

Guía inicio kit BOSON



Índice

ÍNDICE.....	1
CAPÍTULO 1: INTRODUCCIÓN	3
PLUG PLAY	3
AGARRE MÚLTIPLE	3
INTEGRACIÓN CON MICRO:BIT	3
CAPÍTULO 2: MAKECODE Y MICRO:BIT	4
PRUEBA INTRODUCCIÓN A MAKECODE	4
BREVE INTRODUCCIÓN A MICRO:BIT	4
CÓMO UTILIZAR EL MICRO:BIT	4
PASO 1: Abre MakeCode	5
PASO 2: Conecta el micro:bit al ordenador	5
PASO 3: Iniciar un nuevo proyecto	6
PASO 4: Descargar el programa y subirlo a micro:bit	7
CAPÍTULO 3: ¿QUÉ HACE QUE UNA MÁQUINA “COBRE VIDA”?	9
DISPOSITIVOS INTERACTIVOS	9
Unidad de entrada: sensor	9
Unidad de control: micro:bit	9
Unidad de salida: actuador	9
RELACIÓN ENTRE EL PROGRAMA Y EL HARDWARE	9
LA PLACA DE EXPANSIÓN BOSON PARA MICRO:BIT	9
CAPÍTULO 4: ¡MANOS A LA OBRA!	11
PROYECTO 1: EL MISTERIOSO MICRO:BIT	11
LISTA DE COMPONENTES	11
CONEXIÓN	11
PROGRAMA	11
EJERCICIOS	13
PROYECTO 2: LUZ LED PARPADEANTE	14
LISTA DE COMPONENTES	14
CONEXIÓN	14
PROGRAMA	15
EJERCICIO	16
PROYECTO 3: LUZ DE NOTIFICACIÓN	17
LISTA DE COMPONENTES	17
CONEXIÓN	17
PROGRAMA	18
EJERCICIO	20
PROYECTO 4: VENTILADOR ELÉCTRICO	21
LISTA DE COMPONENTES	21
CONEXIÓN	21
PROGRAMA	22
CAPÍTULO 5: UN PASO MÁS ALLÁ	25
PROYECTO 1: VELA ELECTRÓNICA	25
LISTA DE COMPONENTES	25
CONEXIÓN	25
PROGRAMA	26

PROYECTO 2: PUERTA AUTOMÁTICA.....	28
LISTA DE COMPONENTES	28
CONEXIÓN	28
PROGRAMA	29
PROYECTO 3: CAJA DE MÚSICA.....	31
LISTA DE COMPONENTES	31
CONEXIÓN	31
PROGRAMA	32
PROYECTO 4: TIRA DE LED DE COLORES.....	35
LISTA DE COMPONENTES	35
CONEXIÓN	35
PROGRAMA	36
CAPÍTULO 6: CONVIÉRTETE EN UN EXPERTO	42
PROYECTO 1: ESTABILIZADOR ELECTRÓNICO.....	42
LISTA DE COMPONENTES	42
CONEXIÓN	42
PROGRAMA	43
EJERCICIOS	44
PROYECTO 2: PANEL DE DJ.....	45
LISTA DE COMPONENTES	45
CONEXIÓN	45
PROGRAMA	46
EJERCICIOS	49
PROYECTO 3: TIMBRE A DISTANCIA.....	50
LISTA DE COMPONENTES	50
CONEXIÓN	50
PROGRAMA	51
EJERCICIO.....	53
PROYECTO 4: ESCAPAR DEL LABERINTO	54
LISTA DE COMPONENTES	54
CONEXIÓN	54
PROGRAMA	55
EJERCICIO.....	61
<i>Por ejemplo: Cómo crear una casa inteligente con micro:bit y BOSON.....</i>	<i>62</i>

No olvides visitar el blog de DFRobot para ver más tutoriales, proyectos y las últimas tendencias.
Encontrarás el enlace en la esquina superior izquierda de la página.

Capítulo 1: Introducción

Boson es una plataforma de módulos electrónicos pensados en el aprendizaje y enseñanza de los más pequeños del hogar. Dentro de su diseño se destacan elementos como un código de colores para diferenciarlos según su función, flexibilidad de armado y compatibilidad con LEGO.

En esta guía encontrarás diferentes mini-proyectos que puedes armar para comenzar tu aprendizaje con los módulos Boson. Antes de esto hemos de familiarizarnos con el funcionamiento de estos módulos y su uso.

Plug Play

Todos los módulos Boson tienen conectores en común, que hace que poder utilizar los módulos sea tan sencillo como llegar y conectar un cable entre dos módulos diferentes. De esta manera no se requiere soldar y se asegura una estabilidad en la señal tanto de datos como de poder, además de evitar problemas por conexiones en inversa.



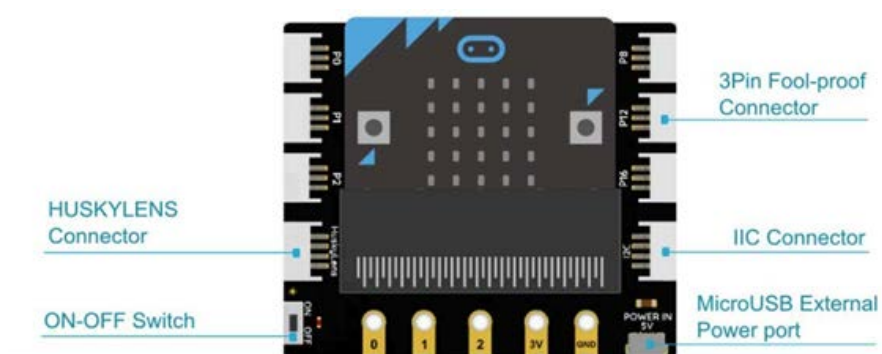
Agarre múltiple

Los módulos Boson incluyen imanes que te permitirán pegarlos donde sea: una pizarra, en el refrigerador o cualquier otra superficie en donde podrías pegar un imán (¡como otro módulo Boson!). Además, cada módulo viene con bases compatibles con bloques LEGO y tornillos para fijarlos



Integración con Micro:bit

Puedes aprovechar todas las virtudes de micro:bit a través de la placa de expansión de Boson. Con ella podrás conectar cualquier módulo que desees y verás como se comportarán como si siempre hubieran sido uno.



Capítulo 2: MakeCode y micro:bit

Prueba Introducción a MakeCode

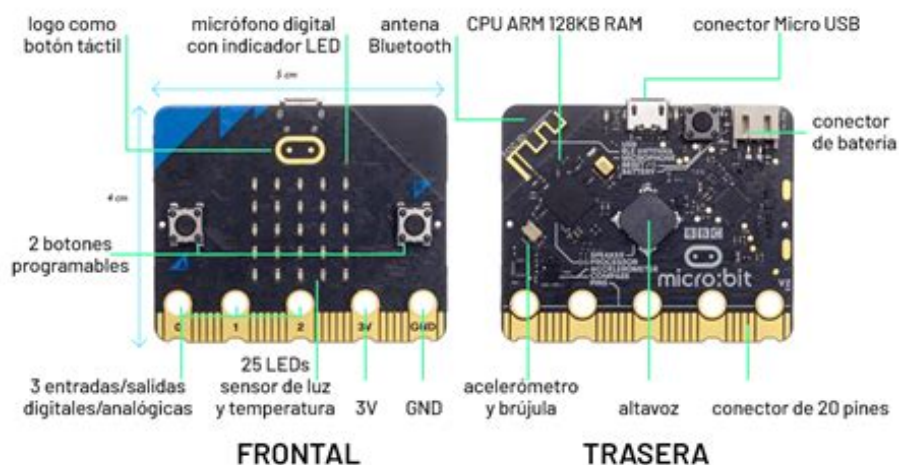
MakeCode para micro:bit es uno de los entornos de programación gráfica más utilizados micro:bit. Se trata de un proyecto de código abierto desarrollado por Microsoft.

Breve introducción a micro:bit

micro:bit es un microcontrolador de bolsillo diseñado para niños y principiantes que están aprendiendo a programar, lo que les permite plasmar fácilmente sus ideas en juegos digitales de creación propia, proyectos interactivos y robótica.

micro:bit incluye una variedad de módulos integrados, entre los que se encuentran una matriz LED de 5x5 (que también incluye detección de luz), 2 botones programables, un detector de movimiento, una brújula y un módulo Bluetooth® Smart. Además, se pueden conectar más módulos, como un servomotor o luces LED RGB, a través de los 5 puertos E/S o 20 conectores.

El micro:bit se puede utilizar para hacer realidad muchas ideas geniales. Todo lo que puedas imaginar: un robot, un instrumento eléctrico o incluso un sistema de domótica. ¡Las posibilidades son infinitas! El micro:bit cuenta con un sinnúmero de características innovadoras, como 25 LED rojos para mostrar mensajes y dos botones programables para controlar juegos o música. Puede detectar movimiento, reconocer gestos y conectarse con otros dispositivos o con Internet a través de una conexión Bluetooth.



El micro:bit está equipado con sensores de luz y temperatura, así como con otros dispositivos de detección habituales, lo que significa que, por sí solo, también puede dar lugar a muchos productos inteligentes comunes que utilizamos en nuestra vida cotidiana.

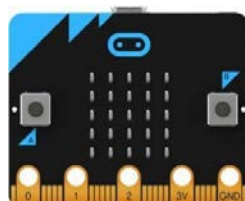
Cómo utilizar el micro:bit

Si eres nuevo en el mundo del micro:bit, puedes empezar con la plataforma de programación en línea, el editor MakeCode, para aprender a programar el micro:bit.

Antes de empezar, asegúrate de tener a mano los siguientes elementos. Además, necesitarás un ordenador con sistemas operativos como Windows 10, Mac OS o Linux y conexión a Internet.

LISTA DE COMPONENTES:

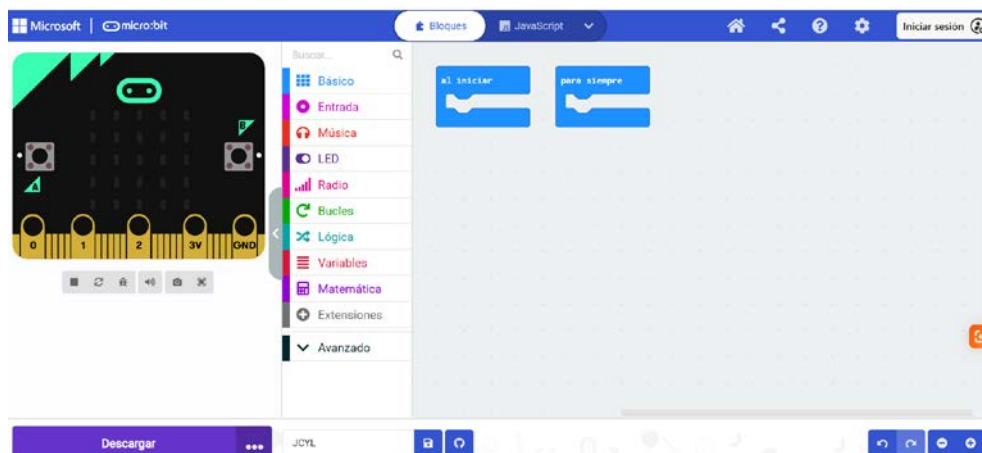
Placa principal de micro:bit



Los siguientes pasos se basan en el sistema operativo Windows 10. Pueden servir de referencia para otros sistemas operativos.

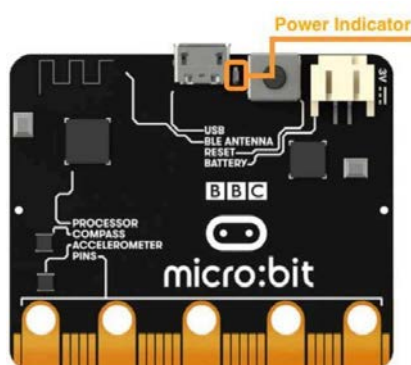
PASO 1: Abre MakeCode

Accede a la página de MakeCode desde el siguiente enlace: <https://makecode.microbit.org/>

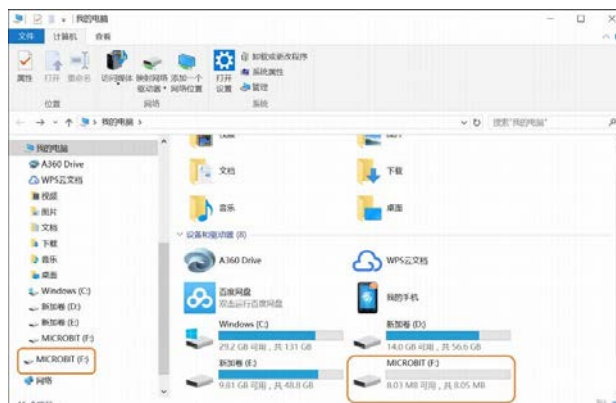


PASO 2: Conecta el micro:bit al ordenador

El micro:bit se conecta al ordenador mediante un cable USB. El indicador de alimentación situado en la parte trasera del micro:bit se iluminará cuando esté conectado.



Antes de programar, debemos asegurarnos de que el ordenador reconoce la placa base. Cuando el micro:bit esté conectado, aparecerá una carpeta llamada "MICROBIT" en "Mi PC".



PASO 3: Iniciar un nuevo proyecto

Antes de iniciar un nuevo proyecto, primero tendremos que familiarizarnos con la interfaz de programación.

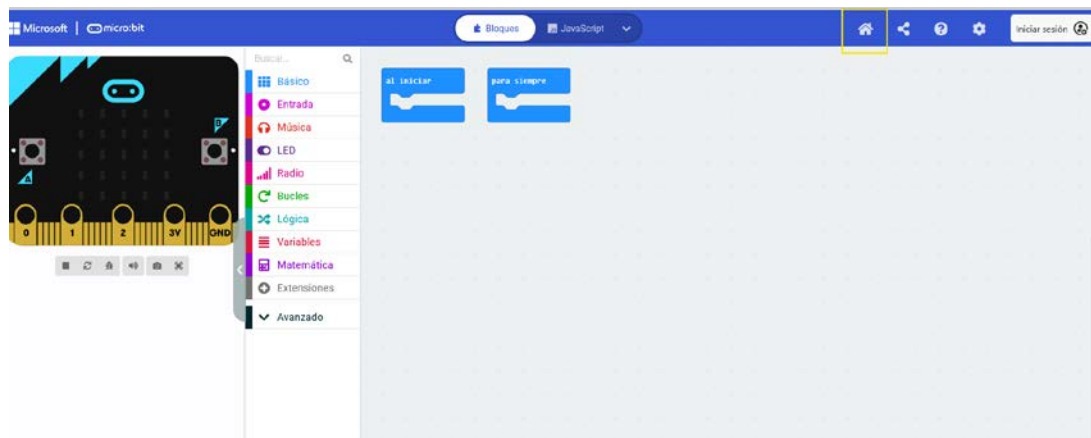


Ventana de simulación: simula el estado de funcionamiento del micro:bit. Durante el proceso de programación, siempre puedes comprobar cómo se ve tu programa a través de esta ventana.

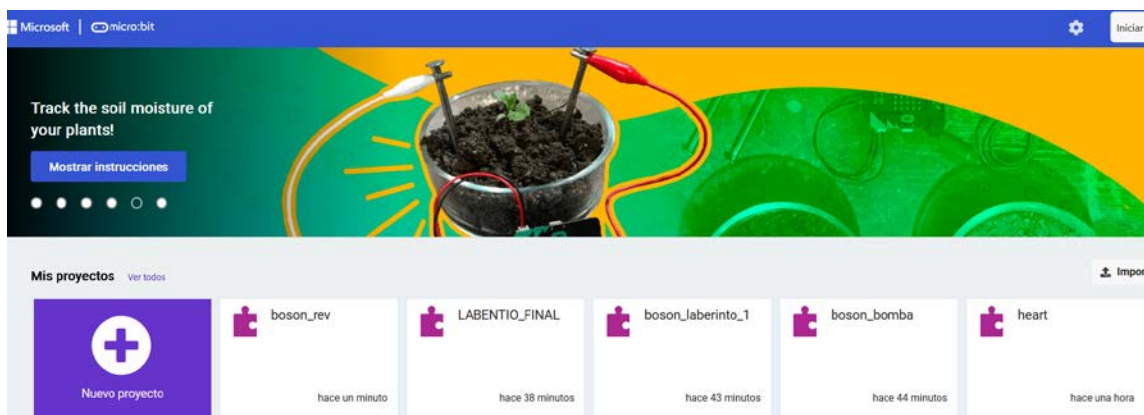
Área de funciones: aquí encontrarás todos los bloques de funciones, incluyendo entrada, salida, bucle, lógica, etc.

Área de programación: arrastra los bloques desde el “Área de funciones”, apíalos y crea tu programa aquí.

Haz clic en “inicio” en la parte superior de la ventana de simulación.

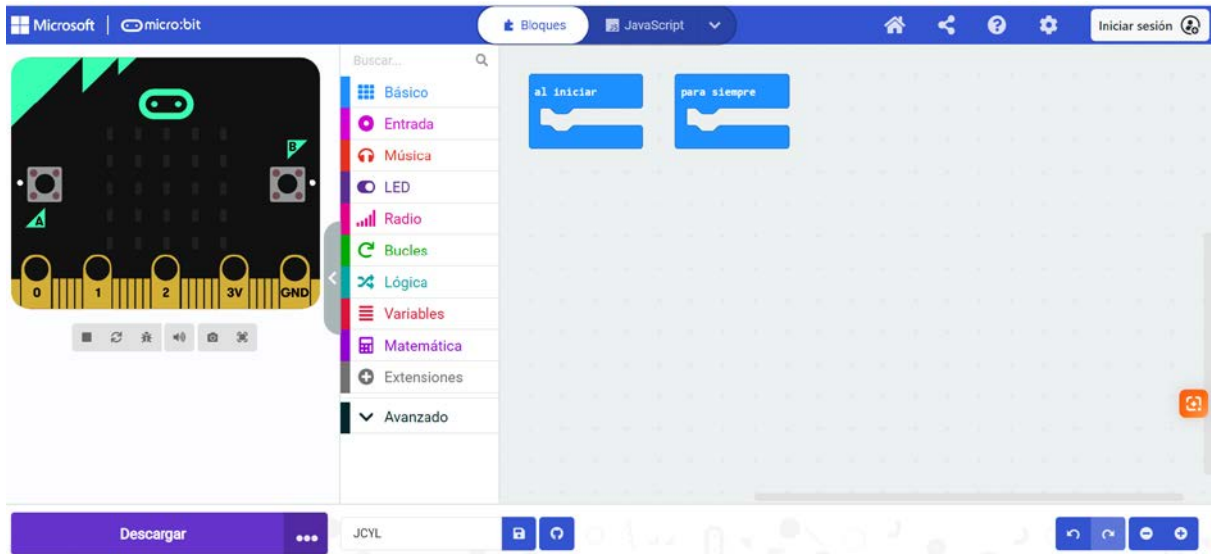


A continuación, haz clic en “nuevo proyecto”.

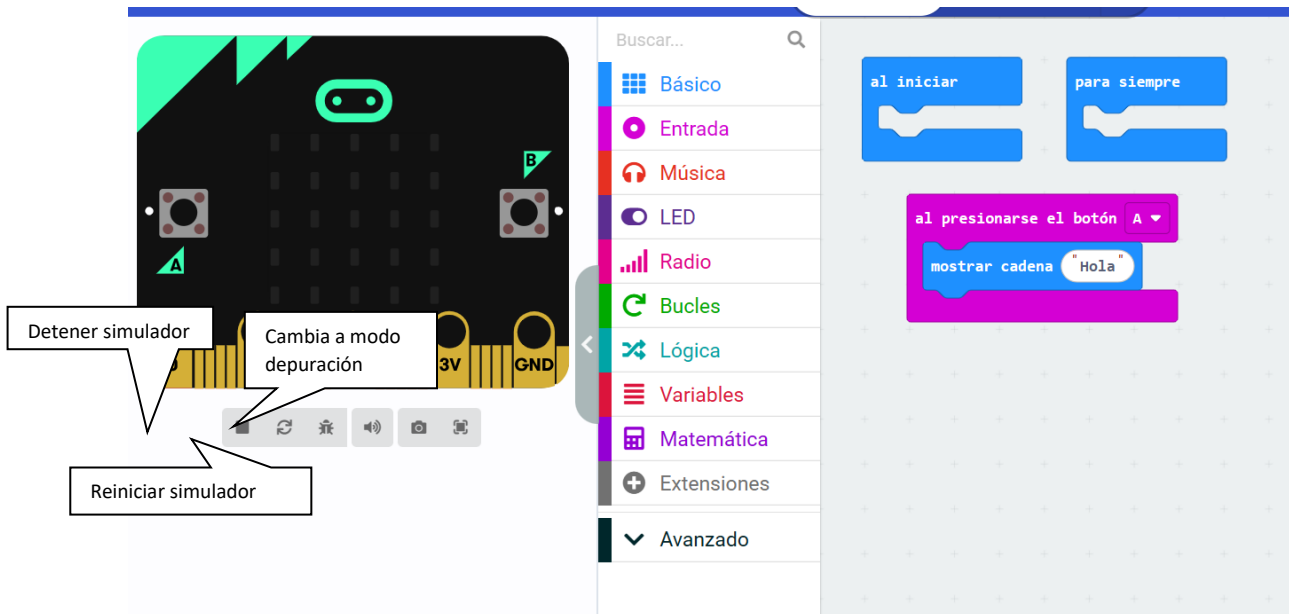


PASO 4: Descargar el programa y subirlo a micro:bit

Una vez que hayamos terminado nuestro programa, ya podemos descargarlo de la página web y subirlo al micro:bit. También podemos cambiar el nombre del proyecto y guardarlo en el navegador. El proyecto se guardará en “Mis cosas”.



Antes de descargarlo, podemos comprobar los resultados simulados en la ventana de simulación. Los botones de la parte inferior de la ventana de simulación se pueden utilizar para controlar el micro:bit analógico.



Haz clic en “Descargar” () en la parte inferior de la ventana de simulación y selecciona “Guardar como” en el cuadro de diálogo emergente.

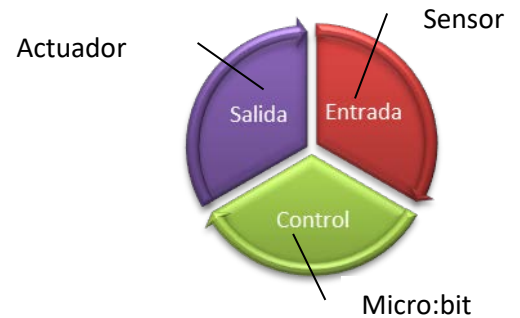
Capítulo 3: ¿Qué hace que una máquina “cobre vida”?

Dispositivos interactivos

En los próximos capítulos, construiremos proyectos interactivos, como un LED único controlado por un botón, una animación sencilla y colorida con LED, o incluso crear nuestra propia música. Estos dispositivos pueden clasificarse como “dispositivos interactivos”. Para comprender mejor su funcionamiento, primero debemos entender cómo están compuestos.

El dispositivo interactivo más sencillo consta de tres partes:

- Una unidad de entrada para la recepción de comandos o la recopilación de datos
- Unidad de control para el procesamiento de datos o señales
- Unidad de salida para enviar datos o ejecutar acciones



Seguramente te sentirás confundido con estos conceptos y tendrás ganas de saber cómo funcionan realmente. Tomemos nuestro propio cuerpo como ejemplo: recopilamos información en forma de luz, sonido, sabor y energía a través de los ojos, los oídos, la nariz y la piel. Esta información llega a nuestro cerebro y determina qué respuesta dar. En última instancia, llevamos a cabo acciones físicas basadas en esta información para modificar el entorno material. Concretamente, si un amigo tuyo, por ejemplo, te dice “hola”, y tú respondes “hola”. En este caso, tus oídos actúan como unidades de entrada, tu cerebro como unidad de control y tu boca desempeña el papel de unidad de salida.

Del mismo modo, al crear proyectos con el micro:bit, utilizaremos diversos sensores como unidades de entrada, el micro:bit como unidad de control y el actuador como unidad de salida.

Unidad de entrada: sensor

Los sensores (también llamados transductores) son componentes físicos que detectan las características del entorno, como la luz, la temperatura, la humedad, etc., y las convierten en datos de señal. En este tutorial se utilizarán sensores como botones, sensores de sonido y sensores de temperatura.

Unidad de control: micro:bit

El micro:bit actuará como unidad de control en este tutorial, utilizando los pines de señal para establecer conexiones entre las unidades de entrada y las de salida, y procesando los datos en el módulo de procesamiento de cálculos.

Unidad de salida: actuador

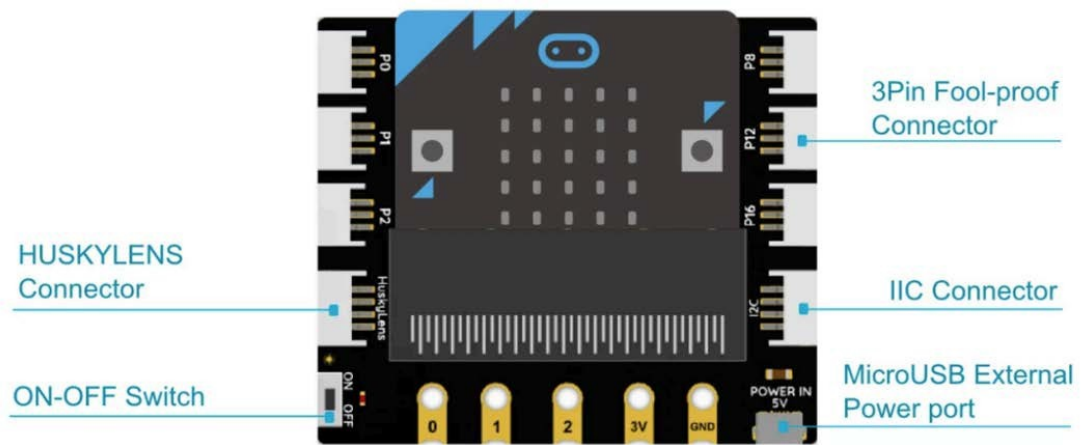
El actuador es el responsable de mover o controlar un sistema o un mecanismo. Convierte la energía eléctrica en movimiento, sonido y luz. En este tutorial se utilizarán actuadores como LED, pequeños ventiladores y servomotores.

Relación entre el programa y el hardware

La unidad de entrada, la unidad de control y la unidad de salida mencionadas anteriormente son componentes físicos. En el caso de un ser humano, el hardware incluye el cuerpo, el cerebro y las extremidades. Sin embargo, lo que realmente controla el comportamiento de nuestro cuerpo físico es nuestra “mente”. El programa, en este caso, es como nuestra mente, y lo que vamos a explicar es cómo configurar la “mente” del micro:bit para que haga todo lo que le pidamos.

La placa de expansión Boson para micro:bit

La placa de expansión Boson permite conectar módulos externos al micro:bit, como pulsadores, módulos de interruptores y sensores. No solo simplifica la conexión de los circuitos, sino que también potencia las capacidades del micro:bit.



Capítulo 4: ¡Manos a la obra!

A estas alturas ya debes tener una idea aproximada de qué son el micro:bit y la placa de expansión, y de cómo funciona MakeCode. Seguro que estás deseando probarlo por ti mismo. ¡Pues bien, comencemos el viaje mágico de Boson con el micro:bit!

Proyecto 1: El misterioso micro:bit

Comenzaremos nuestro primer proyecto, un sencillo panel de emojis, para familiarizarnos un poco más con lo que acabamos de aprender en el capítulo anterior.

Los emojis se pueden ver en casi todos los smartphones y ordenadores, y pueden tener el siguiente aspecto:



Amo



sé guay

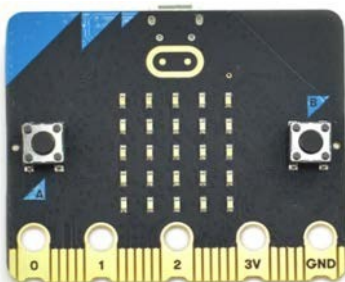


te estoy mirando

En este capítulo, te enseñaremos a diseñar tus propios emojis programando el micro:bit. El micro:bit cuenta con 5×5 (25) LED integrados. Podemos controlar estos LED para formar diferentes patrones y mostrar los emojis.

Lista de componentes

1 × micro:bit



1 × cable USB

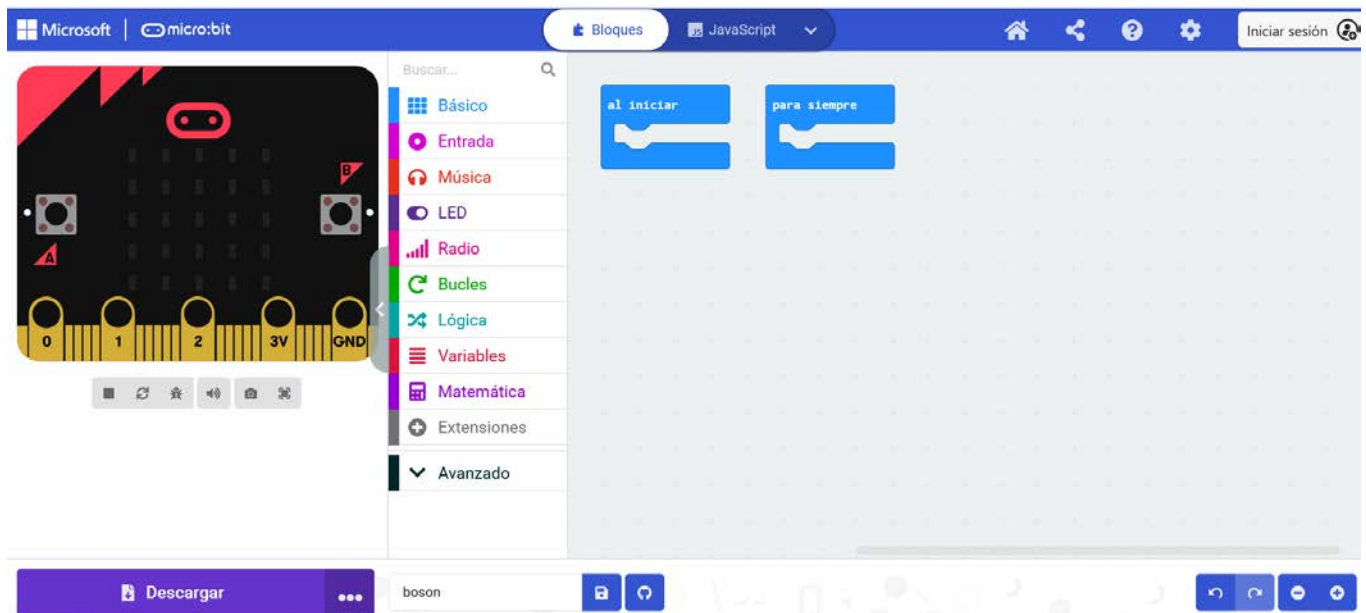


Conexión

Conecta el micro:bit al ordenador mediante el cable USB.

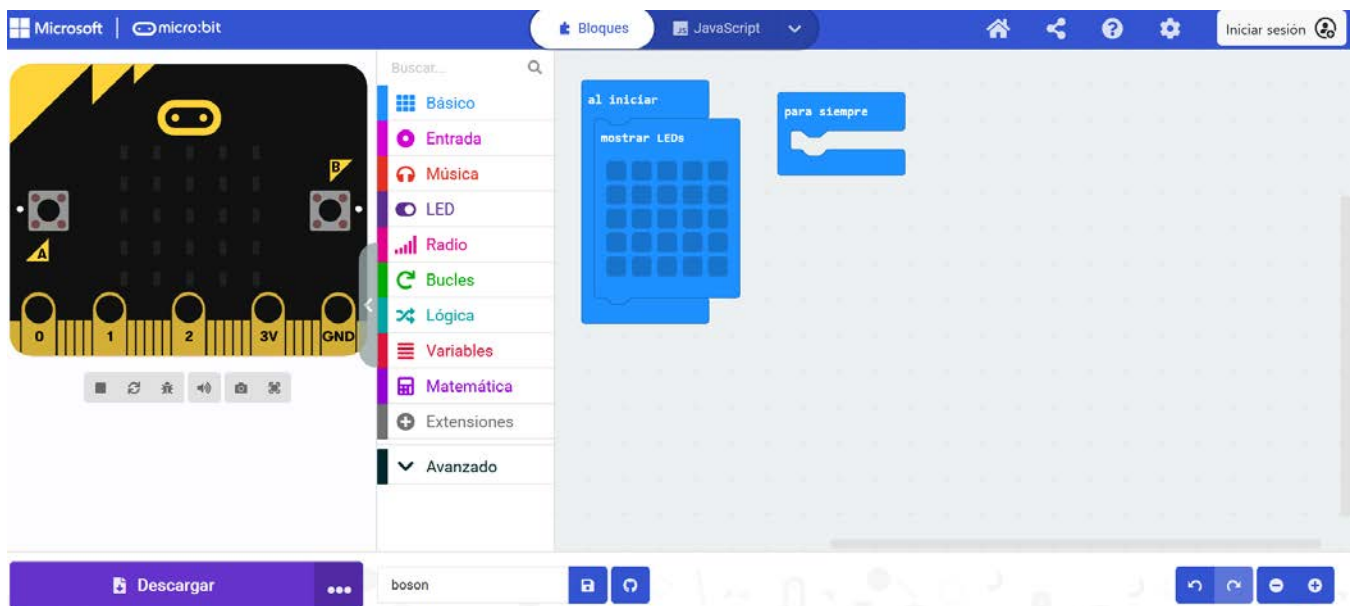
Programa

PASO 1: Inicia un nuevo proyecto. Si has olvidado cómo hacerlo, puedes volver al último capítulo. PASO 2: Haz clic en “Basico” en la parte superior del área de funciones y busca la función “mostrar LEDs”.

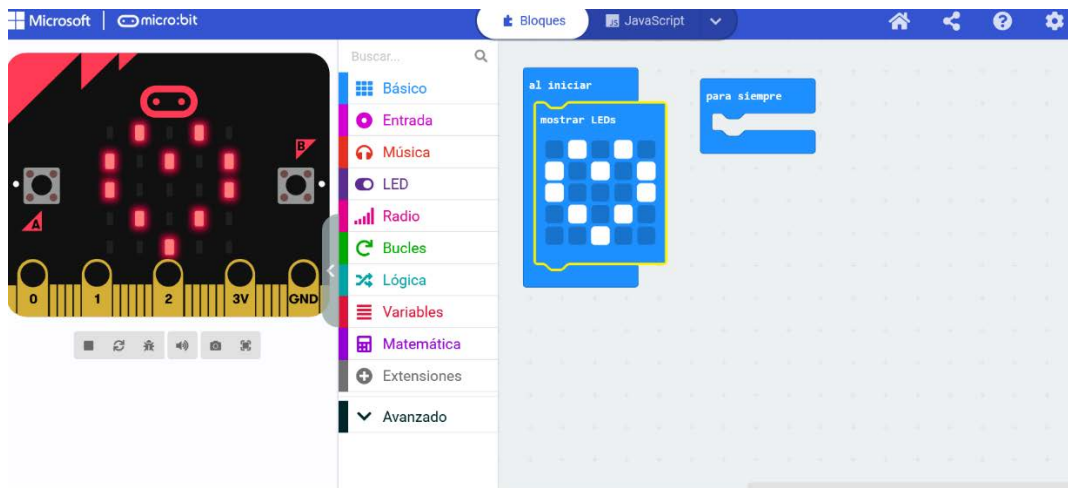


PASO 3: El módulo “al iniciar” aparecerá automáticamente en el área de programación la primera vez que abras la página web de MakeCode, o bien puedes arrastrar la función “al iniciar”, situada en la parte inferior de “Basic”, al área de programación.

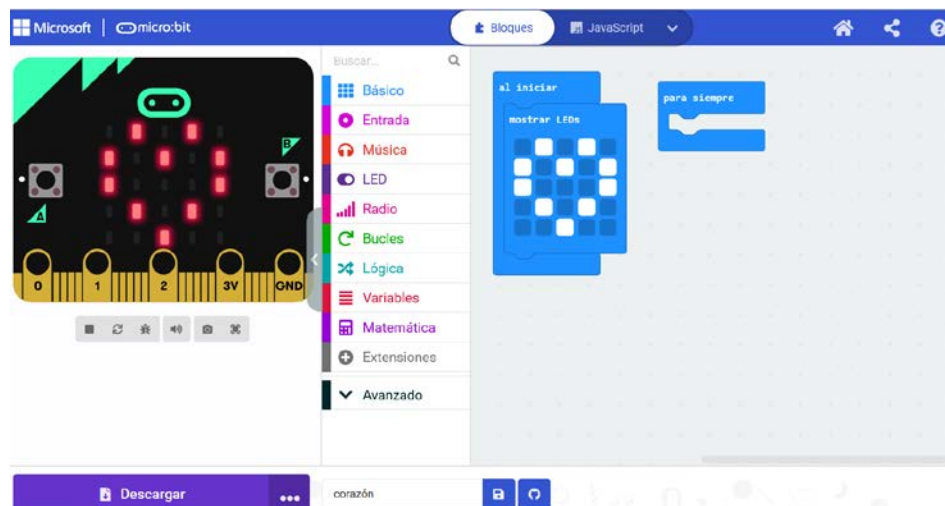
PASO 4: Arrastra la función “mostrar LEDs” al área de programación y colócala dentro de “al iniciar”.



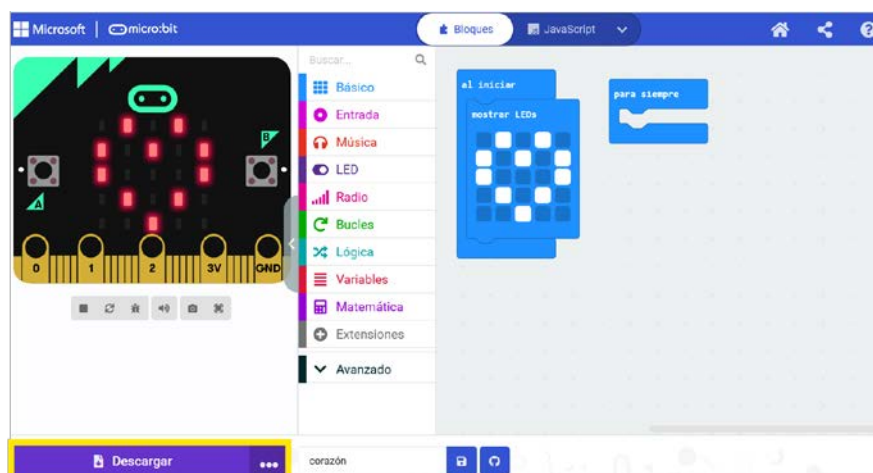
Cuando la función “mostrar LEDs” se encuentre dentro del módulo “al iniciar”, podrás hacer clic en el punto azul claro dentro del bloque para editar el patrón. Además, todo lo que dibujemos dentro del bloque se mostrará en la ventana de simulación.



PASO 5: Aquí dibujamos un corazón en el bloque. Para recordarlo mejor, podemos cambiarle el nombre a “corazón”.



PASO 6: Haz clic en “Descargar” para guardar el archivo del MICROBIT en el ordenador, o bien puedes guardarlo directamente en micro:bit. Si todo va bien, el micro:bit mostrará entonces un emoji de “corazón”. (Si olvidas cómo subir el código, vuelve atrás y consulta el PASO 4 del capítulo 1)



Ejercicios

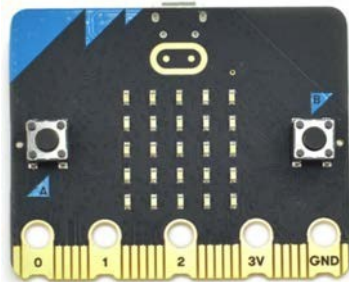
Intenta diseñar un emoji que pueda alternar entre dos patrones diferentes.

Proyecto 2: Luz LED parpadeante

En la parte anterior, hemos aprendido a programar la placa LED para crear nuestros propios emojis. En este capítulo, conectaremos un módulo LED externo y aprenderemos a encenderlo y a hacer que parpadee.

Lista de componentes

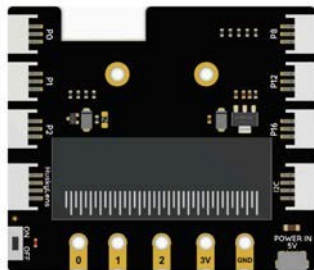
1 × micro:bit



1 × módulo LED de



1 × placa de expansión Boson



1 cable USB de

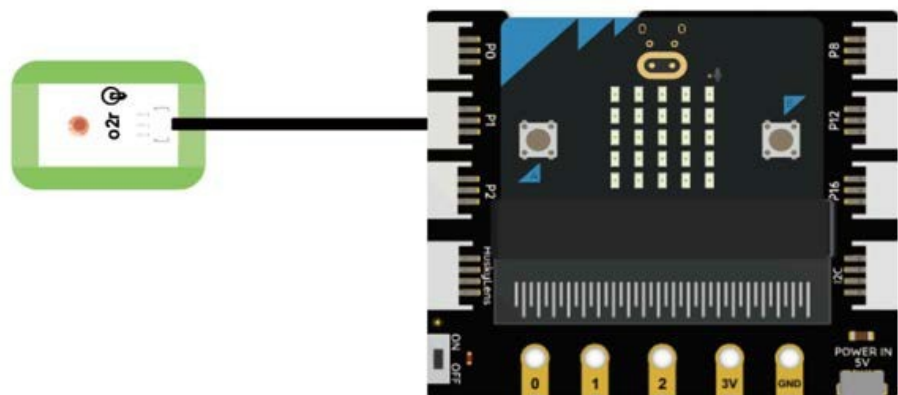


Conexión

- Inserta el micro:bit en la placa de expansión.

Nota: el micro:bit permanecerá dentro de la placa de expansión en todos los capítulos siguientes. Nos saltaremos este paso en todos los capítulos siguientes.

- Conecta el módulo LED a P1



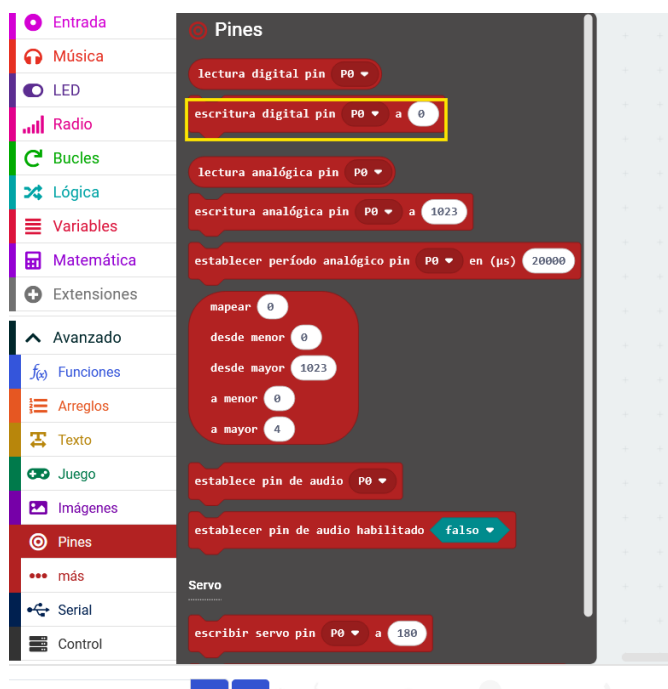
Programa

Tarea 1: Enciende el LED

PASO 1: Abre la página web de MakeCode : <https://makecode.microbit.org> and y crea un nuevo proyecto.

PASO 2: En esta ocasión, dado que el módulo LED está conectado al P1 del micro:bit como un componente externo, para controlar el LED tendremos que establecer el estado del pin al que está conectado. El bloque de funciones “escritura digital” es el que se encarga de ello. “Escritura digital” puede hacer que el pin emita un voltaje alto (representado como 1) o un voltaje bajo (representado como 0), lo que enciende y apaga el LED, respectivamente. Primero, encendemos el LED. Para ello, ve a

“Avanzado” -> “Pines” y configura el pin P1 de escritura digital en HIGH (1).



PASO 3: Coloca la función “escritura digital” en un bucle “para siempre” (el bucle “para siempre” aparecerá en el área una vez creado un nuevo programa; también se puede encontrar en “Básico”) y el LED conectado a P1 se encenderá. El resultado final será el siguiente:



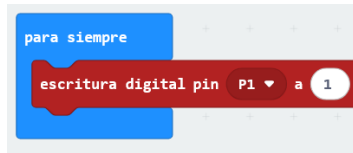
Tarea 2: LED parpadeante

Objetivo: Ahora que sabemos cómo encender un LED, daremos un paso más para aprender a hacer que parpadee.

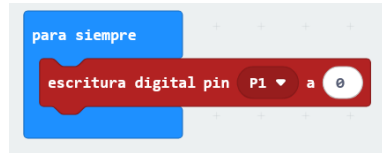
PASO 1: Inicia un nuevo proyecto. Si has olvidado cómo hacerlo, siempre puedes volver al PASO 3 del capítulo 1.

PASO 2: Configura el estado del pin del micro:bit para encender o apagar el LED. LED en este experimento pertenecen a la salida digital. Haz clic en “Avanzado” -> “Pines” y configura el “pin de escritura digital p1” en (0). El valor del pin solo puede ser 0 o 1. Cuando es 1, la luz permanece encendida. Cuando es 0, la luz permanece apagada.

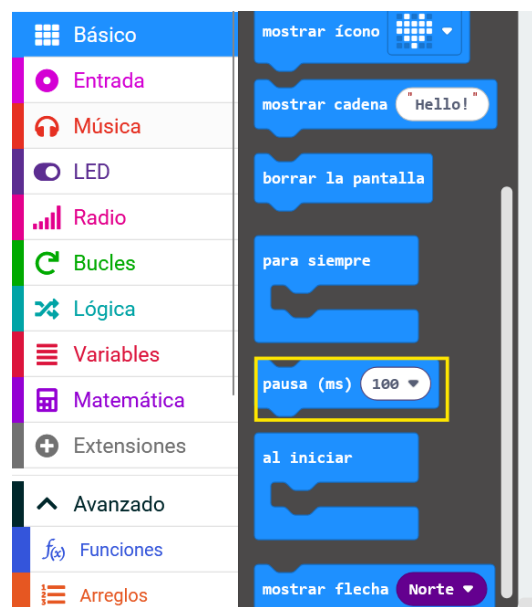
Establece el valor de pin P1 en alto (1) y el LED se encenderá



Establece el valor de pin P1 en bajo (0) y el LED se apagará



PASO 3: También tendrás que mantener el LED externo encendido durante 1 segundo y apagado durante otro segundo, en secuencia. Para ello, ve a “Basic” y selecciona “Pausa (ms) (100)”. La función “Pausa” mantiene el mismo estado durante un periodo de tiempo determinado.



PASO 4: Coloca el bloque “Pausa” en un bucle “infinito” y ajusta los valores.

Si combinamos todos los bloques de funciones mencionados anteriormente, obtendremos el programa final que se muestra a continuación. En esta imagen: configura el “pin P1” en 1 (LED encendido) y “pausa (ms)” en 1000; el pin P1 en 0 (apagado) y “pausa (ms)” en 1000; y el programa dentro del bucle “para siempre” se ejecutará repetidamente en secuencia.



Ejercicio

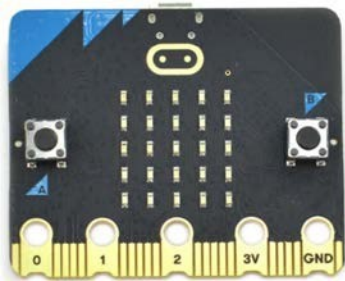
Es posible que hayas oído la señal de emergencia SOS ¿Por qué no invitas a tus compañeros de clase a aprender y a intentar programarla?

Proyecto 3: Luz de notificación

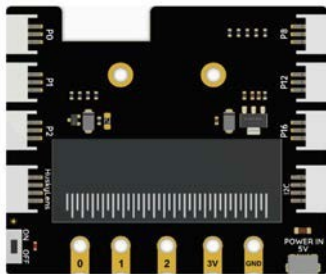
Ahora que ya has aprendido a encender un LED externo a través del micro:bit, vamos a utilizar un pulsador externo para controlar el LED. Además, aprenderemos a ajustar el brillo de la luz mediante un potenciómetro.

Lista de componentes

1 × micro:bit



1 × placa de expansión Boson



1 × módulo LED



1 × módulo pulsador



1 × módulo de mando giratorio (potenciometro)



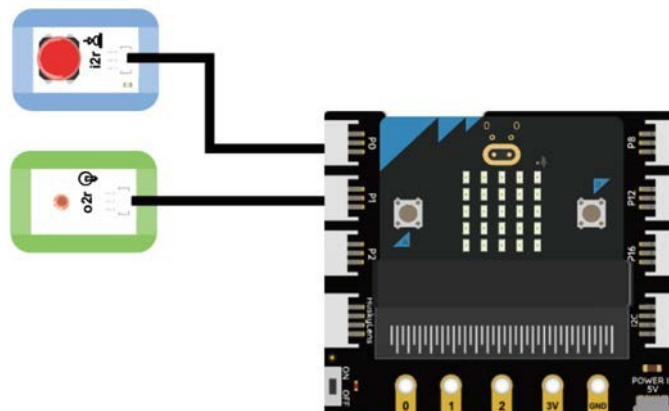
1 × cable USB



Conexión

Conecta el módulo de pulsador (módulo de botón) a P0

Conecta el módulo de LED a P1



Programa

Tarea 1: lámpara controlada por botón

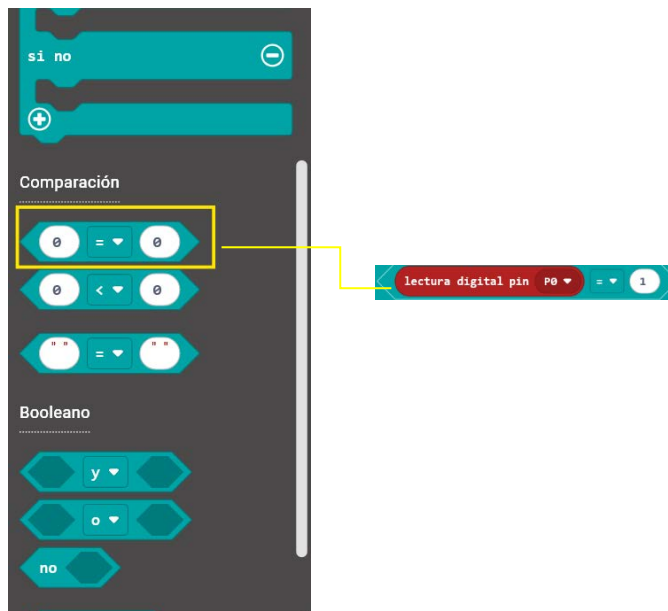
Objetivo: cuando se pulse el botón, el módulo LED se encienda. Cuando se suelte el botón, el LED se apague.

PASO 1: Conecta el módulo de botón a P0. Al leer el valor de P0, el micro:bit obtendrá el estado del botón.

Para vincular el botón con el LED, podemos escribir un programa sencillo: cuando se pulsa el botón, el micro:bit recibe un "1", por lo que estableceremos el valor del LED (pin 1) en 1 y el LED se encenderá. Al soltarlo, estableceremos el valor del pin en 0 y el LED se apagará. Este módulo de pulsador es un módulo de entrada digital y la función "lectura digital" se encuentra en "Avanzado" -> "Pines".



PASO 2: El operador de condición "=" en "Lógica" se utiliza para determinar si el botón está pulsado o no. Si el estado del botón es igual a 1, sabremos que el botón está pulsado, y viceversa.

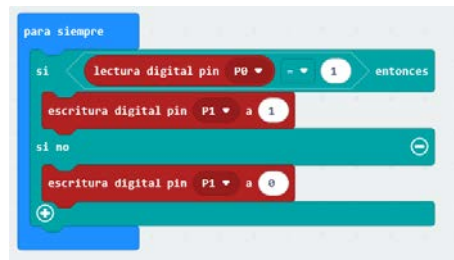


PASO 3: Si se pulsa el botón, el LED externo se enciende; de lo contrario, permanece apagado. Utilizaremos otro operador condicional, la función "si entonces". La función también se encuentra en "Lógica"



PASO 4: Al combinar todas las funciones mencionadas anteriormente, obtendremos el siguiente programa

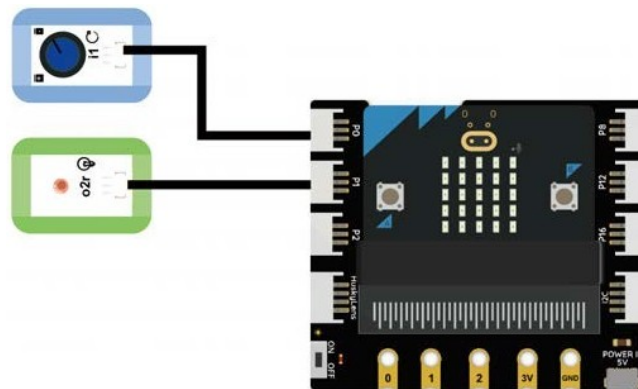
Lo que hace el programa es detectar el estado del botón y encender el LED cuando se pulsa el botón.



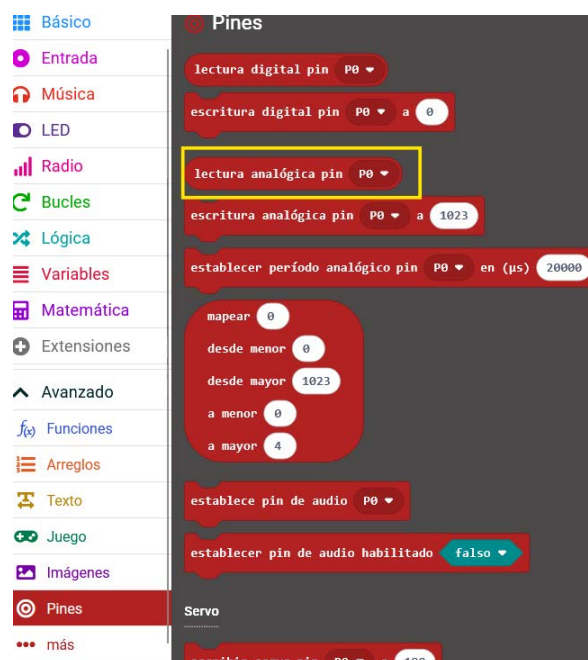
Tarea 2: Control del LED con el mando giratorio

Objetivo: en esta parte, no solo controlaremos el encendido y apagado del LED, sino que también utilizaremos el mando giratorio para ajustar su intensidad.

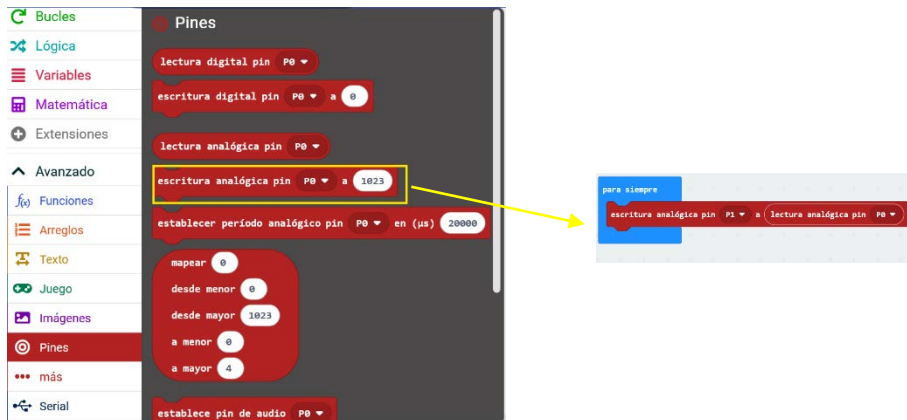
PASO 1: Conecta el circuito como se muestra a continuación. El módulo de pulsador utilizado en la tarea 1 se sustituye ahora por el módulo de potenciómetro.



PASO 2: El brillo del LED está vinculado al valor de entrada del mando giratorio. El mando giratorio es un módulo de entrada analógica, lo que significa que genera una señal en un rango de 0 a 1023, con la que se controla el brillo del LED de forma correspondiente. La función "lectura analógica" se encuentra en "Pines". Además, recuerda configurar el pin en P0.



PASO 3: El LED ahora se utiliza ahora como módulo de salida analógica. Del mismo modo, la señal de salida analógica también oscila entre 0 y 1023. La función “Escritura analógica” se encuentra en “Pins”.



PASO 4: Ahora, lo único que necesitamos es vincular la entrada directamente a la salida para que el LED quede bajo el control del mando. Coloca la función “lectura analógica” dentro de la función “escritura analógica” y, a continuación, coloca todas ellas dentro del bucle “para siempre”, y ya está. El programa final se muestra a continuación.



Ejercicio

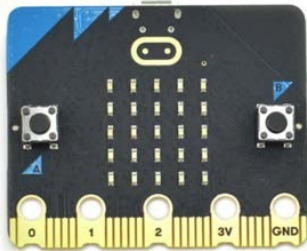
¡Haz que tu LED se parezca más a una lámpara de verdad!

Proyecto 4: Ventilador eléctrico

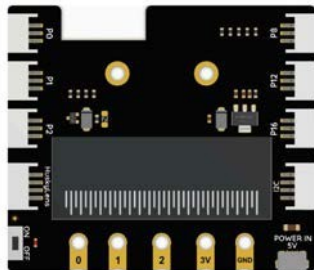
En esta parte, aprenderemos a construir un ventilador eléctrico que funcione a diferentes velocidades. Antes de empezar, repasaremos rápidamente cómo encender y apagar el ventilador con el módulo de pulsador. A continuación, aprenderemos a controlar la velocidad del ventilador con el módulo de mando giratorio.

Lista de componentes

1 × micro:bit



1 × placa de expansión Boson



1 × módulo de ventilador



1 × módulo pulsador



1 × módulo de mando giratorio (potenciómetro)

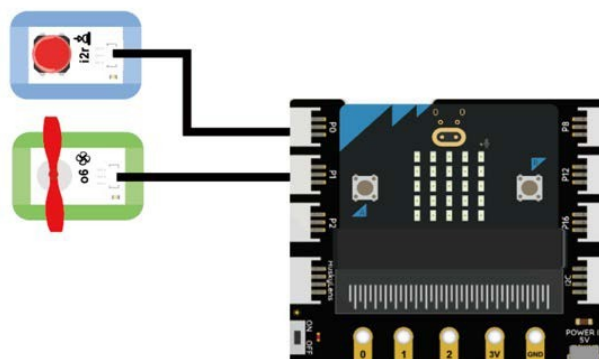


1 × cable USB



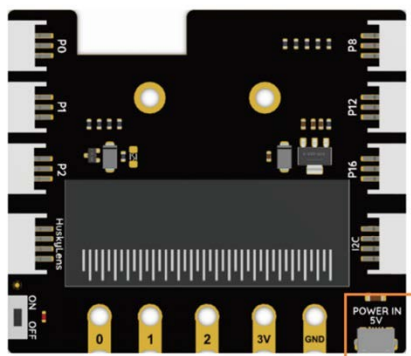
Conexión

Conecta el módulo pulsador (módulo potenciómetro en la segunda parte) al P0. Conecta el módulo del ventilador al P1.



! El ventilador eléctrico “se come” la energía ¡

Al conectar módulos Boson, hay que tener en cuenta que, debido al consumo de energía de los módulos, como los ventiladores o los servos, será necesaria una fuente de alimentación externa. Por lo tanto, primero utilizaremos el cable USB para programar el micro:bit. Una vez terminado, cambiaremos el cable USB por la entrada de alimentación externa de la placa de expansión.



Programa

Tarea 1: Controlar el ventilador con el módulo pulsador

Objetivo: cuando se pulse el botón, el ventilador comience a funcionar; cuando se suelte, el ventilador se detenga.

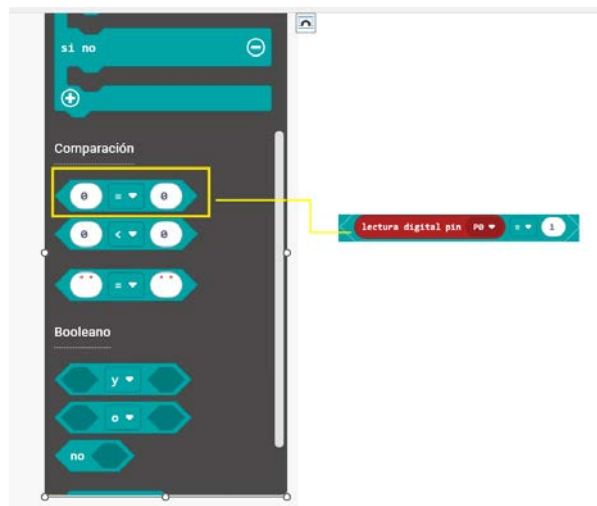
PASO 1: El módulo pulsador se conecta a P0, lo que significa que tendremos que utilizar la función de “lectura digital” en P0 para detectar primero el estado del botón.

El módulo del ventilador se conecta a P1. Del mismo modo, para controlar el ventilador, utilizaremos la función “escritura digital” en P1 para alternar entre las posiciones ON y OFF.

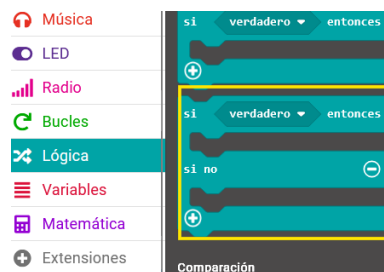
Las funciones “lectura digital” y “escritura digital” se encuentran en la sección “Pines”.



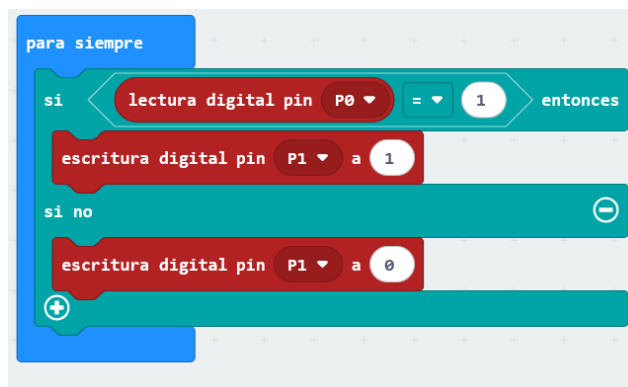
PASO 2: El operador condicional “=” en “Lógica” se utiliza para detectar si se ha pulsado el botón o no. Cuando se pulsa el botón, la función “lectura digital” devuelve “1”, lo que cumple la condición de “1” que hemos preestablecido al principio, por lo que se ejecutará el programa que se encuentra debajo de la instrucción. Sin embargo, cuando devuelve “0”, se omite el programa que se encuentra debajo de la instrucción.



PASO 3: Si se pulsa el botón, el ventilador se encenderá; de lo contrario, permanecerá apagado. La función “si entonces” de la sección “lógica” nos ayudará a tomar la decisión en función del estado del botón.



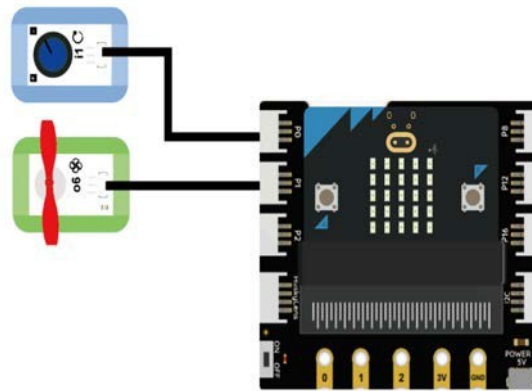
PASO 4: Al combinar todos los bloques de función mencionados anteriormente, deberíamos obtener un programa con la siguiente funcionalidad: cuando se pulsa el botón, el ventilador se enciende; cuando se suelta, el ventilador se apaga.



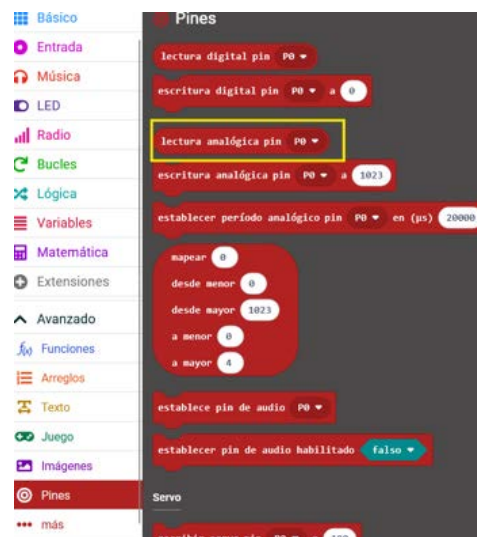
Tarea 2: Ajustar la velocidad del ventilador

Objetivo: En esta tarea, aprenderemos a utilizar un módulo de potenciómetro para controlar la velocidad del ventilador. Cuanto mayor sea el valor de entrada analógica que alcance el potenciómetro, mayor será la velocidad a la que funcione el ventilador, y viceversa. La velocidad del ventilador cambia continuamente a medida que gira el potenciómetro.

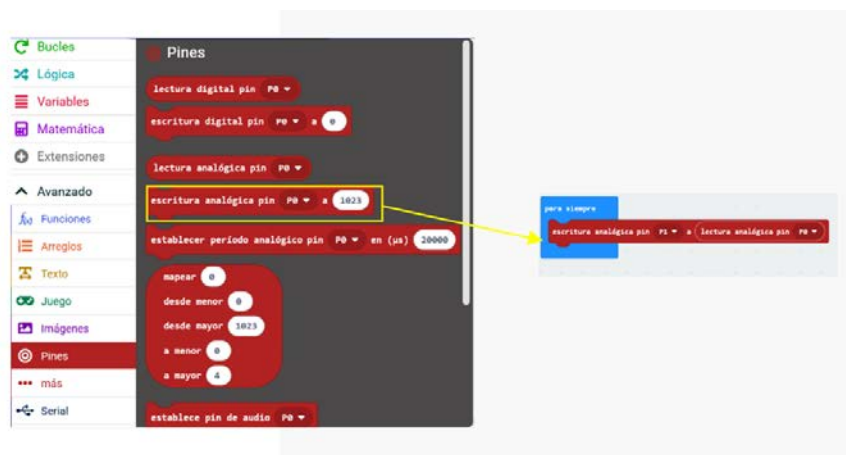
PASO 1: Conecta el circuito como se muestra a continuación. El módulo de botón utilizado en la tarea 1 debe sustituirse por un módulo de potenciómetro.



PASO 2: Volveremos a utilizar aquí el módulo ventilador. Para que la velocidad del ventilador se controle mediante el potenciómetro, debemos establecer una relación entre la velocidad y el valor de entrada del potenciómetro. El potenciómetro es un módulo de entrada analógica, lo que significa que genera una señal que oscila entre 0 y 1023, con la que se controla la velocidad del ventilador. Para leer la señal, conecta el módulo de potenciómetro a P0 y utiliza la función “lectura analógica” en “Pines”.



PASO 3: El módulo del ventilador se utiliza ahora como módulo de salida analógica, y su valor también oscila entre 0 y 1023. La función “escritura analógica” se encuentra en “Pines”



PASO 4: Introduce la función “lectura analógica” en “escritura analógica” y, a continuación, colócalas todas dentro del bucle “para siempre”. El programa final es el siguiente:



Capítulo 5: Un paso más allá

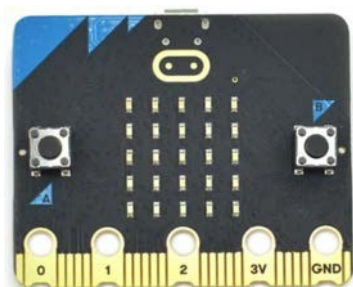
Anteriormente hemos aprendido a utilizar el módulo pulsador y el módulo de mandos giratorios para controlar un ventilador eléctrico. Pero aún queda mucho por hacer; quizá estés deseando explorar más a fondo el micro:bit. En este capítulo, se presentarán más sensores para que tu dispositivo sea más inteligente y funcional.

Proyecto 1: Vela electrónica

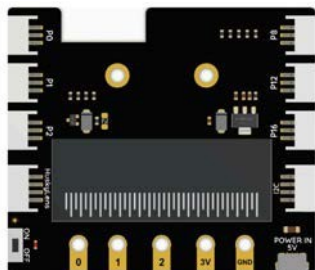
En este proyecto, vamos a crear un dispositivo que simule una vela para reducir las emisiones de carbono. Además, para que se parezca más a una de verdad, la llama debería poder apagarse soplando aire sobre ella. ¡Manos a la obra!

Lista de componentes

1 × micro:bit



1 × placa de expansión Boson



1 × módulo LED



1 × módulo de sensor de sonido

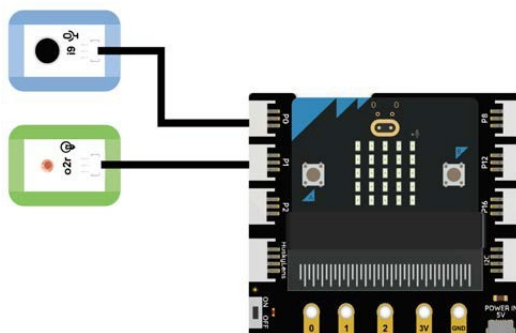


1 × cable USB



Conexión

Conecta el sensor de sonido a P0;
Conecta módulo LED a P1.



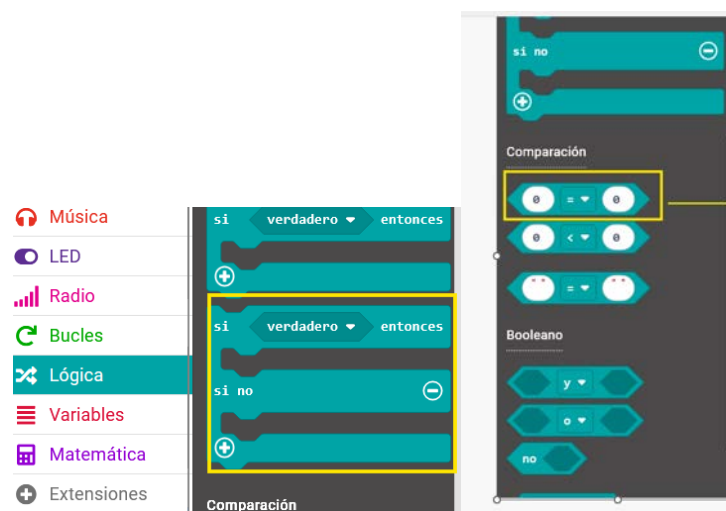
Programa

Para construir una vela electrónica, primero debemos saber cómo funciona el sensor de sonido. Podemos conectar el sensor de sonido al micro:bit y comprobar la intensidad del sonido observando el número que se muestra en el panel de LED.

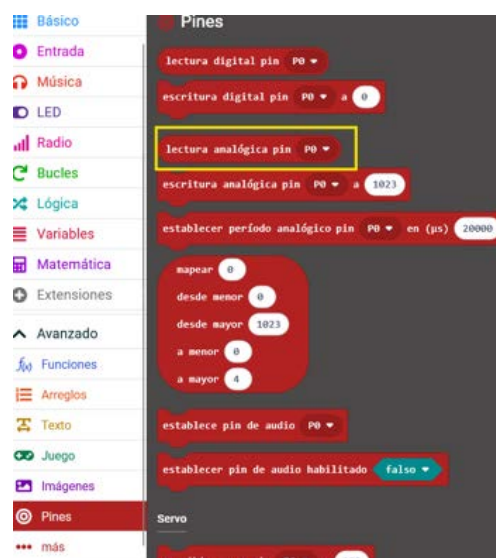
Tarea 1: Medir el volumen

Objetivo: Cuando el valor analógico del sensor de sonido sea inferior a 50, el LED integrado mostrará el número “1”, lo que indica que el volumen es bajo; cuando sea superior a 50, el LED integrado mostrará el número “2”, lo que indica que el volumen es alto.

PASO 1: Dejaremos a un lado el módulo LED por un momento y nos centraremos en la parte lógica. En esta parte se aplicarán la función “si entonces” y el operador “=”. Ambos deben colocarse dentro de un bucle “para siempre”. La función “si entonces” se puede utilizar para determinar qué número debe mostrarse. Esto significa que, si se cumple la condición lógica del lado derecho, se ejecutará la instrucción situada a la derecha de “entonces”, mientras que se ejecutará la instrucción situada a la derecha de “si no” si no se cumple dicha condición.



PASO 2: El sensor de sonido está conectado a P0, lo que significa que utilizaremos la función “lectura analógica” para obtener la lectura en P0. Como se ha explicado anteriormente, la señal analógica oscila entre 0 y 1023.



PASO 3: Ahora, tendremos que hacer que el programa determine el nivel de la voz. El operador “<” de la sección “Lógica” se encarga de ello. Como se muestra en la imagen siguiente, cuando el valor analógico del sensor de sonido es inferior a 50, el LED integrado muestra el número “1”, lo que indica que el volumen es bajo; cuando es superior a 50, el LED integrado muestra el número “2”, lo que indica que el volumen es alto.



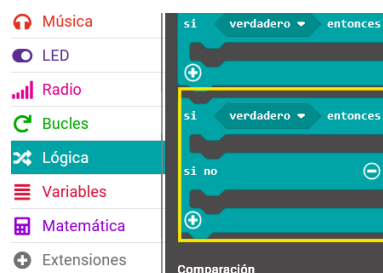
Haz algo de ruido alrededor del sensor, o simplemente sopla un poco de aire para comprobar si el micro:bit es capaz de mostrar números diferentes en función del volumen.

Además, si el sensor es demasiado sensible, es posible que tengas que cambiar el “50” que aparece después del operador “<”. Lo mismo se aplica a la inversa si el sensor no es lo suficientemente sensible.

Tarea 2: vela electrónica

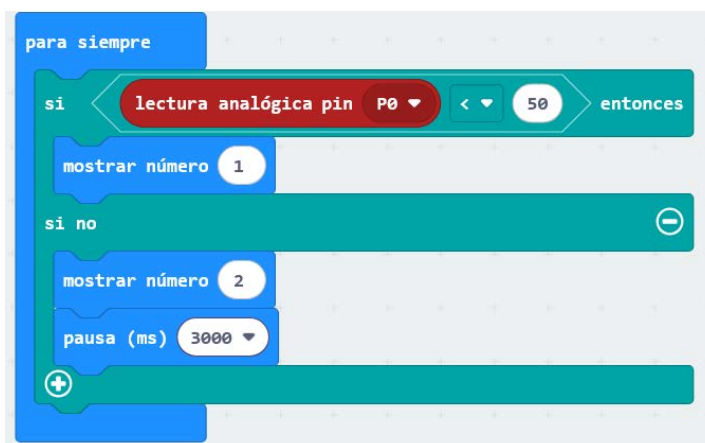
Objetivo: El micro:bit puede medir el volumen mediante el sensor de sonido; basándonos en ello, queremos utilizarlo para controlar la vela. Esto es lo que vamos a programar: en cuanto soplamos aire hacia el sensor, la vela se apaga durante un segundo.

PASO 1: Volveremos a utilizar la función “si entonces” en nuestro programa. Sin embargo, en lugar de controlar un panel de LED, solo tenemos que controlar el LED conectado a P2. La lógica es muy sencilla: cuando el volumen supere el valor preestablecido, la luz se enciende, y viceversa. Como queremos que la vela siga encendida, el programa debe colocarse dentro del bucle “para siempre”.



PASO 2: Sin embargo, ¡eso todavía no es suficiente! Cuando soplamos aire hacia el sensor, el volumen solo superará el valor preestablecido durante un tiempo muy breve. A veces ocurre tan rápido que no podremos percibir el cambio. Para resolver este problema, tendremos que añadir una función de “Pausa” para prolongar la duración. Supongamos que la duración se ha fijado en 3000 (3000 ms = 3 segundos); una vez que soplamos la vela, la luz se apagará durante 3 segundos y, a continuación, volverá al principio del bucle “infinito” y se encenderá de nuevo.

Así es como queda el programa.



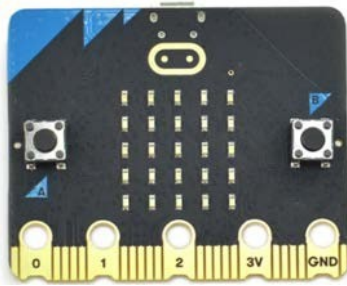
Proyecto 2: Puerta automática

Las puertas automáticas son muy comunes en nuestra vida cotidiana. Cuando estamos a punto de entrar en una tienda, la puerta se abre automáticamente y, a veces, acompañada del sonido “¡Bienvenido~!”.

En esta lección, vamos a aprender a construir una puerta automática utilizando un sensor de movimiento, que detecta si se acerca alguien, y un servo para simular el movimiento de apertura de la puerta. Ahora, ¡Empecemos!

Lista de componentes

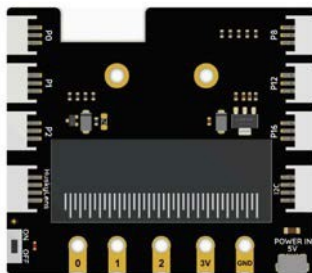
1 × micro:bit



1 × módulo de sensor de movimiento



1 × placa de expansión Boson



1 × módulo de servo



1 × módulo de pulsador



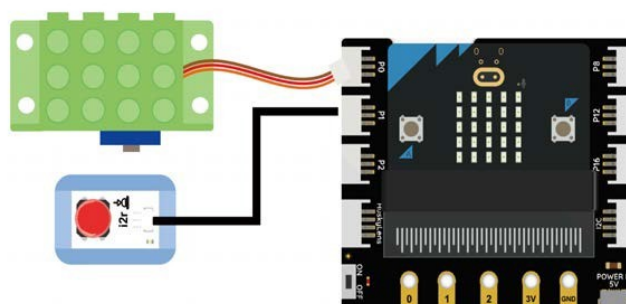
1 × cable USB



Conexión

Conecta el servo a P0

Conecta el módulo pulsador (sensor de movimiento en la segunda parte) a P1

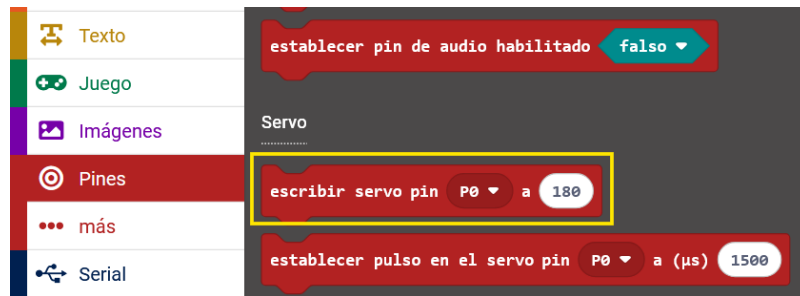


Programa

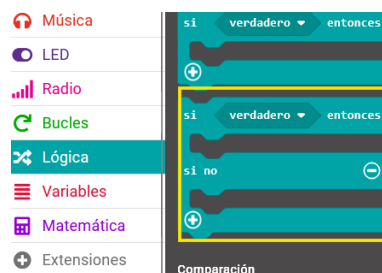
Tarea 1: Utiliza el módulo pulsador para controlar el servo

Objetivo: Cuando se pulse el botón, el servo gire hasta los 100°, y el panel LED del micro:bit muestre una “O”; al soltarlo, el servo se quede en 0°, y el panel LED del micro:bit muestre una “X”.

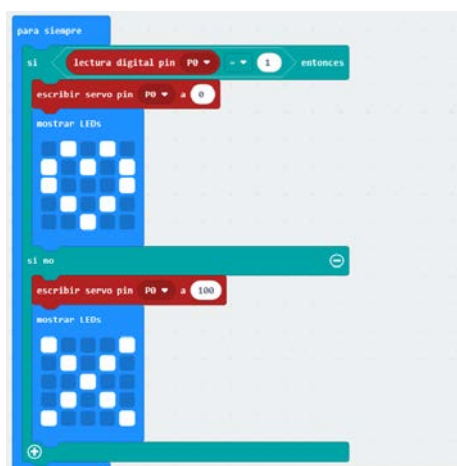
PASO 1: Se utiliza la función “servo write” de la sección “Pines” para controlar el servo. Dado que el servo está conectado a P0, dejaremos “P0” tal cual. El número “180” es el ángulo que queremos que alcance el servo. Al cambiar este valor, podemos configurar el servo para que gire entre 0° y 180°.



PASO 2: Ahora, utiliza la función “si entonces” de para determinar la posición del servo en función del estado del botón. Cuando se pulsa el botón, el servo se coloca a 100°, y en caso contrario, permanece a 0°. Además, no olvides añadir las letras “O” y “X” para indicar el estado del botón.



PASO 3: Al juntarlo todo, obtendremos el programa que se muestra a continuación.

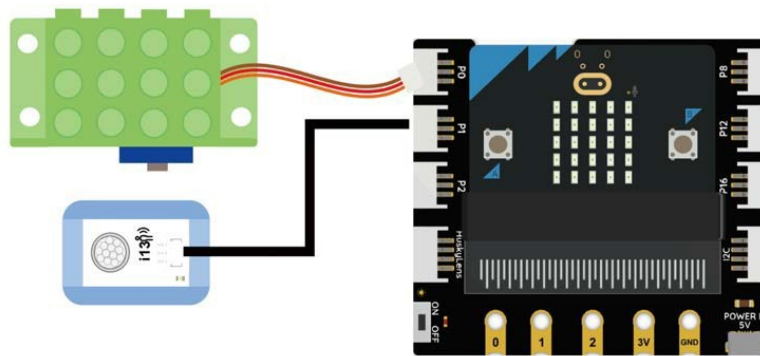


Hay que tener en cuenta que el servo tiene un consumo de energía relativamente elevado, por lo que tendremos que conectarlo a la entrada de alimentación a través del puerto USB externo.

Tarea 2: Puerta con control de movimiento

Objetivo: vamos a sustituir el botón por un sensor de movimiento para controlar el servo. Cuando alguien se acerque, la puerta se abrirá. De lo contrario, permanecerá cerrada.

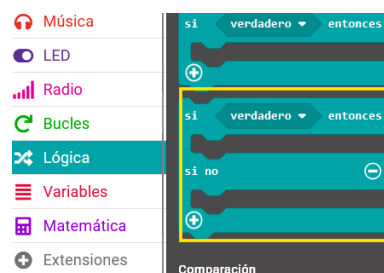
PASO 1: Sustituye el módulo del botón por el sensor de movimiento.



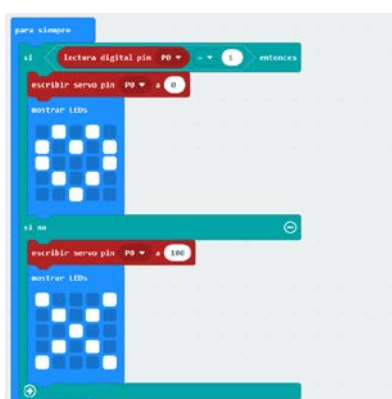
PASO 2: El sensor de movimiento está conectado a P1. De este modo, podemos leer el valor del sensor de movimiento. La función utilizada en esta parte es: "Pines" => "lectura digital". Recuerda sustituir el valor predeterminado P0 por P1.



PASO 3: Cuando el sensor de movimiento detecta a alguien, el servo gira hasta los 100°, y el micro:bit muestra "O"; en caso contrario, el servo gira hasta los 0° y el micro:bit muestra "X". El bloque de funciones utilizado en este paso es "si entonces".



PASO 4: Al combinar todas las funciones mencionadas anteriormente, obtendremos el programa final tal y como se muestra a continuación:

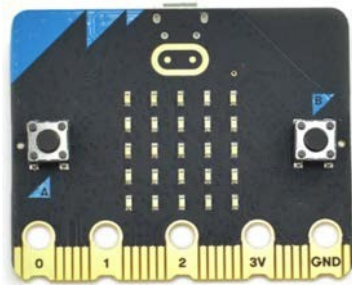


Proyecto 3: Caja de música

Las cajas de música, con sus llamativas formas, siempre han sido muy populares en las tiendas de regalos. Seguro que te han cautivado su aspecto ornamentado y sus ingeniosos diseños. En esta lección, te ayudaremos a construir una caja de música inteligente. Vamos a empezar.

Lista de componentes

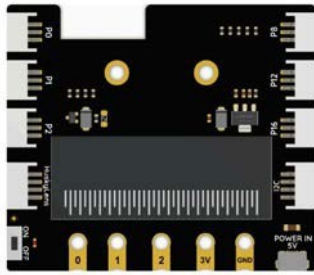
1 × micro:bit



1 × auriculares



1 × placa de expansión Boson



1 × módulo de sensor de movimiento

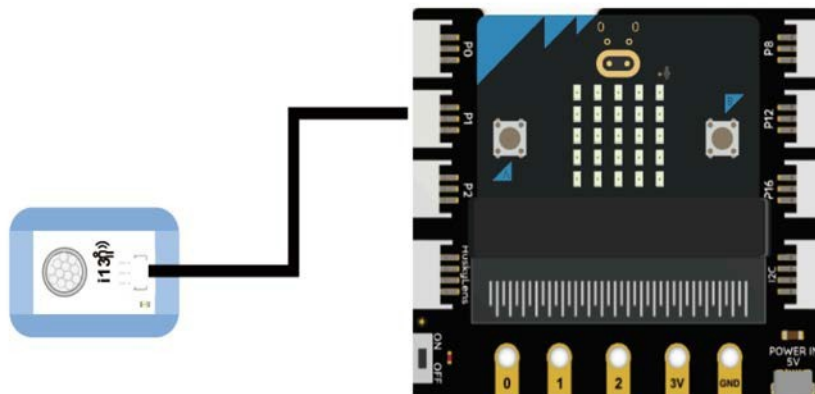


1 × cable USB



Conexión

Conecta el micro:bit al ordenador y enchufa los auriculares en la toma. Conecta el módulo sensor de movimiento a P1.



Programa

Tarea 1: Crear una caja electrónica de música

Objetivo: En esta parte, aprenderemos a hacer que el micro:bit reproduzca una pieza musical.

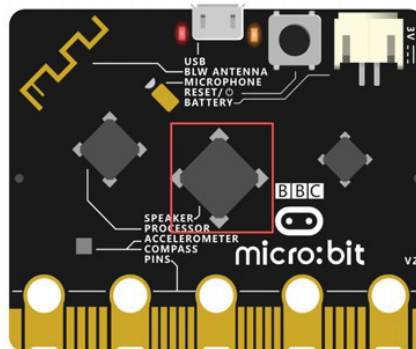
PASO 1: En este paso, utilizaremos la función “Música”. Arrastra la función “Reproducir la melodía “dadadum” una vez” desde “Música” al bucle “para siempre”. También puedes elegir otras canciones disponibles en la lista.



PASO 2: Dado que el “dadadum” dura 4 tiempos, para esta parte se necesita la función “silencio (ms)” de “Música”. El programa final es el siguiente:



El micro:bit viene con un zumbador integrado.



Tarea 2: Caja de música inteligente

Objetivo: En esta tarea construiremos una caja de música inteligente. Cuando alguien se acerque, el micro:bit reproducirá una pieza musical de tu elección; cuando no haya nadie cerca, permanecerá en silencio.

PASO 1: Edita el programa “Twinkle, Twinkle Little Star”.

En primer lugar, intentaremos editar la primera frase de “Twinkle, Twinkle Little Star”. Para editar la frecuencia vocal se necesitan diferentes tonos y notas

小星星

Tone 调号 1=C 1 1 5 5 6 6 5 — 4 4 3 3 2 2 1 音符 Notes

一闪一闪亮晶晶，满天都是小星星，

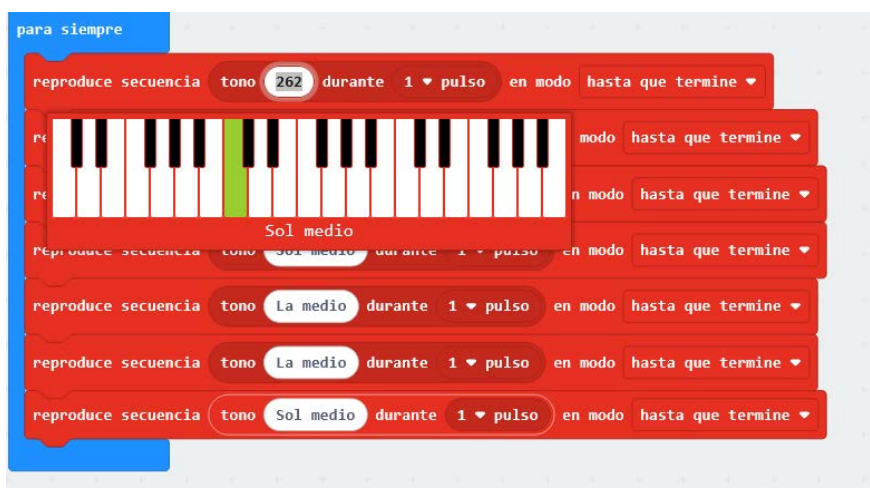
"Twinkle, Twinkle Little Star".

Tomemos como ejemplo el Do central; la frecuencia del sonido y sus notas correspondientes se muestran en la siguiente tabla:

Notas: 1 2 3 4 5 6 7

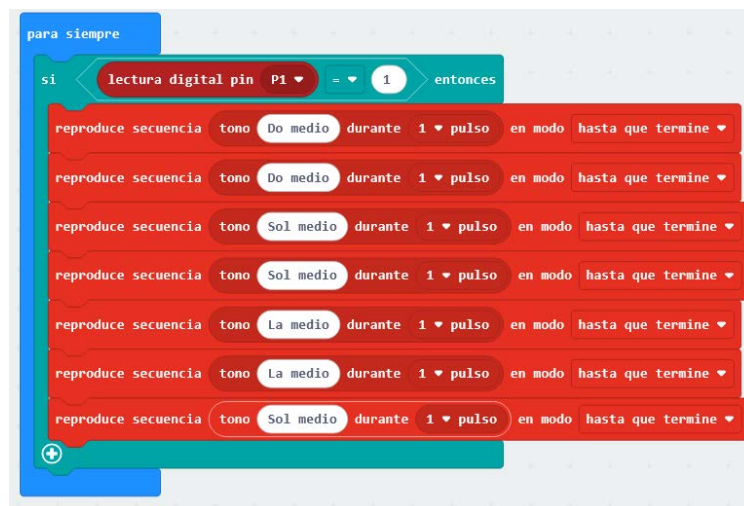
Hz	26	29	33	35	39	44	49
	2	4	0	0	3	1	5

En este paso se utilizará la función "reproducir el tono Do medio durante 1 compás" de la sección "Música". Aquí puedes configurar el tono y los compases.



Cuando se combinan todas las funciones mencionadas anteriormente dentro del bucle "para siempre", nuestro micro:bit es capaz de reproducir la primera frase de "Twinkle, Twinkle Little Star". Pero, ¿puede ser aún más inteligente? El módulo del sensor de movimiento volverá a ser nuestra clave. Al combinarlo con la función "si entonces", la caja de música inteligente cobrará vida.

PASO 2: Una vez conectado el sensor de movimiento y añadida la función "si entonces" al programa, al reunir todas las funciones mencionadas anteriormente, el programa final de la caja de música inteligente queda así:

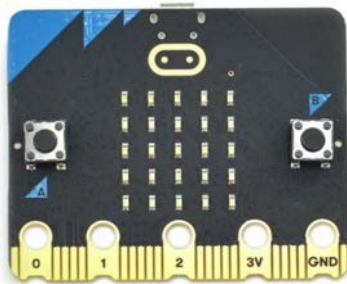


Proyecto 4: Tira de LED de colores

Ahora que ya has aprendido a construir una caja de música inteligente, quizá quieras seguir explorando las posibilidades del micro:bit. Las cajas de música de nuestra vida cotidiana suelen estar decoradas con luces de colores. Por eso, en este proyecto te enseñaremos a crear una tira de LED de colores para ayudarte a construir una caja de música única. ¡Únete a nosotros y adéntrate en el fantástico mundo de la luz!

Lista de componentes

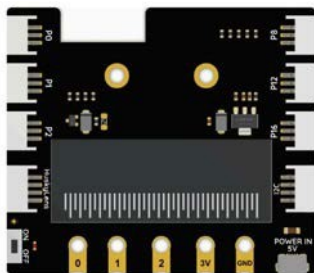
1 × micro:bit



1 × tira de LED RGB



1 × placa de expansión Boson



1 × módulo de sensor de sonido

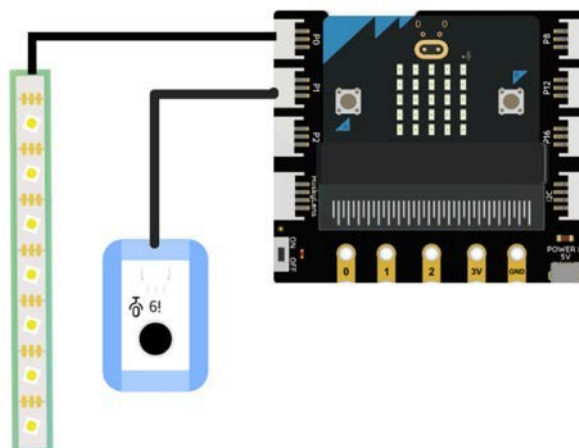


1 × cable USB



Conexión

Conecta la tira de LED RGB a P0.

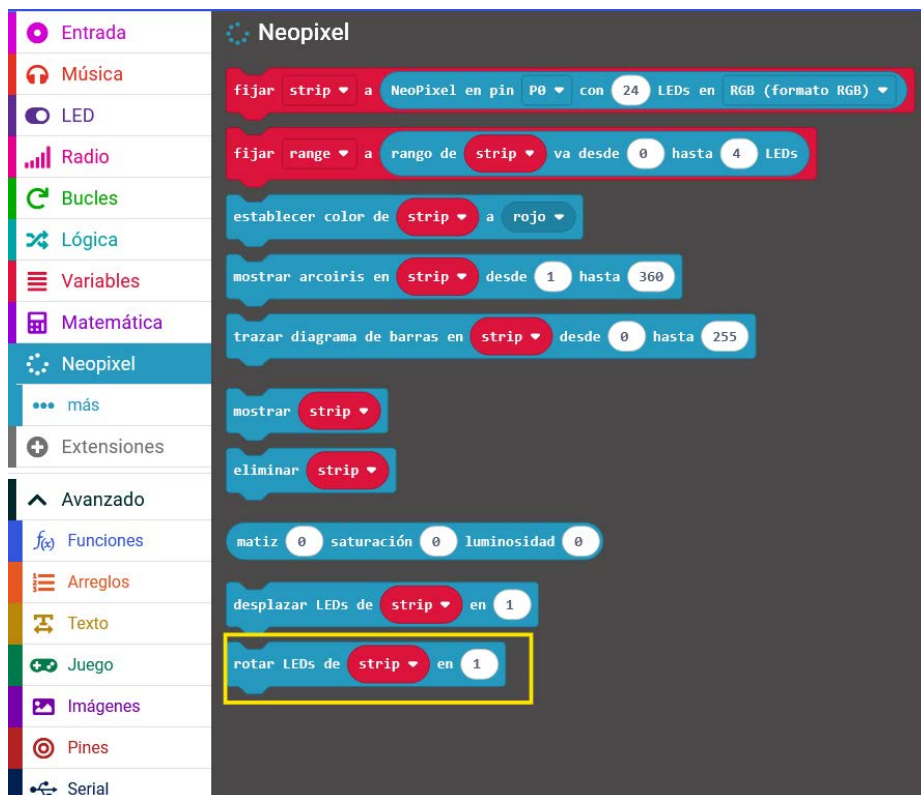
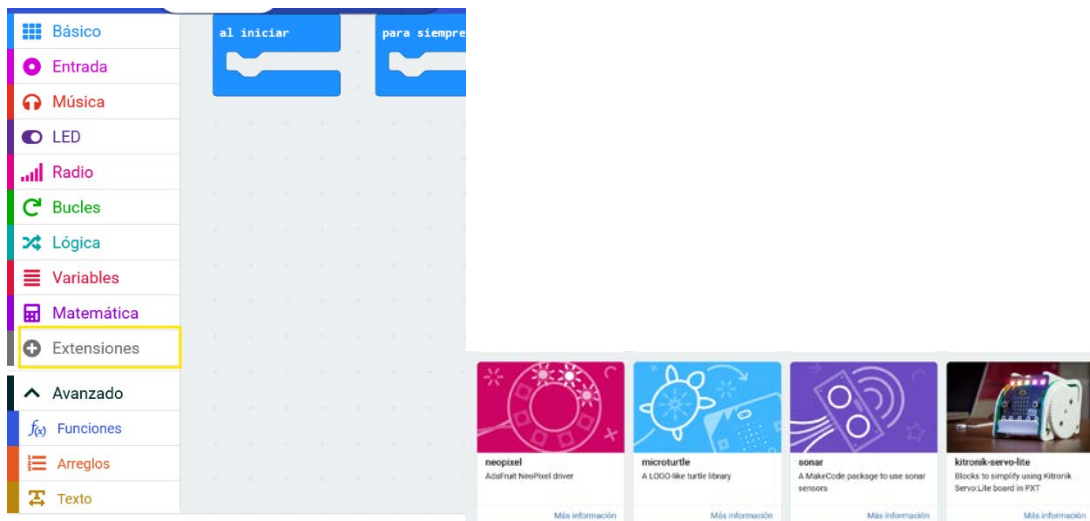


Programa

Tarea 1: Encender la tira LED RGB D

Objetivo: En esta parte, aprenderemos a utilizar el micro:bit para programar la tira de LED RGB.

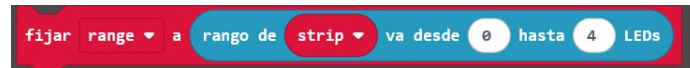
PASO 1: El bloque de funciones para controlar la tira de luces RGB no aparece en la caja de herramientas al iniciar un nuevo proyecto. Para añadir esta función, debemos importar manualmente “Neopixel” desde la parte inferior de la página. Haz clic en “Extensiones” para abrir el gestor de funciones de extensión y, a continuación, selecciona “Neopixel”. Neopixel aparecerá en la lista de funciones.



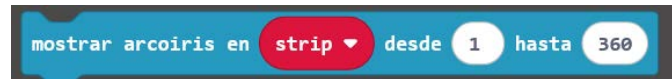
PASO 2: Para empezar a utilizar la función “ ” con Neopixel, primero configuraremos cuántos LED incluye la tira de luces y a qué pin está conectada. Tendremos que introducir estos dos parámetros en el bloque de funciones de la siguiente manera:



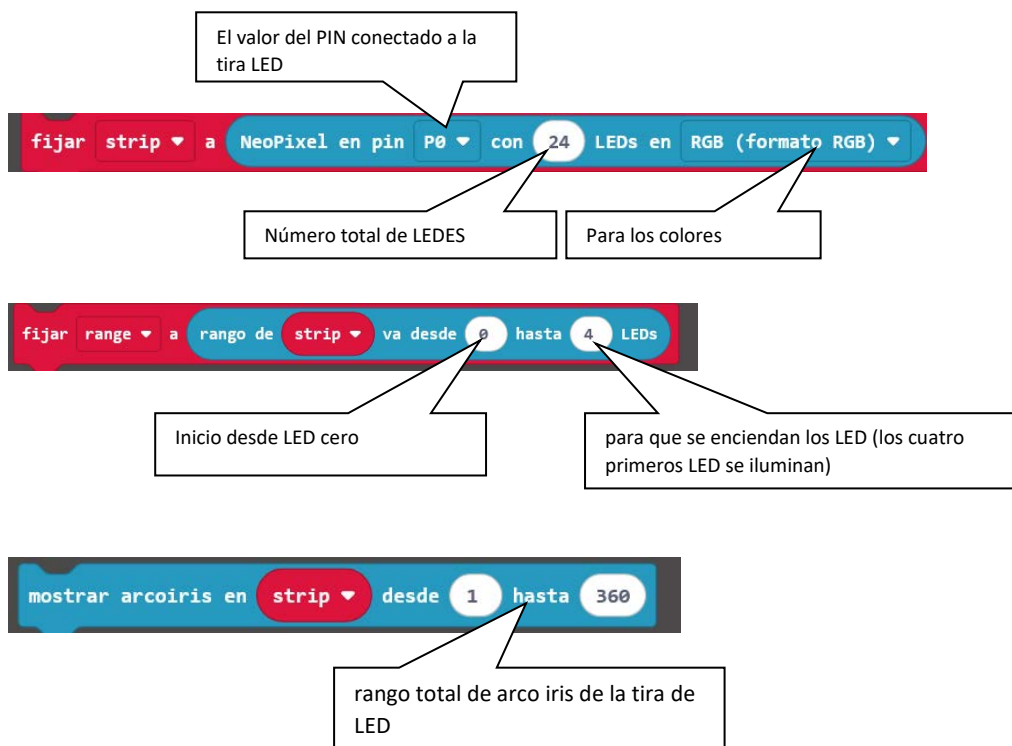
A continuación, debemos configurar cuántos LED (del número total de LED que incluye físicamente la tira) estarán controlados por el programa:



A continuación, utiliza la función “mostrar arco iris” para configurar el efecto de iluminación. Este efecto solo se aplicará a los LED que hayamos definido en el paso anterior. El parámetro dentro del bloque de funciones se utiliza para establecer el degradado en la escala de colores, el 1 representa el rojo, mientras que el 360 representa el azul; cuanto más cercanos estén estos dos números, menor será el cambio en el degradado:



El diagrama te ayudará a comprender en general el significado de cada parámetro: (intenta ajustar el parámetro y observa cómo cambia el efecto de luz)



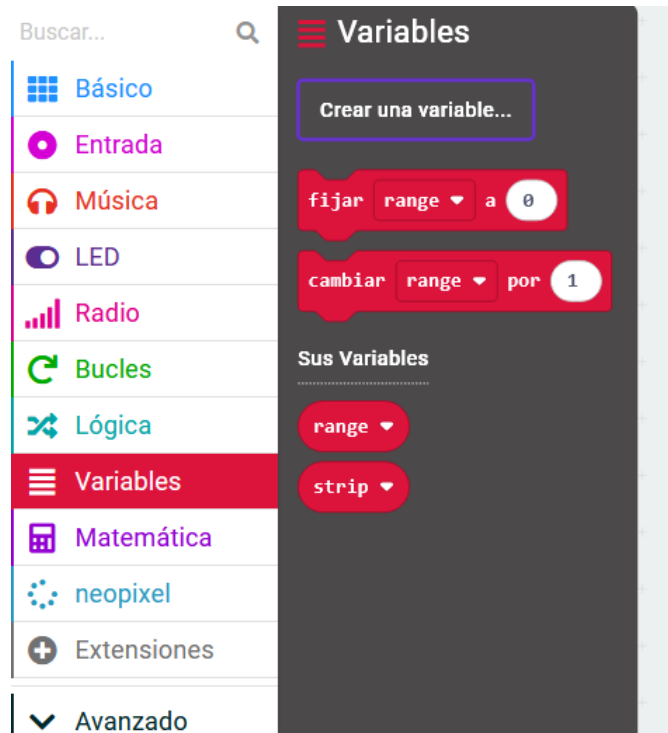
PASO 3: El pin P0 al que está conectada la tira cuenta con 7 LED. Para añadir un efecto de luz arcoíris a la tira, tendremos que establecer el rango de color entre 1 y 360 (el rojo cambia gradualmente a azul). A continuación, combina todos los bloques de función y colócalos dentro del bucle “para siempre”, y nuestro programa final quedará:



Tarea 2: Encender las luces del arcoíris

Objetivo: Ya hemos aprendido encender la tira de LED y a configurar sus efectos de luz. Pero, ¿podemos crear una animación de luz sencilla, por ejemplo, hacer que los LED se enciendan en secuencia?

PASO 1: La clave para crear una animación con la tira de LED es cambiar el parámetro del efecto de luz. Crearemos una variable que cambie con el tiempo y la utilizaremos para sustituir el parámetro constante preestablecido. Para configurar dicha variable, ve a: “Variables” => “Crear una variable” y establece el nombre de la variable (para que quede claro, hemos llamado a la variable “número de LED que se van a encender”). A continuación, haz clic en Aceptar.



Nombre de la nueva variable:

Aceptar

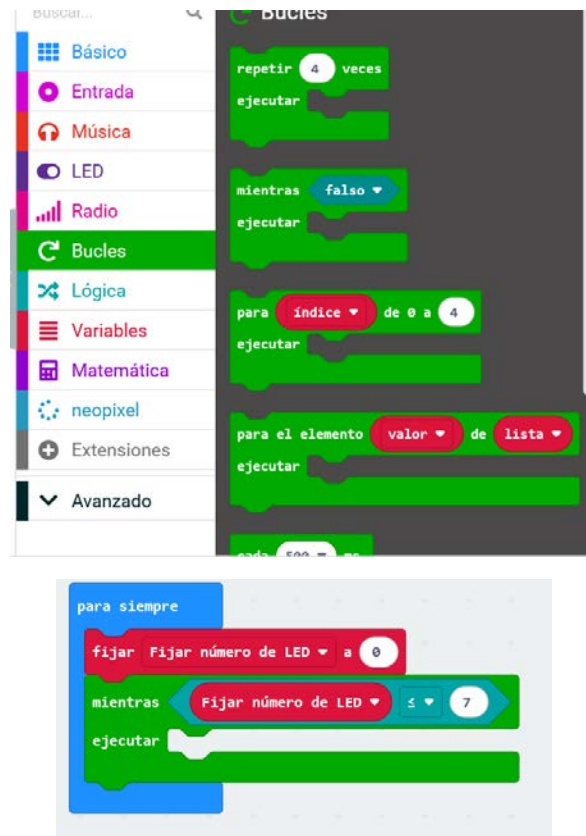
Nombre de la nueva variable:

fijar el número de LES a encender

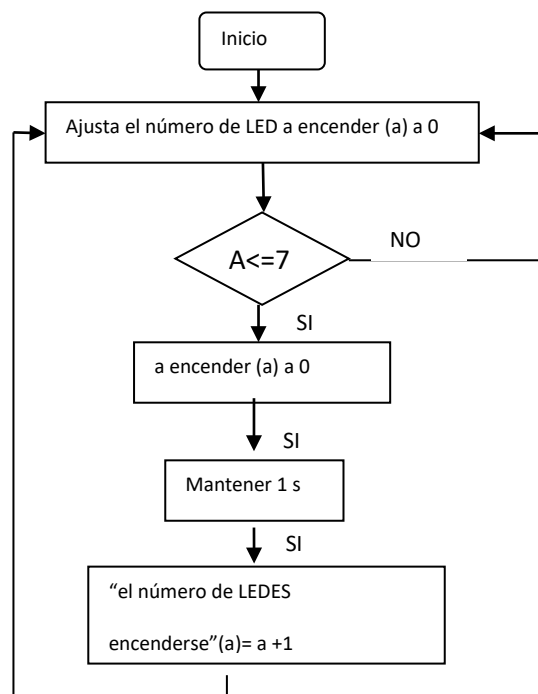
Aceptar

PASO 2: Todos los LED empiezan apagados, lo que significa que el valor inicial de la variable “el número de LED que se van a encender” debe establecerse en 0. A continuación, utilizaremos una función de bucle para que la variable aumente en 1 en cada iteración. Además, para asegurarnos de que el número total de LED nunca supere 7, tendremos que establecer la condición inicial utilizando la función “mientras”.

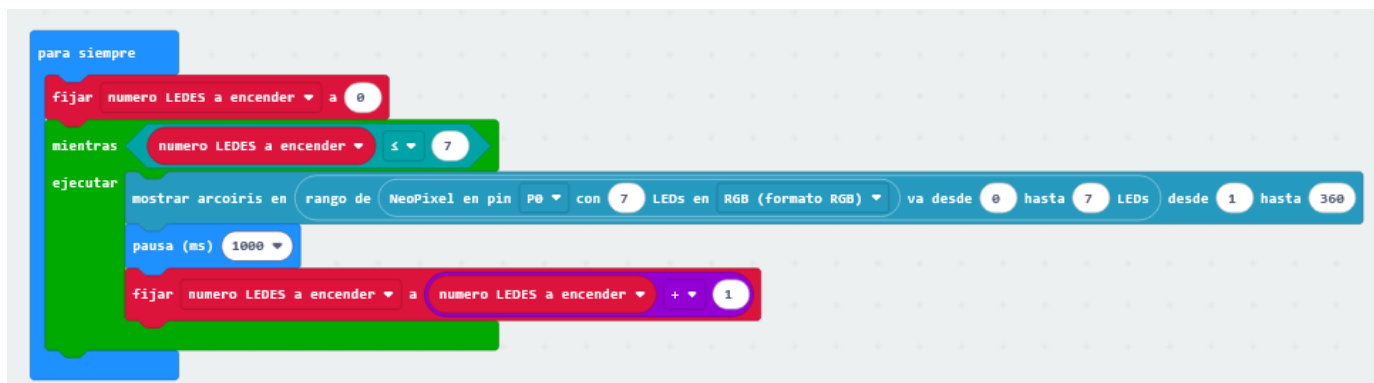
Utilizando el operador lógico “<” para establecer la condición, tendremos lo siguiente: cuando “el número de LED que se van a encender” sea menor que 7, el bucle “mientras” hará que los LED se enciendan uno a uno hasta que todos estén encendidos. Al combinar todas las funciones mencionadas anteriormente, el programa final queda así:



PASO 3: Ahora, lo único que nos queda por hacer es aumentar el parámetro en 1 en cada bucle. Sin embargo, no queremos que esto ocurra demasiado rápido, por lo que una función “pausa” puede ayudarnos a ralentizarlo. El diagrama de flujo que aparece a continuación puede resultarnos útil para entender cómo es el programa:



PASO 4: Para que los LED se iluminen uno a uno cada segundo, tendremos que introducir la variable “número de LED que se van a encender” dentro de la función “rango de LED desde” e incluir una “pausa de 1000 ms” en el bucle. Y esto es lo que tenemos.



Tarea 3: Tira de LED RGB controlada por un sensor de sonido

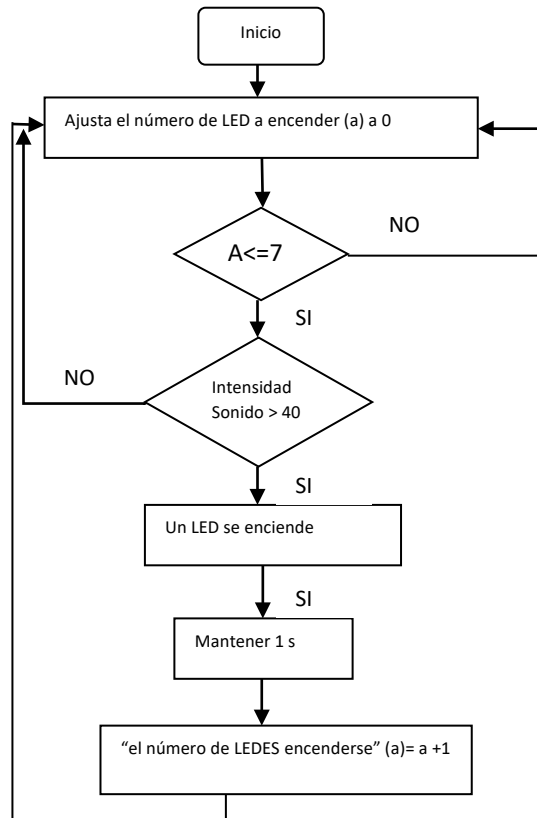
Objetivo: Lo que aprenderemos en esta parte es a utilizar un sensor de sonido para controlar la tira de LED: cuando se detecta sonido, se enciende 1 LED más en la tira.

PASO 1: En primer lugar, tenemos que configurar el sensor de sonido tal y como hicimos en los capítulos anteriores. Recuerda que, en nuestro caso, el sensor está conectado a P1.

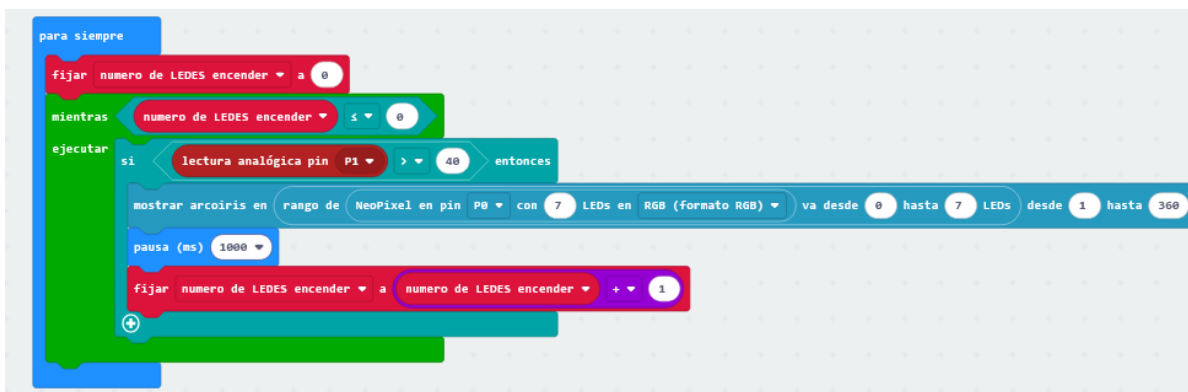


PASO 2: Cuando el valor supere 40 y el número total de LED encendidos sea inferior a 7, se encenderá un LED más. El sensor de sonido detectará el valor de forma continua.

El siguiente diagrama de flujo puede resultar útil para comprender el mecanismo:



PASO 3: Añade la condición “si la lectura analógica de P1 > 40” al programa que creamos en la parte anterior; el programa final quedará así:



Capítulo 6: Conviértete en un experto

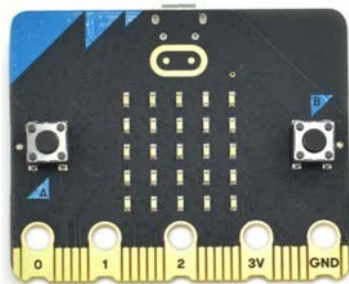
En los capítulos anteriores hemos creado un montón de proyectos divertidos con diferentes tipos de módulos. Es posible que ahora tengas un conocimiento más profundo del micro:bit. ¿Es eso suficiente para seguir explorando? La respuesta es, sin duda, ¡NO! No te vayas, vamos a dar un paso más en el mundo del micro:bit.

Proyecto 1: Estabilizador electrónico

¿Has oído hablar del acelerómetro? El acelerómetro mide la aceleración en tres direcciones y hoy en día es muy popular en todo tipo de dispositivos electrónicos. Por ejemplo, si giramos el teléfono o el portátil 90 grados, la pantalla también girará 90 grados automáticamente, lo que ahorra mucho tiempo a la hora de ajustar la orientación. En esta parte, ¡vamos a experimentar con el acelerómetro utilizando el micro:bit!

Lista de componentes

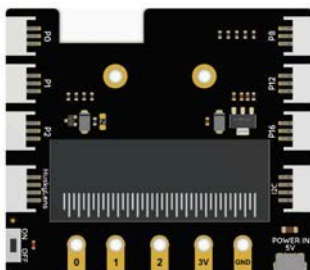
1 × micro:bit



1 × módulo servo



1 × placa de expansión Boson

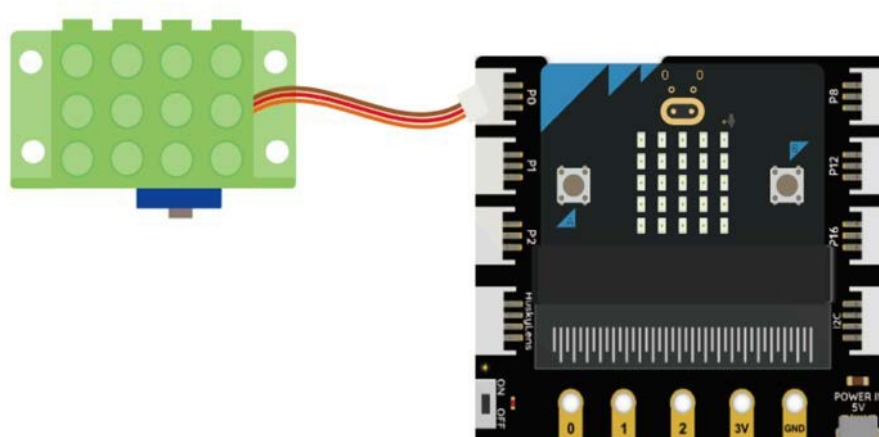


1 × cable USB



Conexión

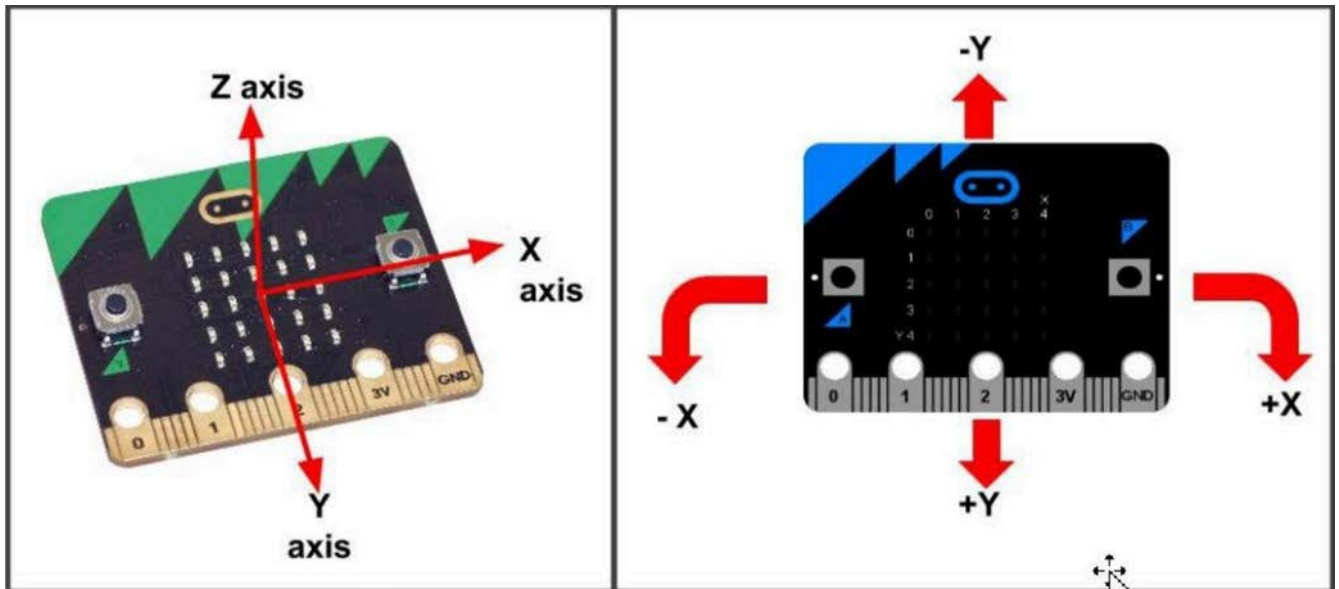
Conecta el servo a P0



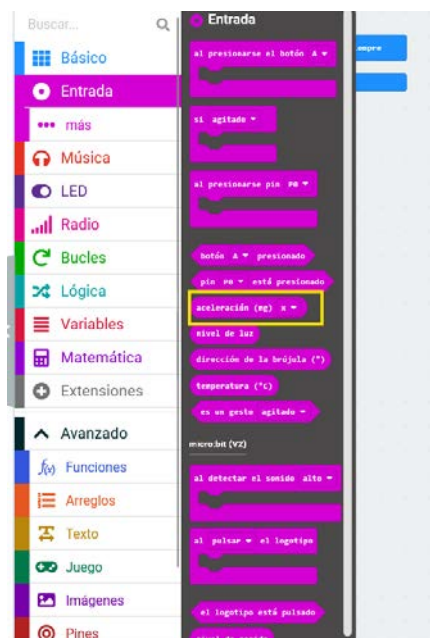
Programa

Objetivo: Fijar el servo en el micro:bit, de modo que el servo siga apuntando hacia arriba independientemente de cómo se incline la placa.

El micro:bit detecta el movimiento automáticamente a través del acelerómetro. La fuerza de la gravedad también es una forma de aceleración, y el micro:bit puede detectar la fuerza gravitatoria a lo largo de los ejes X, Y y Z, donde X corresponde a la dirección izquierda-derecha, Y a la dirección delante-detrás y Z es perpendicular a la placa y corresponde a la dirección arriba-abajo.



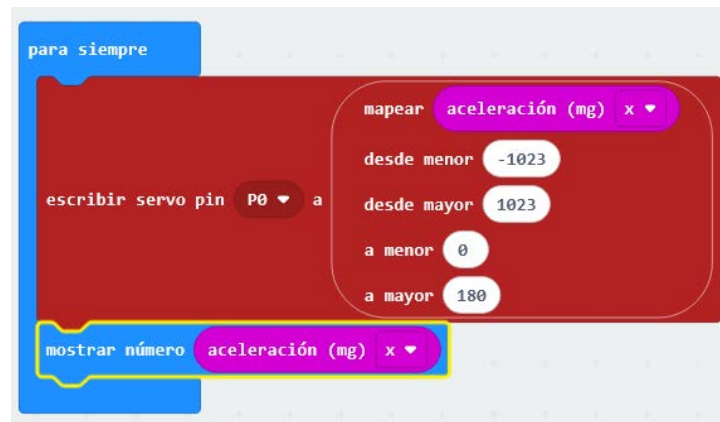
Los valores de los ejes X, Y y Z se pueden obtener mediante la función “acceleration” en la sección “Input”.



Cuando la placa principal se coloca en posición horizontal sobre la mesa, la fuerza gravitatoria actúa en dirección arriba-abajo, de modo que los valores de los ejes X e Y son cercanos a cero y el del eje Z ronda los 10000 (mm/s^2), lo que equivale aproximadamente a la aceleración gravitatoria.

Al inclinar la placa, los valores de X e Y cambian. Si se inclina hacia la izquierda, el valor de X es negativo; si se inclina hacia la derecha, es positivo.

PASO 1: Cuando la placa se inclina hacia la izquierda, el acelerómetro generará un valor de salida negativo en la dirección X. Giraremos el micro:bit hacia los 0 grados para compensar este ángulo y viceversa. Dado que el valor de salida del acelerómetro oscila entre -2048 y 2048, y en nuestro caso, el valor máximo que alcanzará es -1023 y 1023, podemos hacer que el rango de -1023 a 1023 se corresponda con el de 0 a 180 grados del servo mediante la función “mapear”.



PASO 2: Fija el servo a la placa de expansión con cinta adhesiva para que ambos permanezcan relativamente fijos. Ten en cuenta que la orientación del servo debe ser la misma que la del puerto USB, de modo que el eje del engranaje quede perpendicular al eje X y el ángulo de giro del servo sea complementario al grado de inclinación del micro:bit.

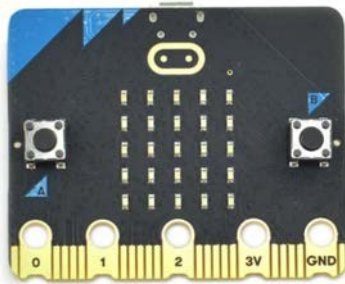
Ejercicios

Intenta construir una brújula utilizando la brújula electrónica integrada en el micro:bit.

Proyecto 2: Panel de DJ

Lista de componentes

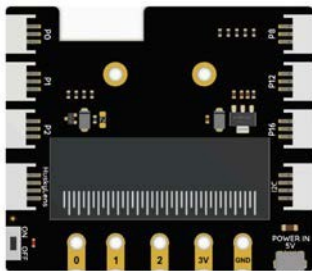
1 × micro:bit



1 × módulo de mando



1 × placa de expansión Boson



1 × cable USB



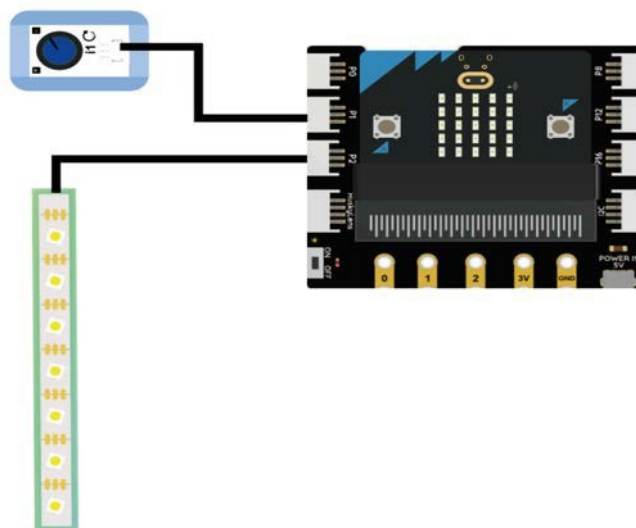
1 × tira de LED RGB



Conexión

Conecta el módulo de mando giratorio (Potenciómetro) a P1;

Conecta la tira de LED RGB a P2.



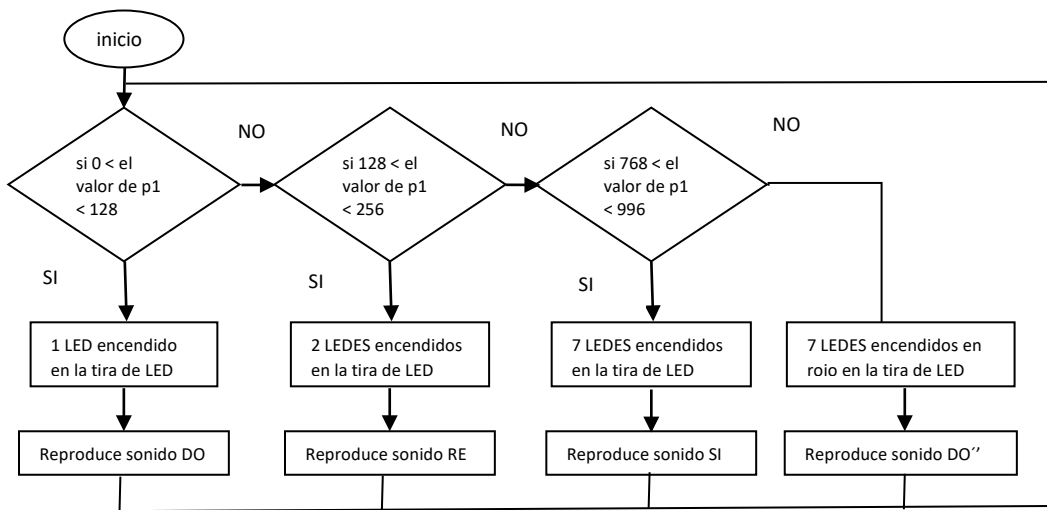
Programa

Objetivo: En este proyecto, es combinar la caja de música inteligente y la tira de LED para crear una actuación musical con la tira de LED al ajustar manualmente el ángulo de rotación del mando.

Al girar el mando manualmente, el valor de P1 cambia de 0 a 1023 y se divide en 8 pequeños intervalos, que corresponden a una octava ("Do ~ Do'") y al número de LED que se iluminarán.

El valor de entrada de P1	Sonido	La tira de LED
0-128	Do	Encender un LED
128-256	Re	Encender dos LED
256-384	Mi	Encender tres LED
384-512	Fa	Enciende cuatro LED
512-640	So	Enciende cinco LED
640-768	La	Encender seis LED
768-996	Ti	Se encienden siete LED
996-1023	DO'	Enciende siete LED en rojo

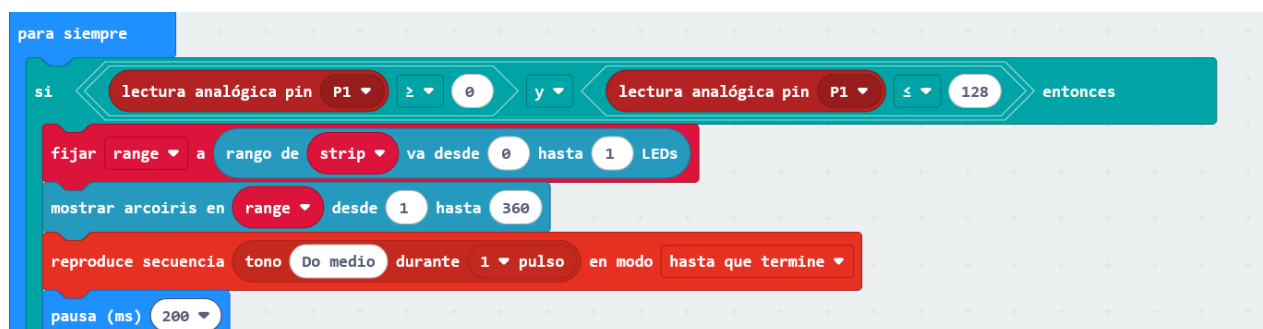
PASO 1: El diagrama lógico que aparece a continuación te ayudará a hacerte una idea general de cómo crear un DJ PANEL.



PASO 2: En este paso, aprenderemos a editar la función “si entonces”. En primer lugar, coloca el “si entonces” de “Lógica” debajo del bucle “para siempre”. Como podemos ver a continuación, hay un icono azul con forma de engranaje (marcado con un círculo rojo, como se ve a continuación) en la esquina superior izquierda de la función “si no si”. Haz clic en él y arrastra la instrucción “si no si” desde el lado izquierdo hasta la parte central, entre los elementos “si” y “si no” del lado derecho. El siguiente diagrama te ayudará a completar este paso.



PASO 3: En este paso, se aplicarán la función “y” y el operador “<” de la sección “Lógica” para evaluar el rango de valores de entrada de P1. Se utilizarán funciones de “NeoPixel” para controlar el efecto de iluminación de la tira de LED GRB. Además, se utilizarán funciones de “Música” y “pausa (ms)” de “Básico” para controlar la música. A continuación, se muestra el programa finalizado en este paso:



PASO 4: Dado que solo hay 7 LEDES, el octavo sonido, el “do”, se puede expresar mediante un efecto de iluminación diferente.



PASO 5: Combina todas las funciones mencionadas anteriormente; el programa final queda así:





Ejercicios

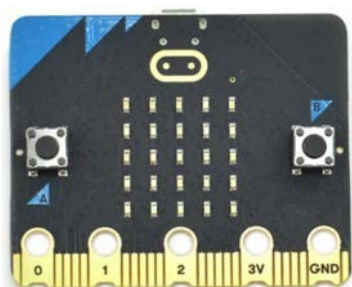
Seguro que te has dado cuenta de que el micro:bit cuenta con una matriz de LED de 5×5. ¡Por qué no sacarle el máximo partido! Intenta utilizarla para mostrar las notas musicales correspondientes a medida que se van tocando.

Proyecto 3: Timbre a distancia

Para quienes tienen un timbre en casa, puede ser bastante habitual que, a veces, el sonido no sea lo suficientemente alto como para llegar a todas las habitaciones. Para resolver este problema, una alarma inalámbrica puede resultar útil. En este capítulo, aprenderemos a hacer que el timbre controle un LED remoto que podamos llevar encima.

Lista de componentes

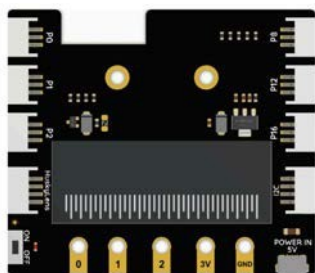
2 × micro:bit



1 × módulo de sensor de movimiento



1 × placa de expansión Boson



1 × módulo de mando giratorio (potenciómetro)



1 × módulo LED



1 × cable USB



Conexión

En este proyecto necesitaremos dos micro:bit: uno se conecta al timbre y envía el mensaje (**extremo emisor**), mientras que el otro recibe la señal y controla el indicador LED (**extremo receptor**). El micro:bit se comunica por radio

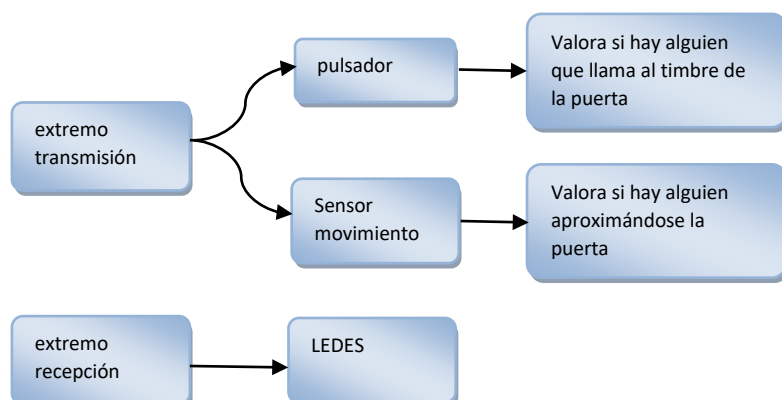


Diagrama de conexión del transmisor micro:bit:

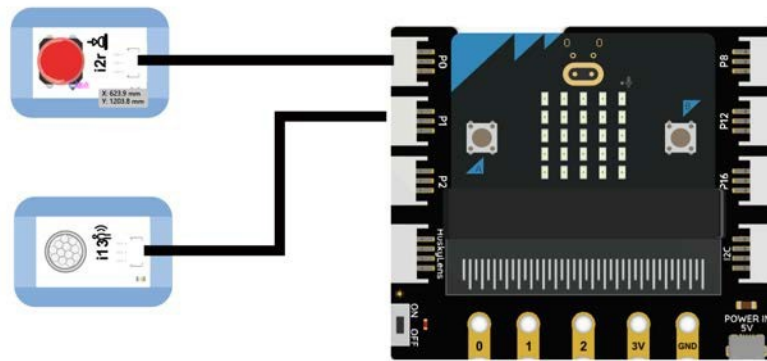
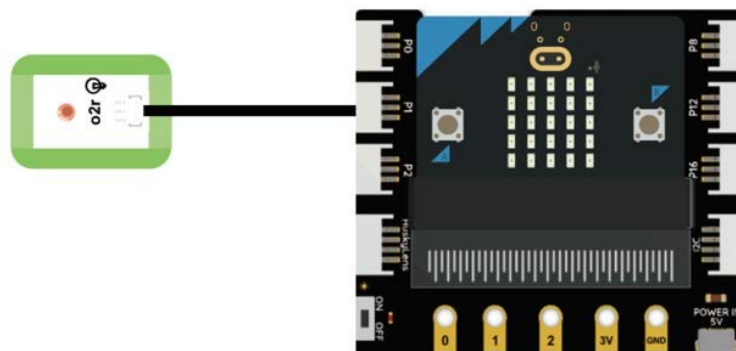


Diagrama de conexión del receptor micro:bit:

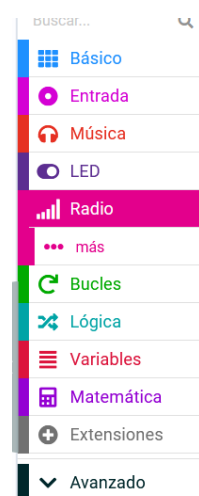


Programa

Tarea 1: Programar el trans

PASO 1: Ya estamos bastante familiarizados con el módulo pulsador y el sensor de movimiento; sin embargo, ¿cómo podemos transmitir su estado por radio? Para lograrlo, aprenderemos cómo funciona exactamente la radio en la siguiente sesión.

La función “Radio” se encuentra en la parte izquierda de la ventana de MakeCode.



En primer lugar, debemos comprender las tres funciones más utilizadas de la sección “Radio”: “establecer grupo”



Descripción de la función: Hay un total de 255 canales de radio diferentes. El transmisor y el receptor deben configurarse en el mismo grupo para poder comunicarse.

Enviar número”:



Descripción de la función: envía un número por radio.

“al recibir radio”:

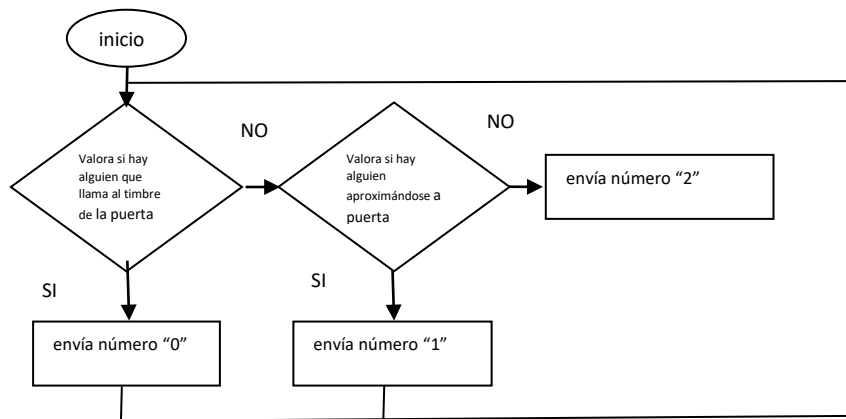


Descripción de la función: Al recibir un número, se ejecutará el programa dentro del bucle “al recibir radio”.

También tendremos que programar el receptor para que realice la acción correspondiente en función del número recibido.

Nota: la variable “recibirNumero” dentro de la función “al recibir radio” representa el número recibido.

PASO 2: Una vez que sabemos cómo utilizar la función de radio, es el momento de emplear módulos de entrada para activar la señal de radio. Para que resulte un poco más complicado, intentaremos programar el micro:bit para que envíe un mensaje cuando alguien toque el timbre y otro mensaje cuando simplemente pase alguien por allí. Para ello, necesitaremos el módulo pulsador y un sensor de movimiento. El diagrama de flujo que aparece a continuación te da una idea de cómo escribir el programa

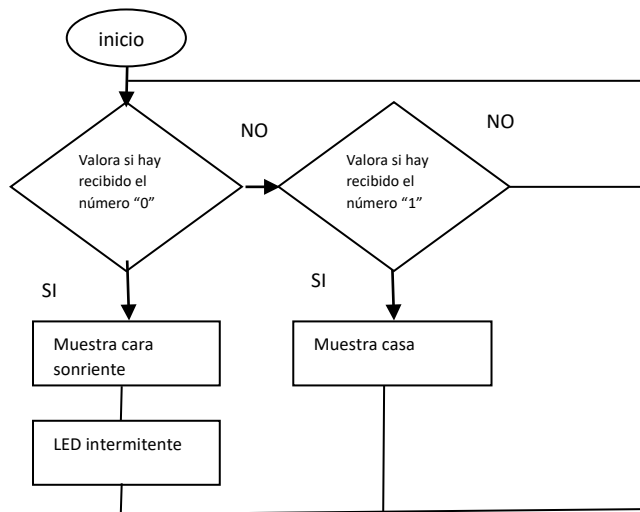


PASO 3: Une los bloques de función. El programa completo tendrá el siguiente aspecto.

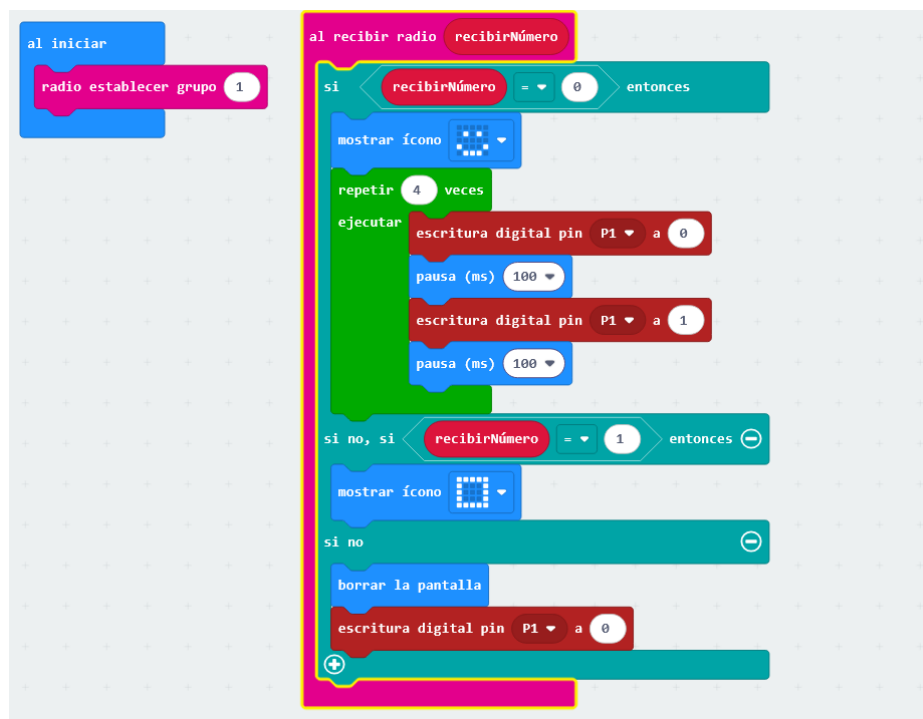


Tarea 2: Programar el receptor

PASO 1: El receptor es capaz de ejecutar diferentes acciones en función de la señal recibida. El diagrama de flujo que aparece a continuación te ayudará a crear el programa.



PASO 3: Une los bloques de función. El programa completo quedará así.



Ejercicio

Añade un mensaje de “Por favor, espera” al timbre cuando alguien llame a la puerta.

Proyecto 4: Escapar del laberinto

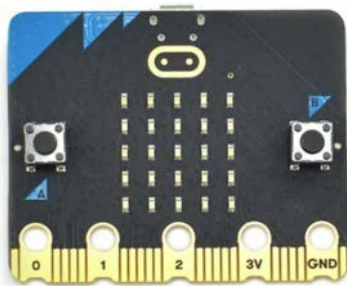
¡Micro:bit está atrapado dentro del laberinto con una bomba de relojería! El pobre Micro:bit no puede moverse por sí mismo.

Sin embargo, envía señales a través del panel LED para indicar el camino. ¡Tenemos que ayudarlo a escapar del laberinto antes de que explote la bomba!



Lista de componentes

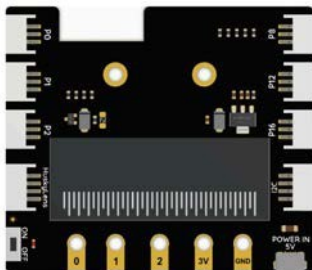
1 × micro:bit



1 × tira de LED RGB



1 × placa de expansión Boson

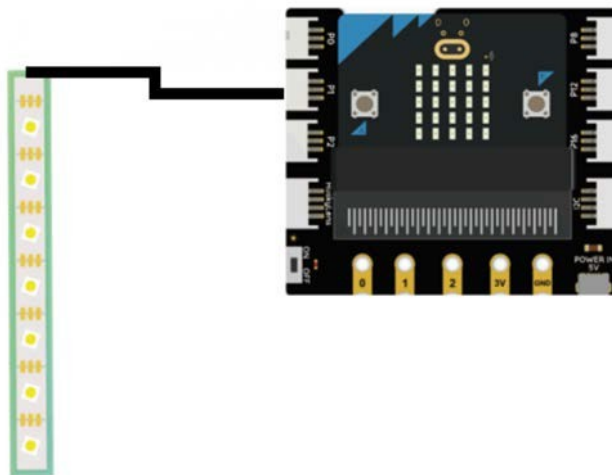


1 × cable USB



Conexión

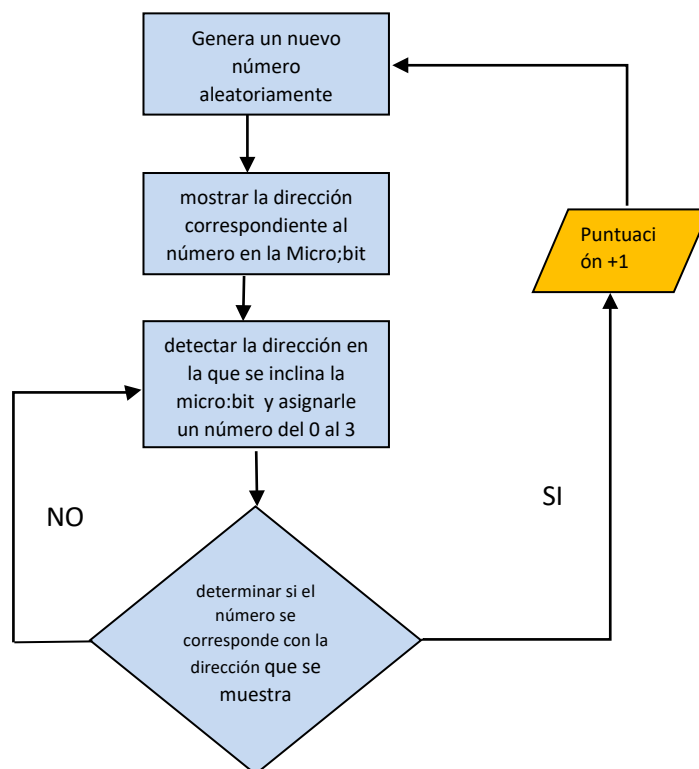
Conecta la tira de LED RGB a P1.



Programa

PASO 1: En primer lugar, necesitaremos para que el micro:bit genere aleatoriamente algunos movimientos con el fin de simular el laberinto. Además, el micro:bit debería ser capaz de detectar en qué dirección se inclina. Si las dos direcciones coinciden, se añadirá 1 punto a la puntuación final, lo que significa que estamos un paso más cerca de escapar del laberinto.

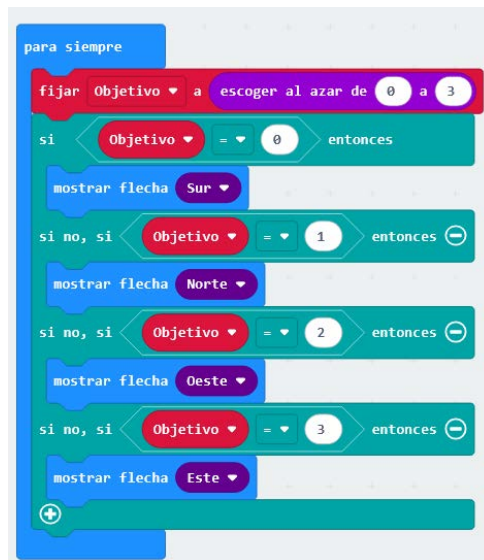
El siguiente diagrama de flujo puede resultar útil para explicar cómo funciona el programa.



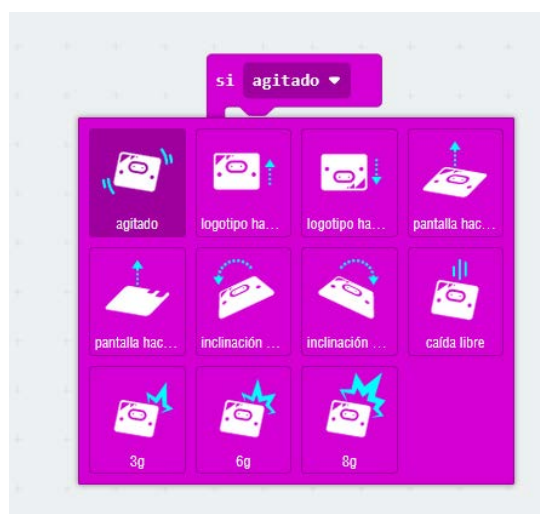
Utilizaremos los números del 0 al 3 para representar las cuatro direcciones de destino, respectivamente. La función “escoger al azar” de la sección “Matemática” nos ayuda a obtener un número aleatorio entre 0 y 3.

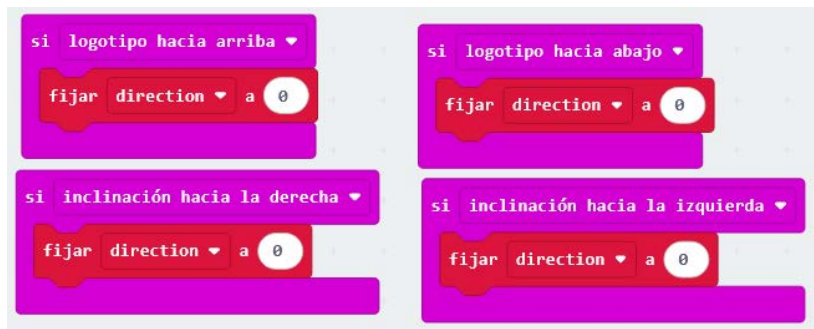


PASO 2: Mostrar la flecha en el panel LED. La flecha debe corresponder a la dirección de destino generada aleatoriamente que hemos obtenido anteriormente. Una vez más, la función “si entonces” será nuestra elección para realizar esta tarea.



PASO 3: Utiliza la función “si agitado” en la sección “entrada” para detectar en qué dirección se inclina el micro:bit. El número que aparece dentro del bucle de la función representa cuatro direcciones diferentes.





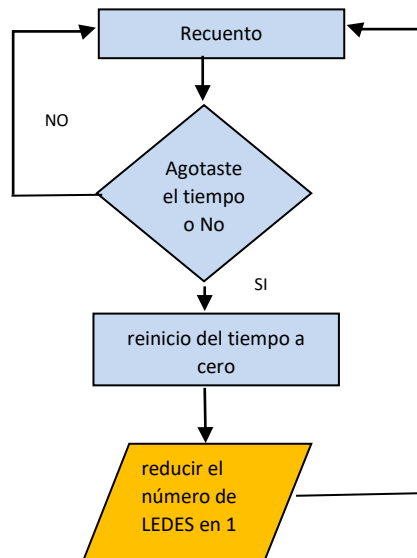
PASO 4: Coloca todo lo que hemos mencionado dentro de un bucle “si entonces” para comparar si las dos direcciones coinciden. Además, no te olvides del bucle “para siempre” para que el juego siga en marcha continuamente.



Nota: resultará útil añadir una función “mientras dentro del bucle, para evitar que aparezcan direcciones idénticas seguidas.

PASO 5: es el momento de añadir la bomba. La bomba empezará a contar atrás en cuanto comience el juego. Además, utilizaremos la tira de LED RGB para simular la mecha. A medida que pase el tiempo, los LED de la mecha se irán apagando uno tras otro a un ritmo constante. Si el micro:bit no consigue escapar del laberinto antes de que se agote la mecha, el juego habrá terminado.

El diagrama de flujo que aparece a continuación muestra cómo configurar los LED para que se enciendan uno a uno a medida que pasa el tiempo.



Explicaremos las funciones que necesitamos para lograrlo.

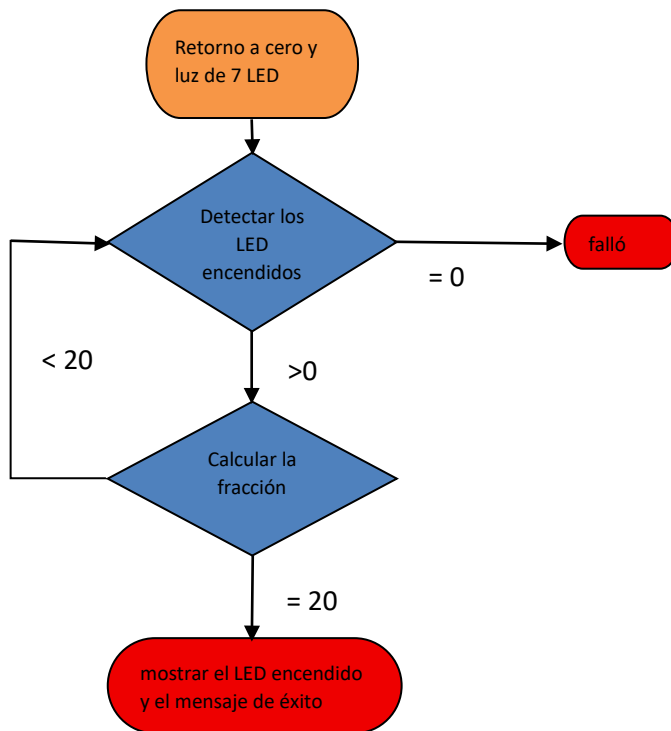
Nota: El Micro:bit necesita algo más de tiempo para procesar los datos, por lo que el reloj funcionará un poco más lento de lo que hayamos configurado.



El siguiente programa se utiliza para establecer cuántos LED se encienden.



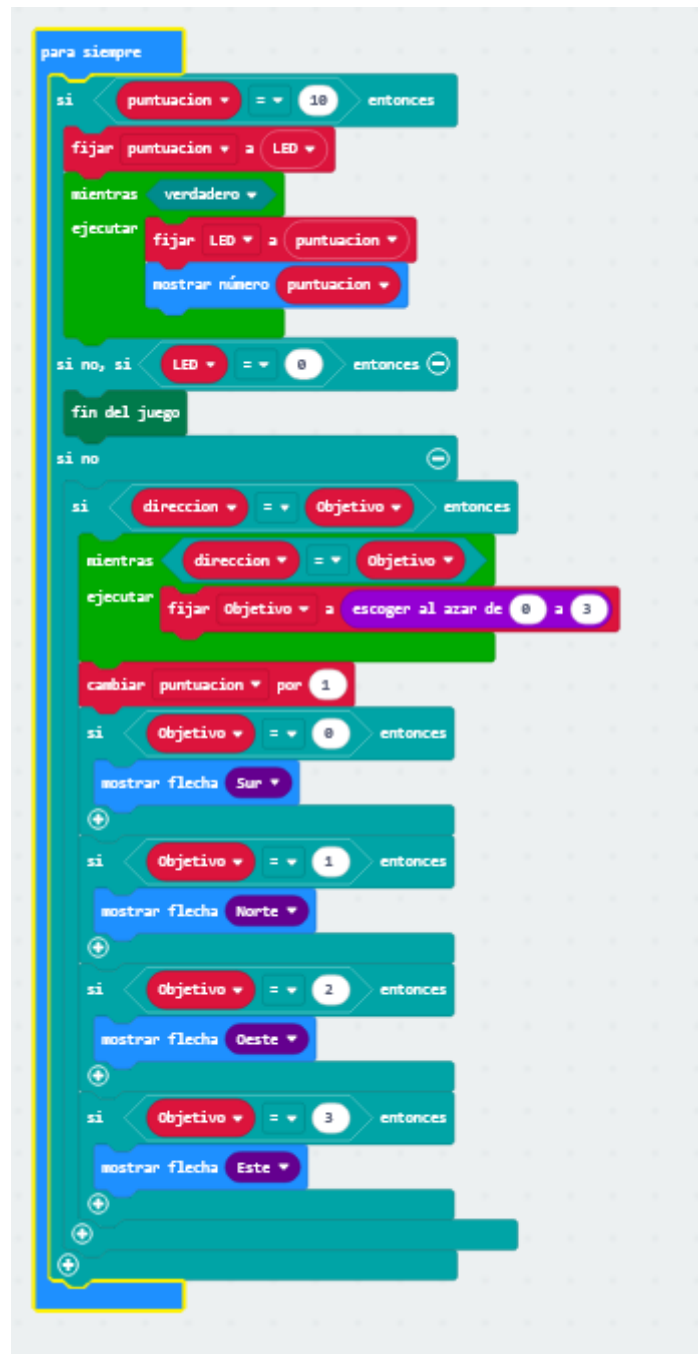
PASO 6: Necesitaremos un mecanismo de puntuación que nos indique si hemos conseguido escapar. El siguiente diagrama de flujo te puede dar una idea de cómo funciona...



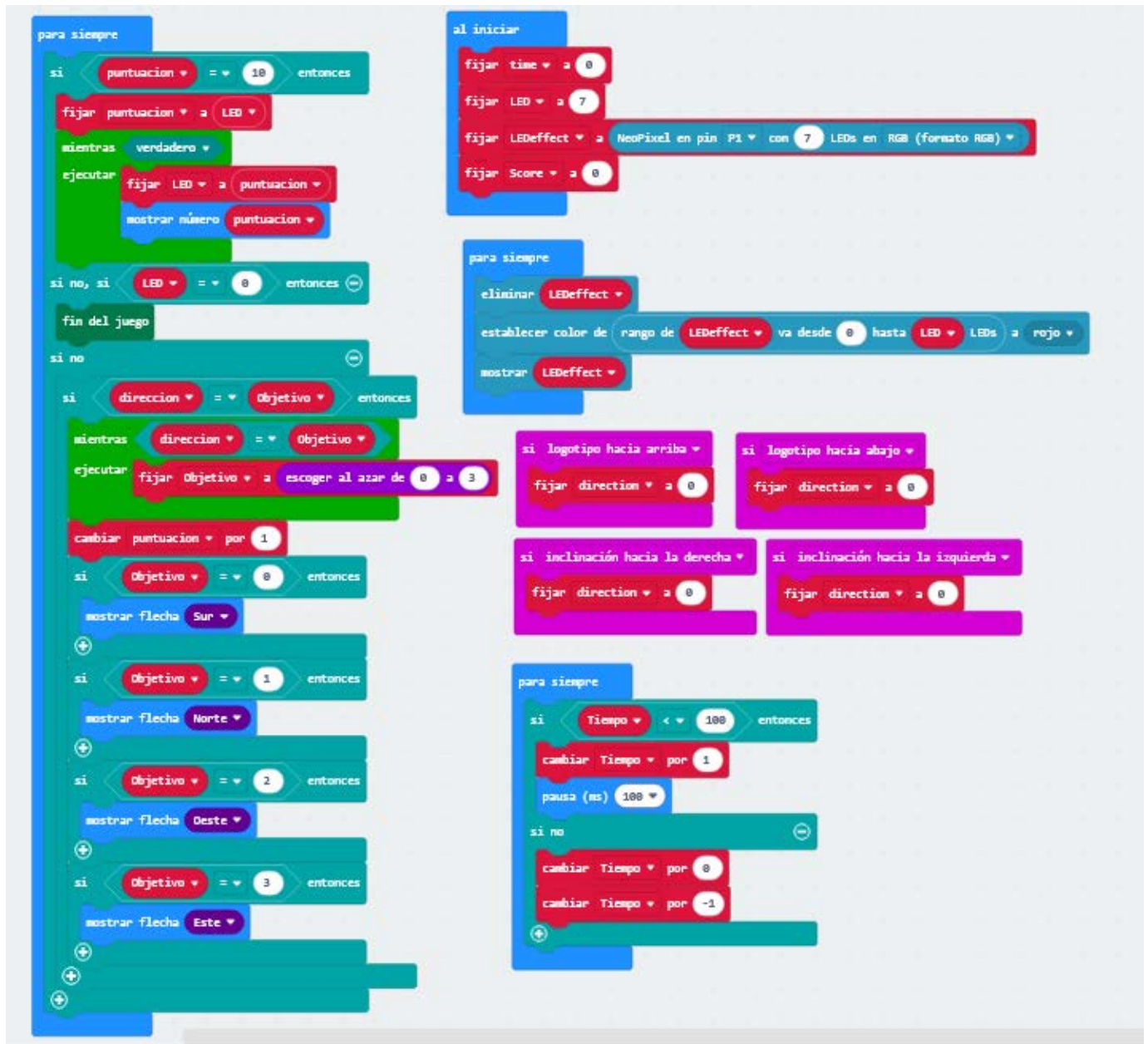
Recuerda poner a cero la puntuación al inicio del programa.



PASO 7: Ahora, integra la función de puntuación en el programa principal.



PASO 8: Así es como queda el programa final



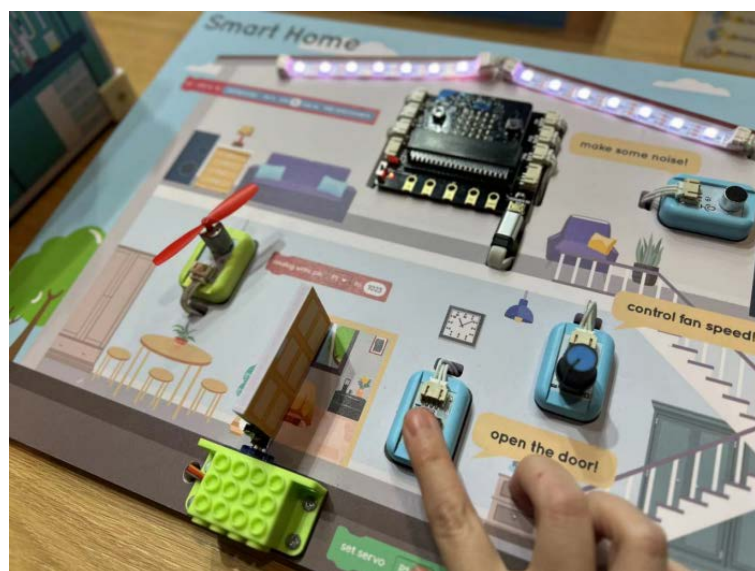
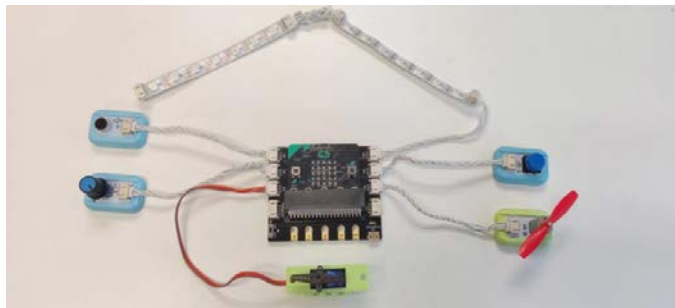
Ejercicio

Intenta que el reloj marque el tiempo más rápido para aumentar la dificultad; Programa los LED para que parpadeen a un ritmo que se base en el tiempo restante;

¡Un reto poco intuitivo! Establece diferentes direcciones por pares y comprueba si aun así puedes terminar el juego. Por ejemplo, tendrás que colocar la ficha de arriba para que coincida con una flecha hacia la izquierda, y la de abajo para que coincida con una flecha hacia la derecha.

Tu aventura con el micro:bit no termina aquí.

Por ejemplo: [Cómo crear una casa inteligente con micro:bit y BOSON](#)



Si quieres compartir tus ideas con nosotros, ¡visita el blog de DFRobot y cuéntanoslas!

[¡Bienvenido al blog de DFRobot!](#)