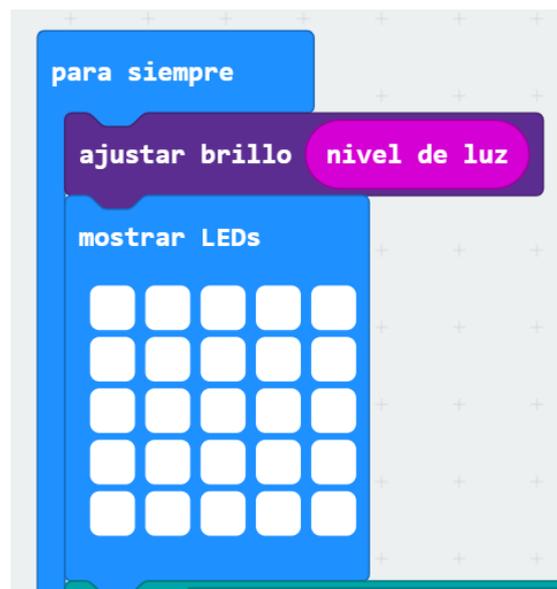
Nuestro primer reto es el siguiente: en una habitación a oscuras, o con poca luz, aplicamos un foco de luz (una linterna, por ejemplo). Y deseamos que maqueen, con ayuda de su sensor de luz, se dirija automáticamente hacia el origen del foco de luz.

Para ello instalamos, como otras veces, la extensión maqueen en nuestro makecode.

Creamos dos variables. Una para guardar el valor actualizado del sensor de luz y otra para determinar el sentido de giro del coche, en búsqueda de la máxima iluminación posible.



Dentro del bloque para siempre, ajustamos el brillo de los LED de micro:bit al nivel recibido por el sensor de luz.



Ahora toca pensar. Nuestra variable “cantidadLUZ” toma el valor 0 al iniciarse el programa. Por lo tanto, a poca cantidad de luz que haya en la habitación, el valor del sensor de luz será mayor que este valor 0.

En este caso, moveremos en línea recta a maqueen para que se acerque al probable foco emisor. Actualizaremos el valor de la variable “cantidadLUZ” y esperaremos una cantidad de tiempo (una décima de segundo) antes de analizar otra vez el valor del sensor de luz.

```

si < nivel de luz > > cantidadLUZ < entonces
  Motor ambos sentido avanzar velocidad 50
  fijar cantidadLUZ a nivel de luz
  pausa (ms) 100
si no
  
```

Si el valor del sensor de luz es menor que el valor almacenado en “cantidadLUZ” significa que nos estamos alejando del foco emisor. Por lo que giraremos a maqueen para reorientar su dirección.

Inicialmente giraremos a la izquierda (variable “girar” igual a 0). Actualizaremos nuevamente el valor de la variable “cantidadLUZ” y esperaremos nuevamente otra décima de segundo antes de seguir con el programa.

```

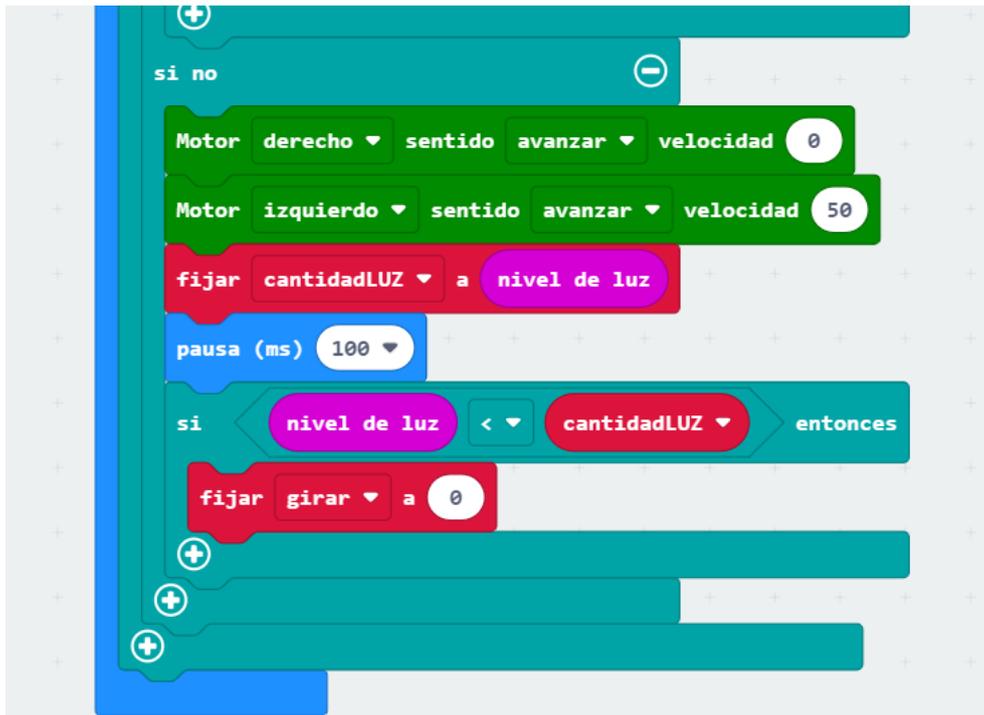
si no
  si girar = 0 entonces
    Motor izquierdo sentido avanzar velocidad 0
    Motor derecho sentido avanzar velocidad 50
    fijar cantidadLUZ a nivel de luz
    pausa (ms) 100
  
```

Pasada la décima de segundo, tendremos que preguntarnos si el valor del sensor de luz sigue siendo inferior al valor actualizado anteriormente en “cantidadLUZ”. En caso afirmativo, tendremos que girar en sentido contrario (hacia la derecha). Para ello, actualizamos la variable “girar” al valor 1.

```

si nivel de luz < cantidadLUZ entonces
  fijar girar a 1
si no
  
```

La parte final del código es idéntica a los bloques contenidos en el condicional Si, con la salvedad de que giraremos a la derecha y de que el cambio de “girar” del final pasaría a 0.



Puedes usar una linterna para probar el programa.

2.2 Sensor de sonido

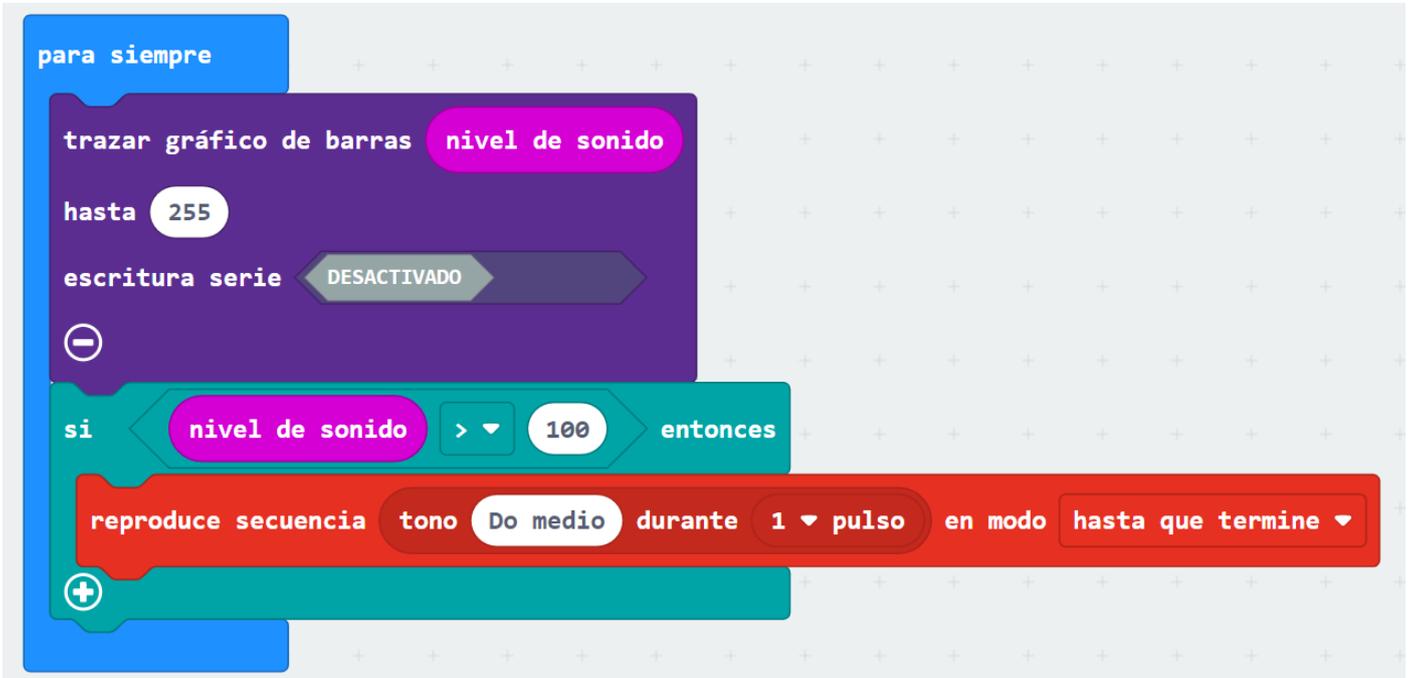
Pasemos ahora al sensor de sonido. Micro:bit tiene asociado un micrófono con el que recoger el sonido del exterior.

Vamos a diseñar un sistema que avise, con un zumbido, cuando superamos un nivel de ruido determinado.



Sobre este modelo base, vamos a introducir la siguiente mejora: la pantalla de diodos LED se encenderá en función del nivel de ruido registrado por el micrófono. El sensor de sonido también se mueve en el rango [0, 255].

Podemos conseguirlo automáticamente con el bloque “trazar gráficos de barra”, ubicado en la sección LED.

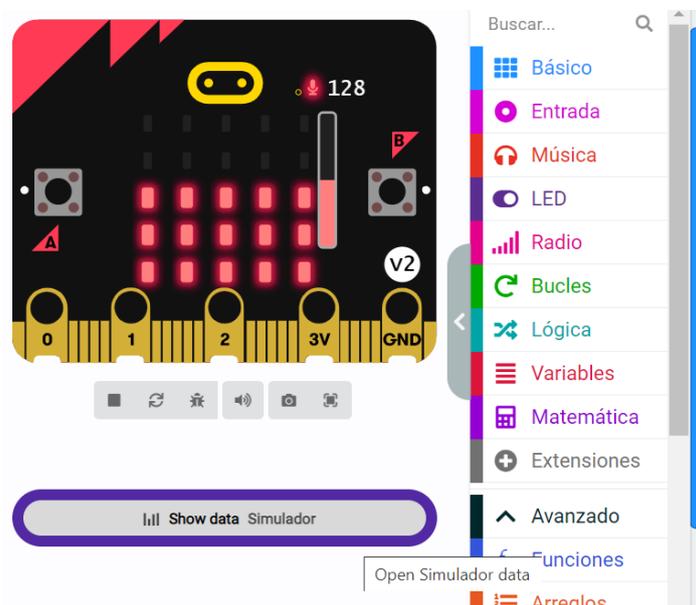


¿Cómo funciona este nuevo bloque?

Ilumina los LED con una proporcionalidad directa (regla de tres que has estudiado en matemáticas) que depende del valor indicado (en nuestro caso, nivel de sonido). En el campo “hasta” indicamos el valor máximo que puede tomar nuestro sensor de sonido (255), y que provocará el encendido de todos los LED.

¿Qué significa la opción “escritura serie”?

Significa que puedo guardar los datos recogidos del sensor a lo largo del tiempo, pulsando en la opción “Show data simulador”.



Cuando vemos los datos, un gráfico nos muestra (para distintos valores de tiempo) el valor registrado por el micrófono (en el intervalo de 0 a 255).



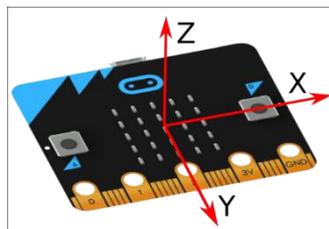
Incluso podemos exportar los datos como un archivo .csv (que se puede abrir con una hoja de cálculo EXCEL) para visualizar los datos y trabajar con ellos.

21.617	168
22.161	194
22.695	212
23.226	226
23.759	226
24.295	232
24.827	232
25.359	243
25.891	215

La primera columna de datos representa el tiempo en segundos (fíjate que la tercera cifra decimal representa milisegundos), mientras que la segunda columna de datos son los niveles de sonido (de 0 a 255) asociados a cada tiempo.

2.3 Sensor de aceleración (acelerómetro)

La placa tiene integrada un sensor de aceleración en cada una de las componentes espaciales (x, y, z).



La imagen superior muestra los ejes de cada una de las componentes.

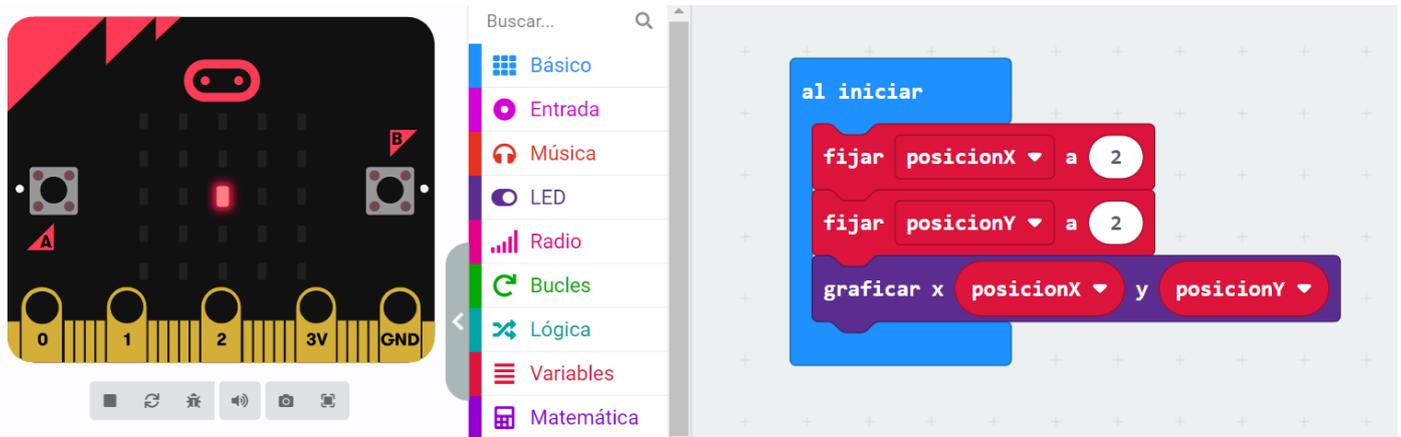
Si giro la placa en la dirección indicada por el eje X, se activa (por gravedad) el acelerómetro de la componente X.

Si giro la placa en la dirección indicada por el eje Y, se activa (por gravedad) el acelerómetro de la componente Y.

Y si subo y bajo la placa en la dirección marcada por el eje Z, se activa (por la fuerza aplicada al subir y bajar) el acelerómetro de la componente Z.

Utilizando esta utilidad, vamos a mover un LED encendido a lo largo de toda la placa.

Creamos dos variables: "posicionX" y "posicionY". Iniciamos ambas variables a la posición (2,2) para que se encienda el LED del centro de la pantalla (la primera componente es el valor horizontal X, mientras que la segunda componente es el valor vertical Y).



Si giramos la placa hacia la derecha en la dirección X, deberemos sumar una unidad a "posicionX". ¡OJO! Esto lo haremos siempre que el valor de "posicionX" sea menor que 4 (ya que no hay LEDs más allá del valor 4 de la componente horizontal).

