



错误!未定义书签。

<b>Contenidos</b>	错误!未定义书签。
<b>Capítulo 1: MakeCode y micro:bit</b>	<b>4</b>
Introducción a MakeCode	4
Breve introducción a Micro:bit	4
Como utilizar micro:bit	5
PASO 1: Abrir MakeCode	6
PASO 2: Conectar Micro:bit a una computadora	6
PASO 3: Iniciar un nuevo proyecto	7
PASO 4: Descargue el programa y cárguelo a Micro:bit	9
<b>Capítulo 2: Que hace que una maquina “Cobre vida”</b>	<b>12</b>
Dispositivos interactivos	12
Unidad de entrada — sensor	12
Unidad de control —Micro:bit	12
Unidad de salida — actuador	13
Relación entre Hardware y programa	错误!未定义书签。
Placa de expansión Boson para Micro:bit	13
<b>Capítulo 3: Manos a la obra!</b>	<b>14</b>
<b>Proyecto 1: El misterioso Micro:bit</b>	<b>14</b>
Lista de componentes	错误!未定义书签。
Conexión	错误!未定义书签。
Programa	15
Ejercicios	错误!未定义书签。
<b>Proyecto 2: Luz LED intermitente</b>	<b>19</b>
Lista de componentes	错误!未定义书签。
Conexión	20
Programa	20
Ejercicios	错误!未定义书签。
<b>Proyecto 3: Luz de Notificación</b>	<b>23</b>
Lista de componentes	错误!未定义书签。

Conexión.....	24
Programa .....	24
Ejercicios.....	28
<b>Proyecto 4: Fan electrico.....</b>	<b>29</b>
Lista de componentes .....	错误!未定义书签。
Conexión.....	30
Programa .....	31
<b>Capítulo 4: Un poco mas allá.....</b>	<b>35</b>
<b>Proyecto 1: Vela Electronica .....</b>	<b>35</b>
Lista de componentes .....	错误!未定义书签。
Conexión.....	36
Programa .....	36
<b>Proyecto 2: Puerta Automatica .....</b>	<b>40</b>
Lista de componentes .....	错误!未定义书签。
Conexión.....	41
Programa .....	41
<b>Proyecto 3: Caja Musical.....</b>	<b>46</b>
Lista de componentes .....	错误!未定义书签。
Conexión.....	47
Programa .....	47
<b>Proyecto 4: Tira LED de colores .....</b>	<b>51</b>
Lista de componentes .....	错误!未定义书签。
Conexión.....	52
Programa .....	52
<b>Capítulo 5: Conviertase en experto .....</b>	<b>60</b>
<b>Proyecto 1: Estabilizador electronico .....</b>	<b>错误!未定义书签。</b>
Lista de componentes .....	错误!未定义书签。
Conexión.....	61
Programa .....	61
Ejercicios.....	63
<b>Proyecto 2: Panel DJ .....</b>	<b>64</b>
Lista de componentes .....	错误!未定义书签。
Conexión.....	65

Programa .....	65
Ejercicios.....	67
<b>Proyecto 3: Timbre remoto .....</b>	<b>69</b>
Lista de componentes .....	错误!未定义书签。
Conexión.....	70
Programa .....	71
Ejercicios.....	74
<b>Proyecto 4: Escape del laberinto .....</b>	<b>错误!未定义书签。</b>
Lista de componentes .....	错误!未定义书签。
Conexión.....	76
Programa .....	76
Ejercicios.....	83

Recuerde visitar el blog de DFRobot para más tutoriales, proyectos y últimas tendencias. El enlace se encuentra en la esquina superior izquierda de la página.

## Capítulo 1: MakeCode y micro:bit

### Interacción a MakeCode

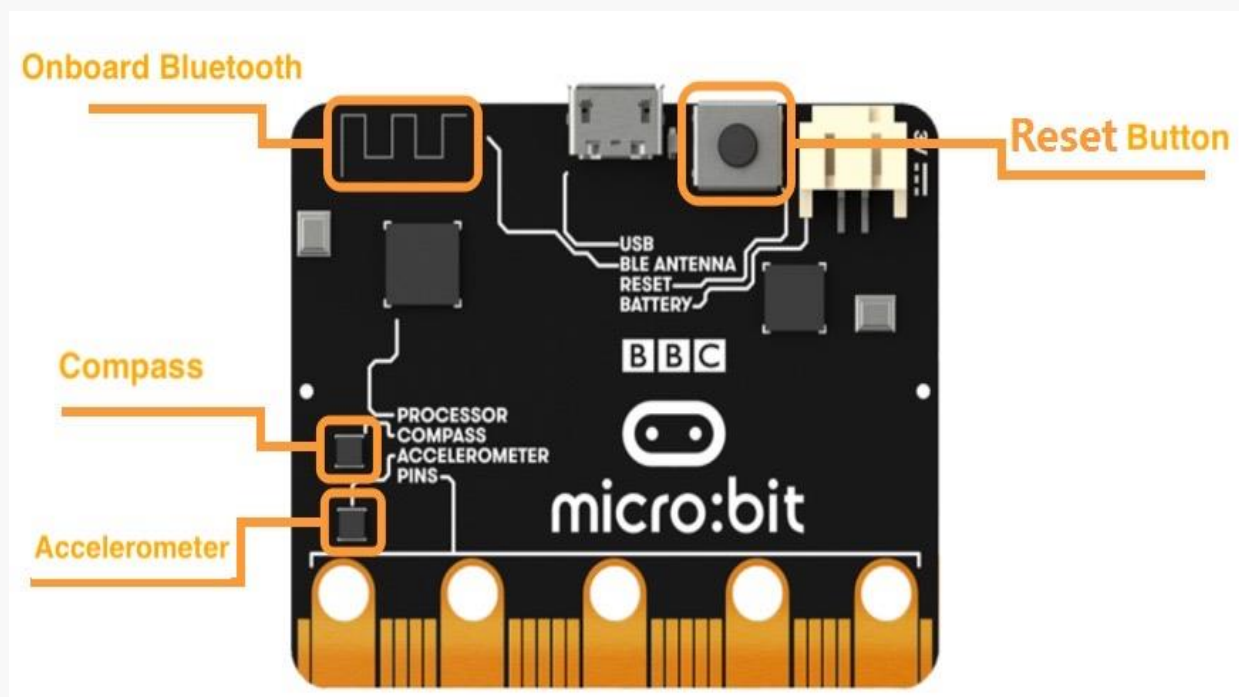
MakeCode para micro: bit es uno de los entornos de programación gráfica más utilizados desde el sitio web Micro:bit. Es un proyecto de código abierto desarrollado por Microsoft.

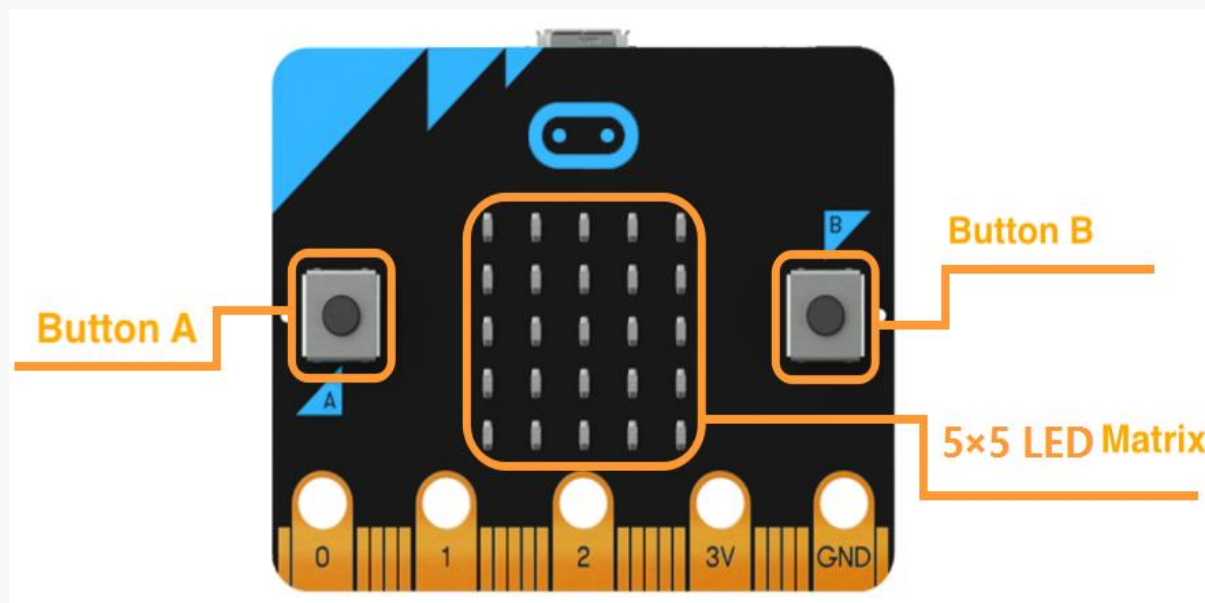
### Una breve introducción a micro: bit

micro:bit es un microcontrolador de bolsillo diseñado para niños y principiantes que aprenden a programar, permitiéndoles aportar fácilmente ideas en juegos digitales, proyectos interactivos y robótica.

micro:bit viene con una variedad de módulos a bordo, incluyendo una matriz LED 5x5 (también soporta detección de luz), 2 botones programables, detector de movimiento, brújula y módulo Bluetooth® Smart. Además, puede conectar más módulos como un servomotor, luces LED RGB a través de 5 anillos de E/S o 20 conectores de extremo.

Con el micro:bit se pueden realizar muchas ideas interesantes. Todo lo que pueda imaginar, un robot, un instrumento eléctrico o incluso un sistema de domótica. ¡Las posibilidades son infinitas! El Micro:bit contiene una serie de características innovadoras como 25 LED rojos para mostrar mensajes y dos botones programables para controlar el juego o la música. Puede detectar movimiento, reconocer gestos y estar interconectado con otros dispositivos o con Internet a través de una conexión Bluetooth.





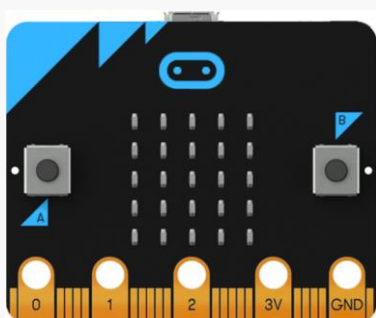
El Micro:bit está equipado con sensores de luz y temperatura y otros dispositivos de detección comunes, lo que significa que también puede producir una gran cantidad de productos inteligentes comunes utilizados en nuestra vida diaria.

## Como utilizar micro:bit

Si eres nuevo en Micro:bit, puedes empezar con la plataforma de programación en línea - The MakeCode Editor, para aprender a programar el Micro:bit.

Antes de empezar, por favor asegúrese de que los siguientes elementos estén listos a mano. Además, necesitará un ordenador con sistemas operativos como Windows 10 / Mac OS / Linux y con conectividad a Internet.

### LISTA DE COMPONENTES:



Placa base Micro:bit

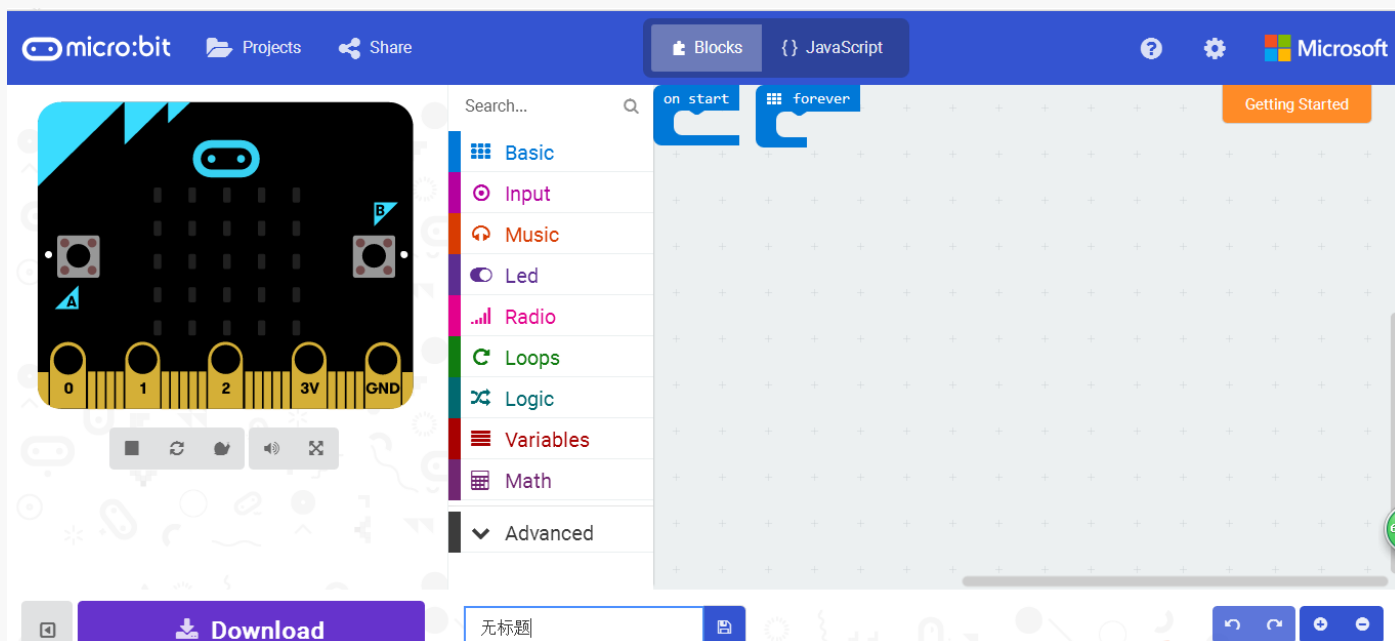


Cable USB

Los siguientes pasos están basados en el sistema operativo Windows 10. Puede ser utilizado como referencia para otros sistemas operativos.

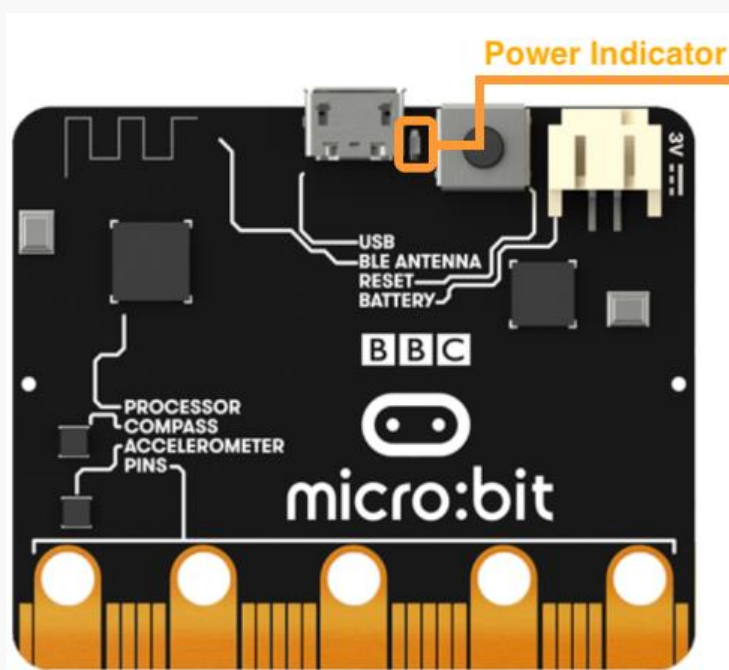
## Paso 1: Abriendo MakeCode

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

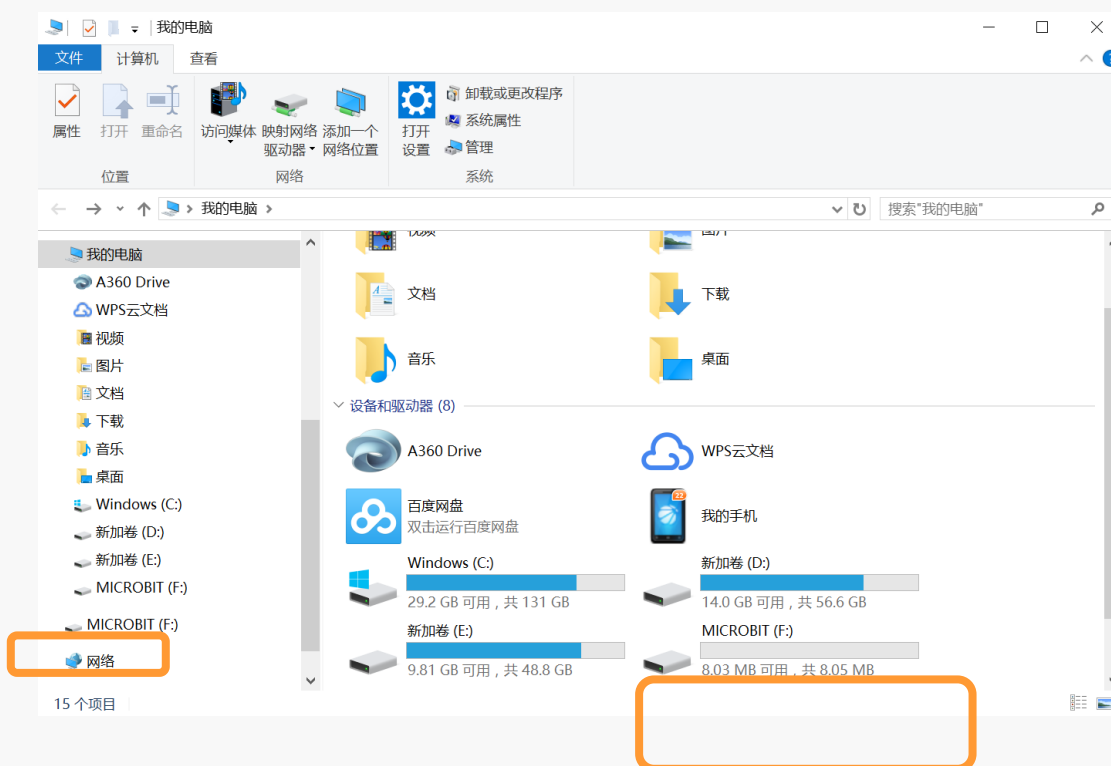


## PASO 2: Conectar Micro:bit a la computadora

El Micro:bit se conecta al ordenador a través del cable USB. El indicador de alimentación en la parte posterior del micro:bit se encenderá cuando esté conectado.

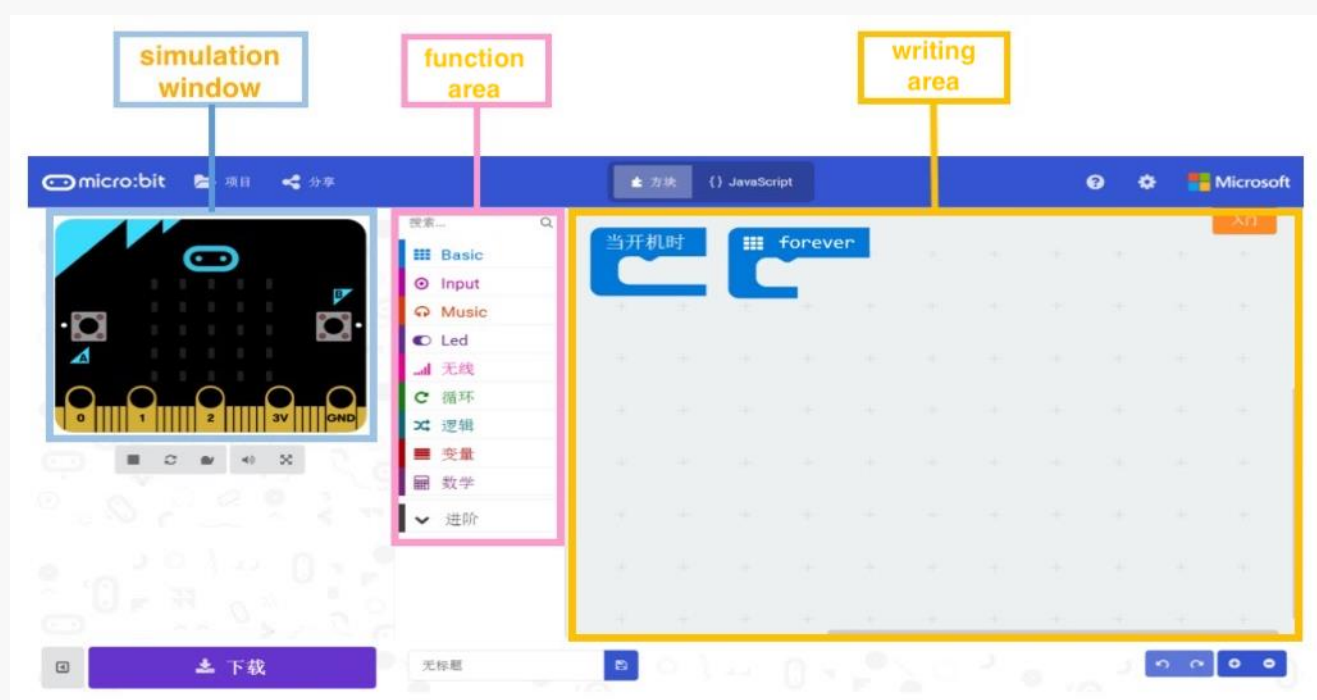


Antes de programar, debemos asegurarnos de que la placa base es reconocida por el ordenador. Cuando el Micro:bit está conectado, un directorio "MICROBIT" aparecerá en "Este Equipo".



### PASO 3: Iniciar un Nuevo proyecto

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

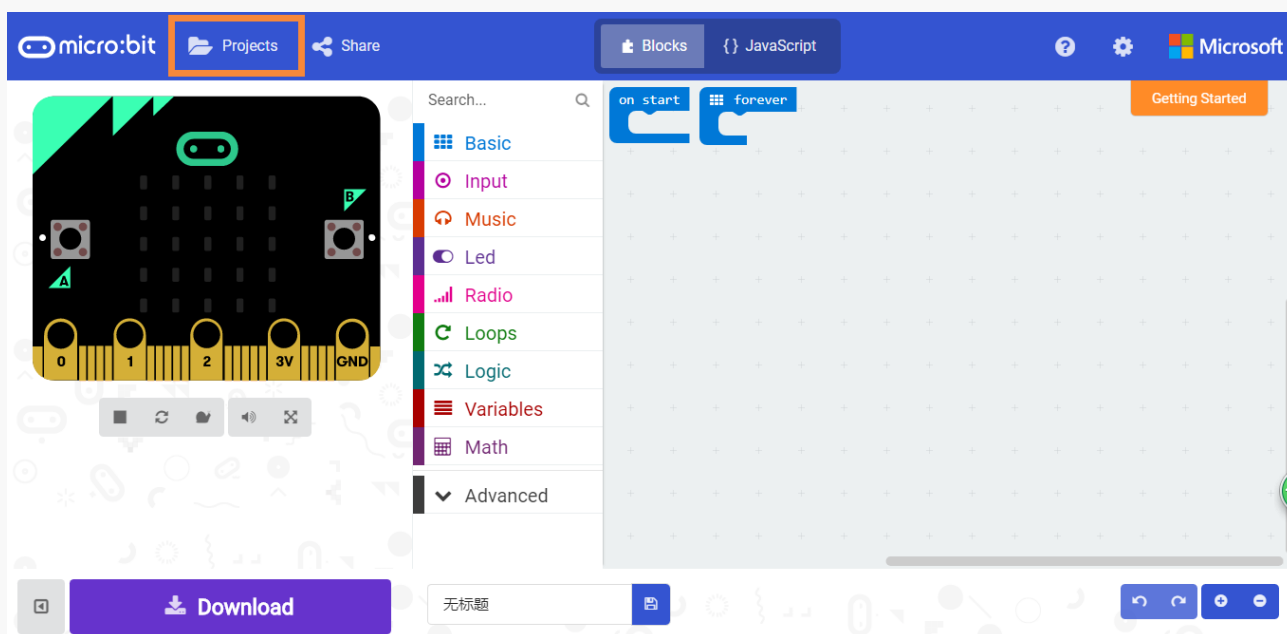


Ventana de simulación: simula el estado de funcionamiento del Micro:bit. Durante el proceso de programación, siempre puede comprobar cómo se ve su programa a través de la ventana.

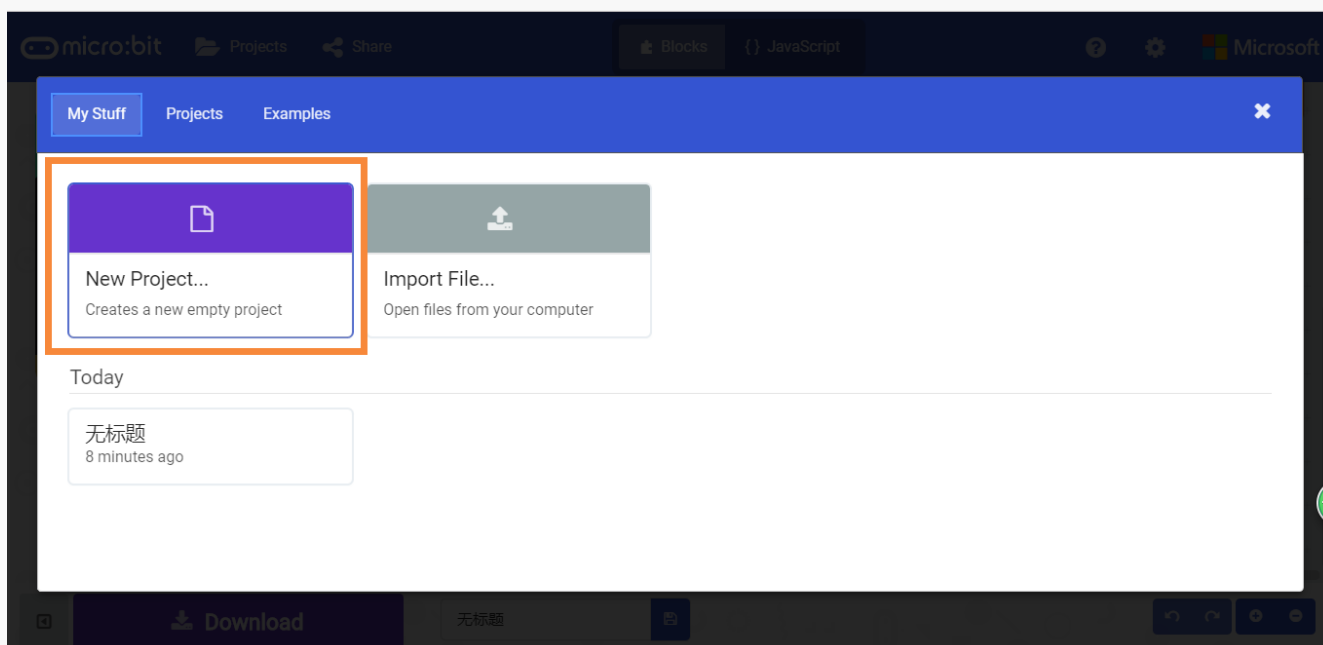
Área funcional: Donde podrá encontrar todos los bloques de funciones, incluyendo entrada, salida, bucle, lógica, etc.

Área de programación: Arrastre los bloques desde el "Área de funciones", apílelos y construya su programa aquí.

Haga clic en "Projects" en la parte superior de la ventana de simulación.



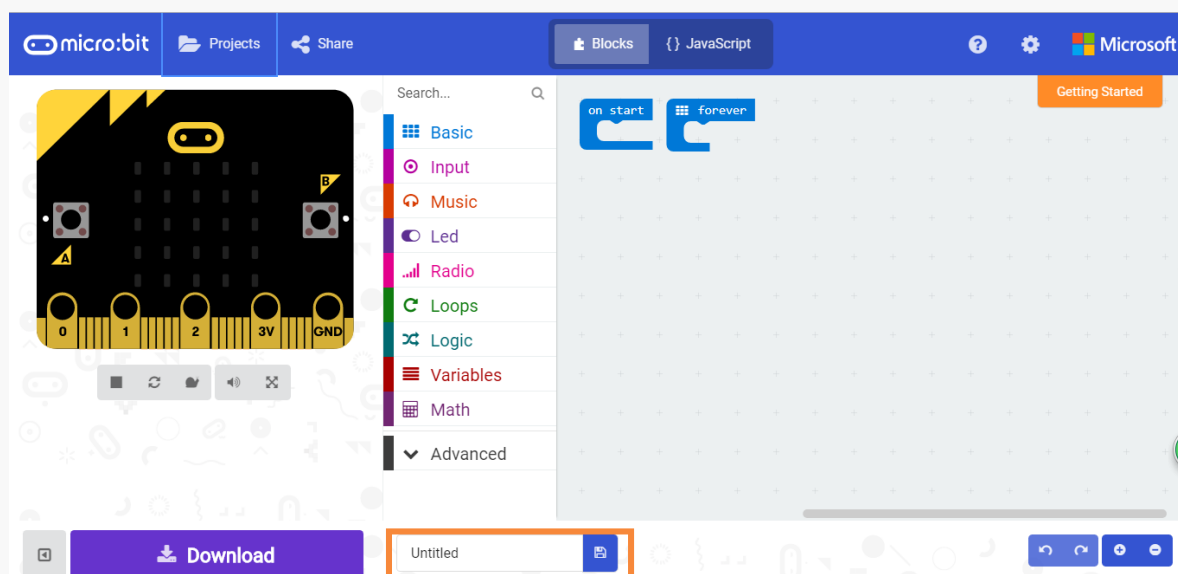
Luego, clic en "New Project".



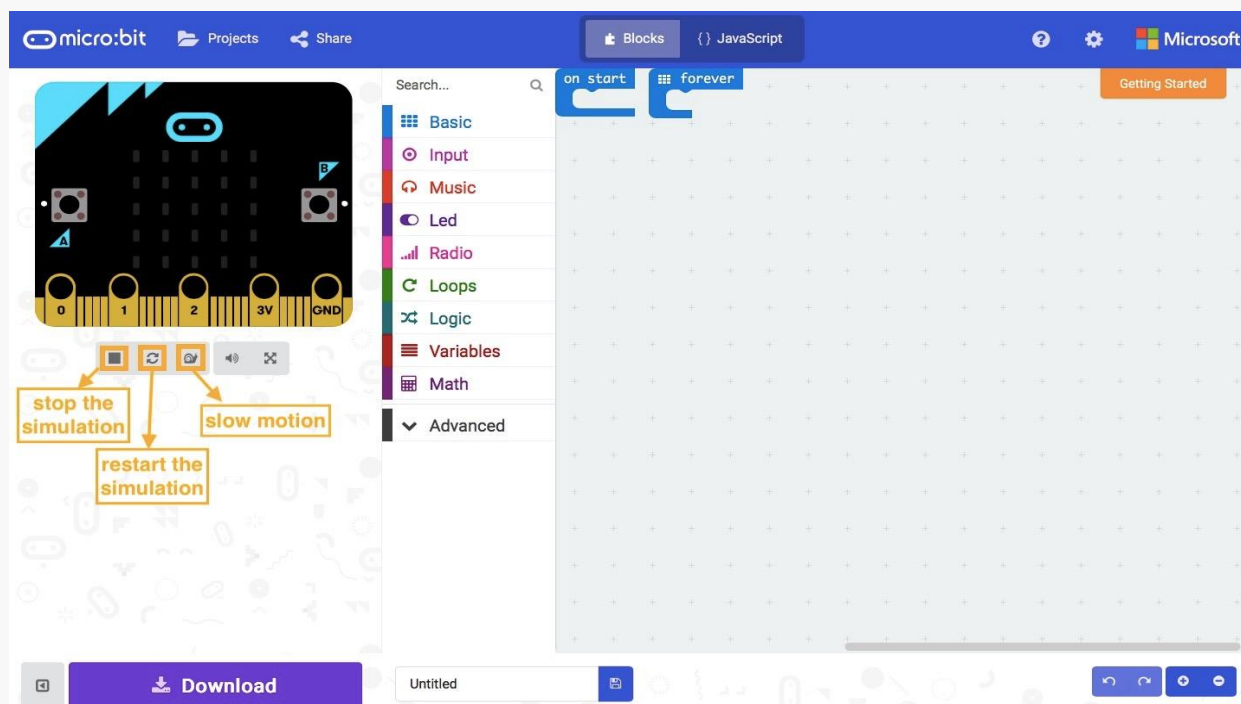


## PASO 4: Descargue el programa y cárguelo a Micro:bit

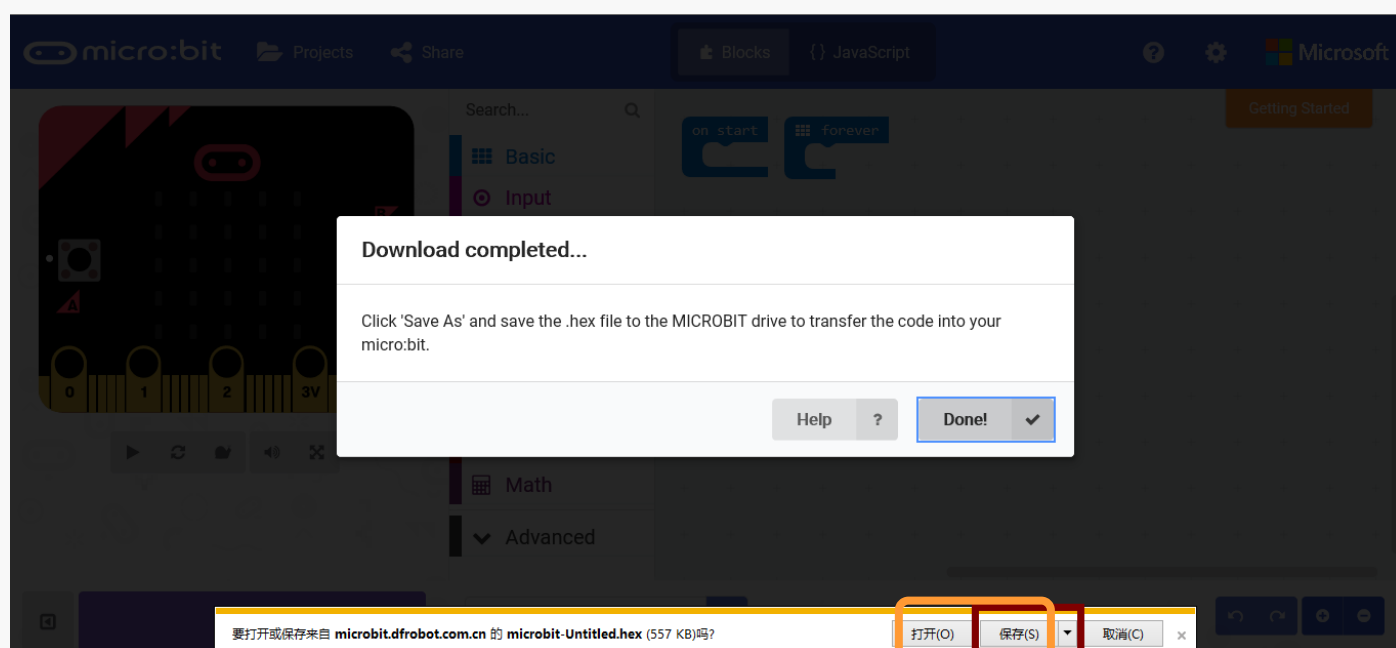
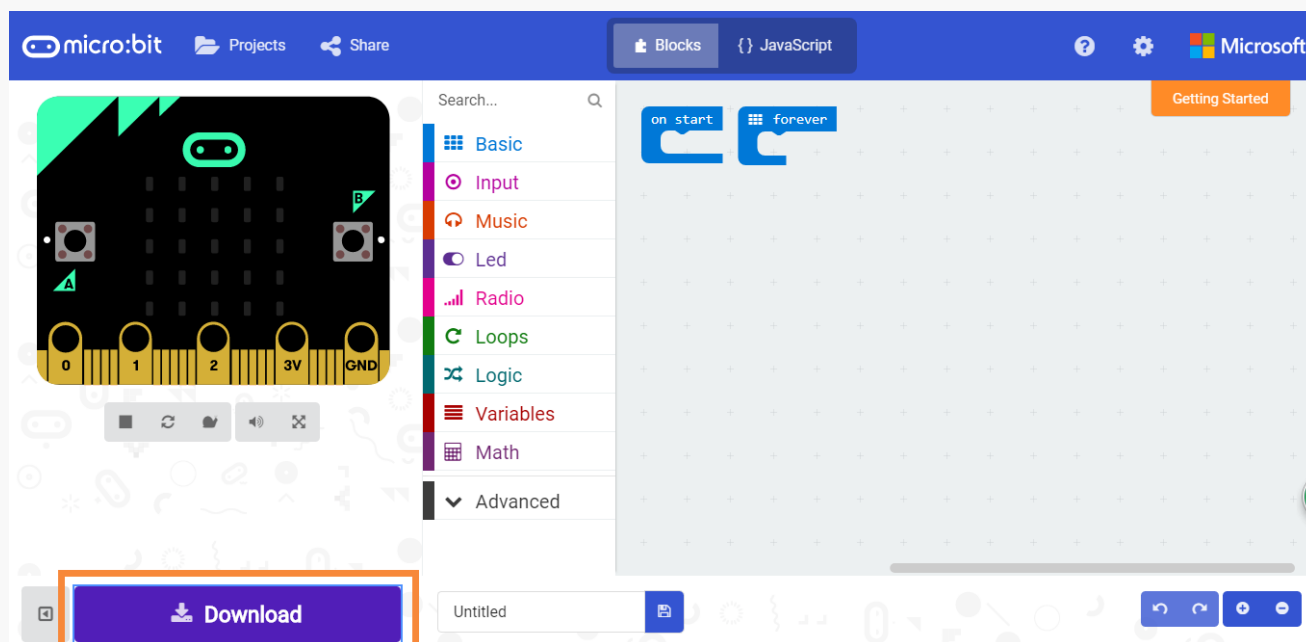
Cuando hayamos terminado nuestro programa, ya podemos descargarlo desde la web y subirlo a Micro:bit. También podemos renombrar el proyecto y guardarlo en el navegador. El proyecto se quedará en "my stuff".



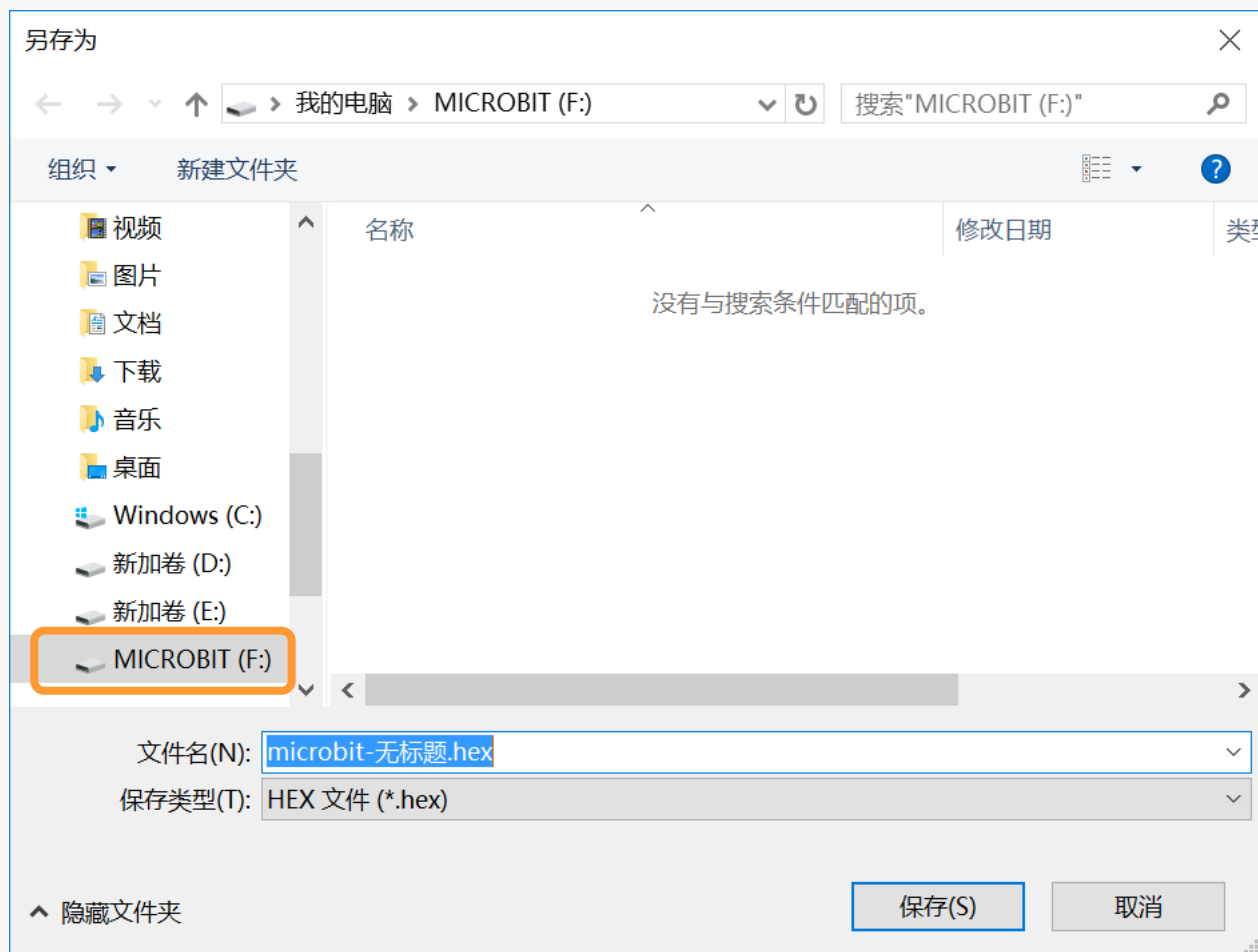
Antes de la descarga, podemos comprobar los resultados simulados en la ventana de simulación. Los botones de la parte inferior de la ventana de simulación permiten controlar el micro:bit analógico.



Haga clic en "Download" en la parte inferior de la ventana de simulación y seleccione "Save as" en el cuadro de diálogo emergente.



Elija guardar el archivo ".hex" en "MICROBIT Disk" y haga clic en "Save".



Durante el proceso de descarga, el indicador de encendido en la parte posterior del Micro:bit parpadeará. Una vez terminado, dejará de parpadear y seguirá funcionando.

Ahora todo está bien para las configuraciones básicas. Recuerda visitar nuestro blog para ver más proyectos Micro:bit. Por favor, también deje un comentario si tiene alguna pregunta o nueva idea para compartir.

<https://www.dfrobot.com/index.php?route=DFblog/blogs>

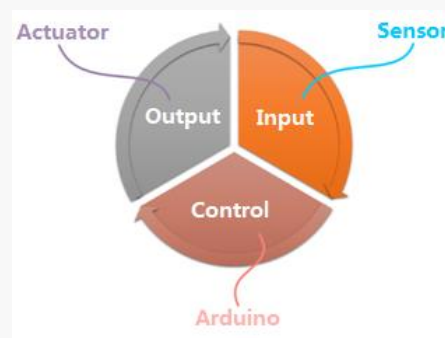
## Capítulo 2: Que hace que una maquina “Cobre vida”

### Dispositivos interactivos

En los próximos capítulos, crearemos proyectos interactivos como un solo LED controlado por botones, una simple animación LED colorida o incluso crearemos nuestra propia música. Estos dispositivos pueden ser categorizados como "dispositivos interactivos". Para ayudar a construir una mejor comprensión de sus mecanismos, necesitamos primero entender cómo están compuestos.

El dispositivo de interacción más sencillo consta de 3 partes :

- Unidad de entrada para aceptación de comandos o recopilación de datos
- Unidad de control para el procesamiento de datos o señales
- Unidad de salida para enviar datos o ejecutar



Usted debe estar confundido por estas concepciones y ansioso por saber cómo funcionan realmente. Tomemos nuestro propio cuerpo como ejemplo, recogemos información en forma de luz, sonido, sabor y poder a través de nuestros ojos, oídos, nariz y piel. Esta información entra en nuestro cerebro y determina qué respuesta tomar. En última instancia, tomamos medidas físicas basadas en esta información para cambiar el entorno material. Específicamente, tu amigo, por ejemplo, te dice "hola", y tú le respondes "hola". Aquí, sus oídos actúan como unidades de entrada, su cerebro actúa como una unidad de control, y su boca juega el papel de unidad de salida.

Del mismo modo, al construir proyectos utilizando el Micro:bit, utilizaremos una variedad de 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 ambientales como la luz, la temperatura, la humedad, etc. y las convierten en datos de señal. En este tutorial, se aplicarán sensores como botones, sensores de sonido y sensores de temperatura.

#### Unidad de control — micro: bit

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

## Unidad de salida — actuador

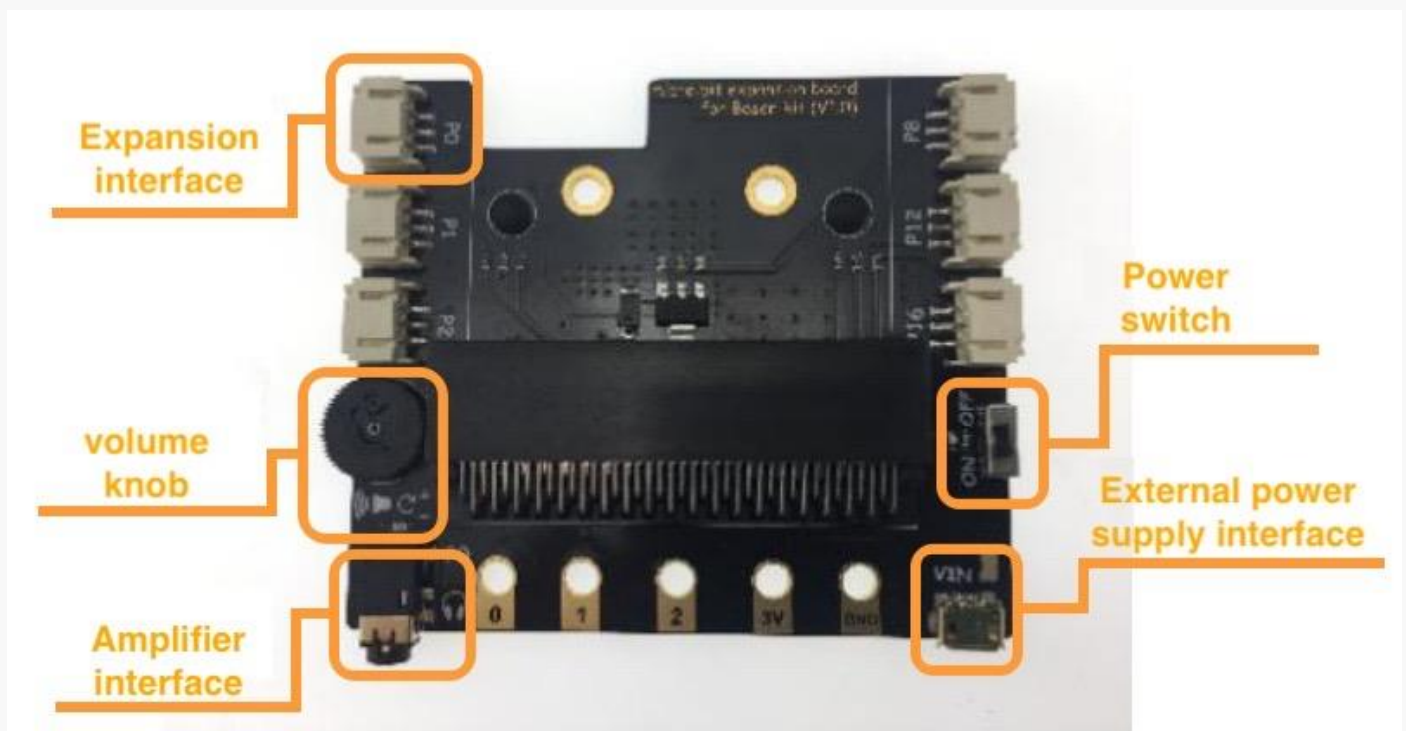
El actuador es el responsable de mover o controlar un sistema o mecanismo. Convierte la energía eléctrica en movimiento, sonido y luz. En este tutorial, los actuadores como LEDs, pequeños ventiladores y servos serán aplicados.

## Relación entre Hardware y Programa

La unidad de entrada, la unidad de control y la unidad de salida mencionadas anteriormente son componentes físicos. En cuanto al 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 aquí es como nuestra mente y lo que explicaremos es cómo configurar la "mente" del micro:bit para que haga lo que le pidamos que haga.

## Placa de expansión Boson para Micro:bit

La placa de expansión de Boson permite que el micro:bit conecte módulos externos, incluyendo botones, módulos de interruptores y sensores. No sólo simplifica la conexión del circuito, sino que también hace que el micro: un poco más potente.



## Capítulo 3: manos a la obra!

Ahora puede tener una idea aproximada sobre qué son Micro:bit y tarjeta de expansión y cómo funciona MakeCode. Debes estar ansioso por ponerle la mano encima tú solo. Así que, comencemos el viaje mágico de Boson para 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 emoji se pueden ver en casi todos los smartphones y ordenadores, pueden aparecer de la siguiente manera:



Amor



Está tranquilo



Te estoy observando

En este capítulo, te mostraremos cómo diseñar tus propios emojis programando el micro:bit.

El micro:bit trae incorporado una matriz de led de 5x5 (25) leds, los cuales podemos controlar y formar patrones para mostrar emojis.

### Lista de componentes

1 × micro:bit



1 × Cable USB



## Conexión

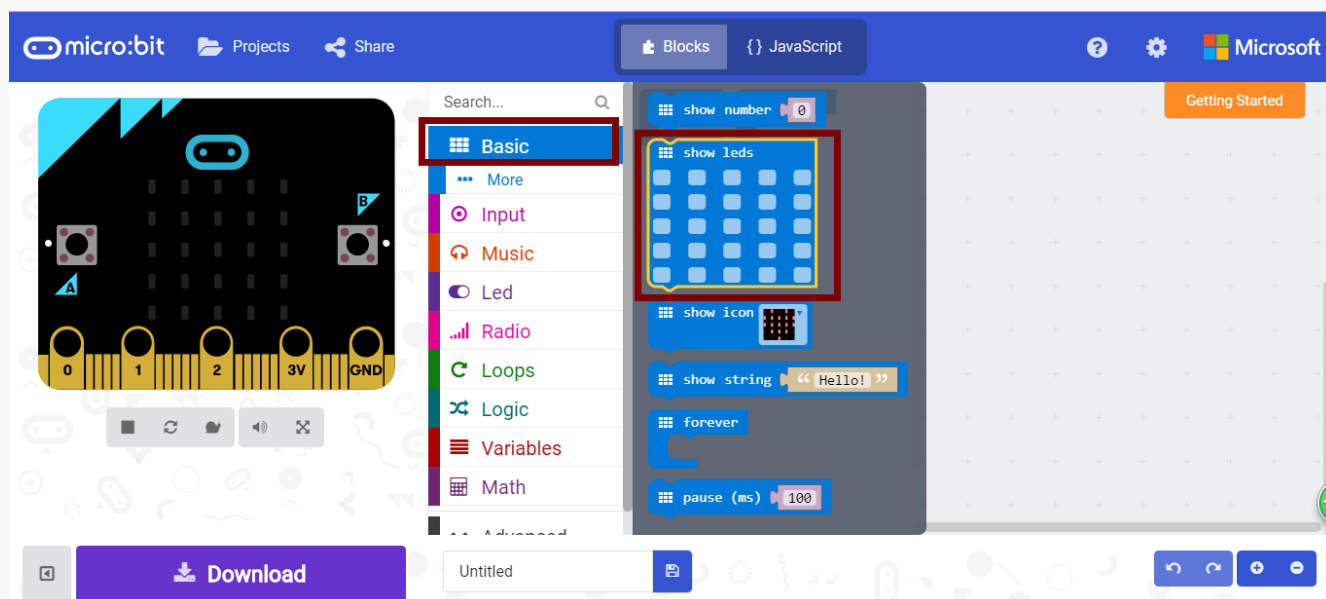
Conecte el Micro:bit al ordenador a través del cable USB.

## Programar

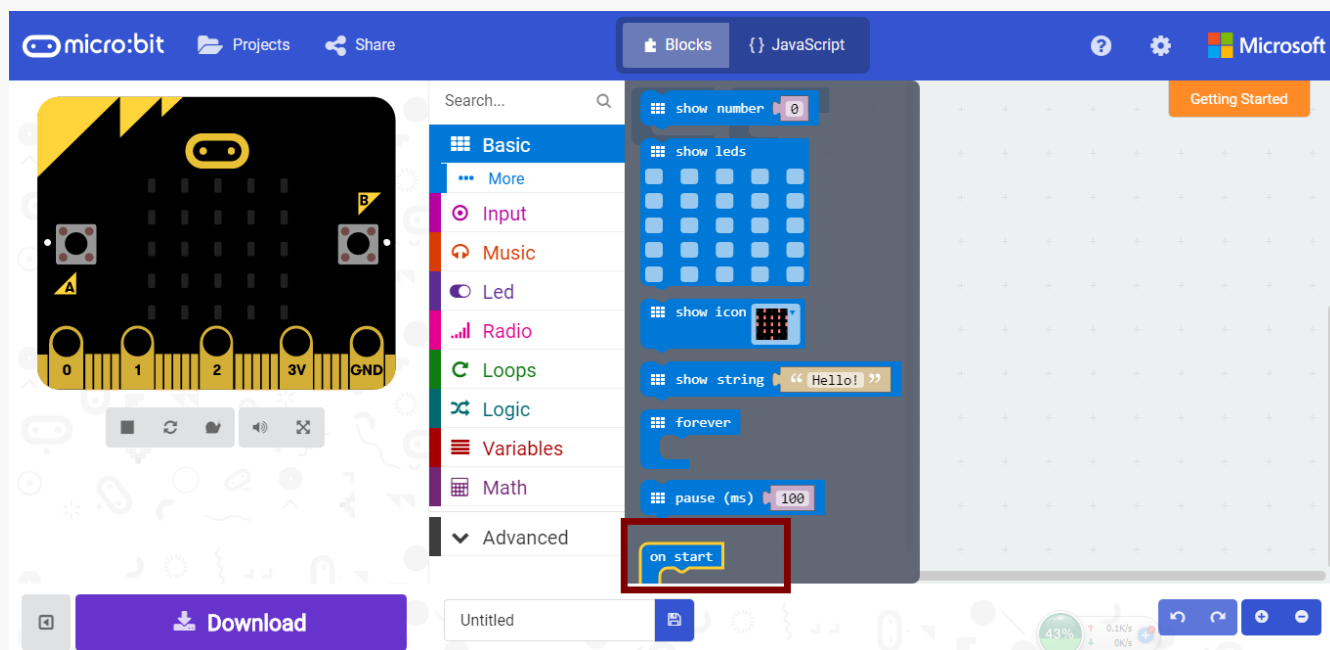
Abra MakeCode: <https://makecode.microbit.org/>.

PASO 1: Comience un nuevo proyecto. Puede volver al último capítulo si olvida cómo hacerlo.

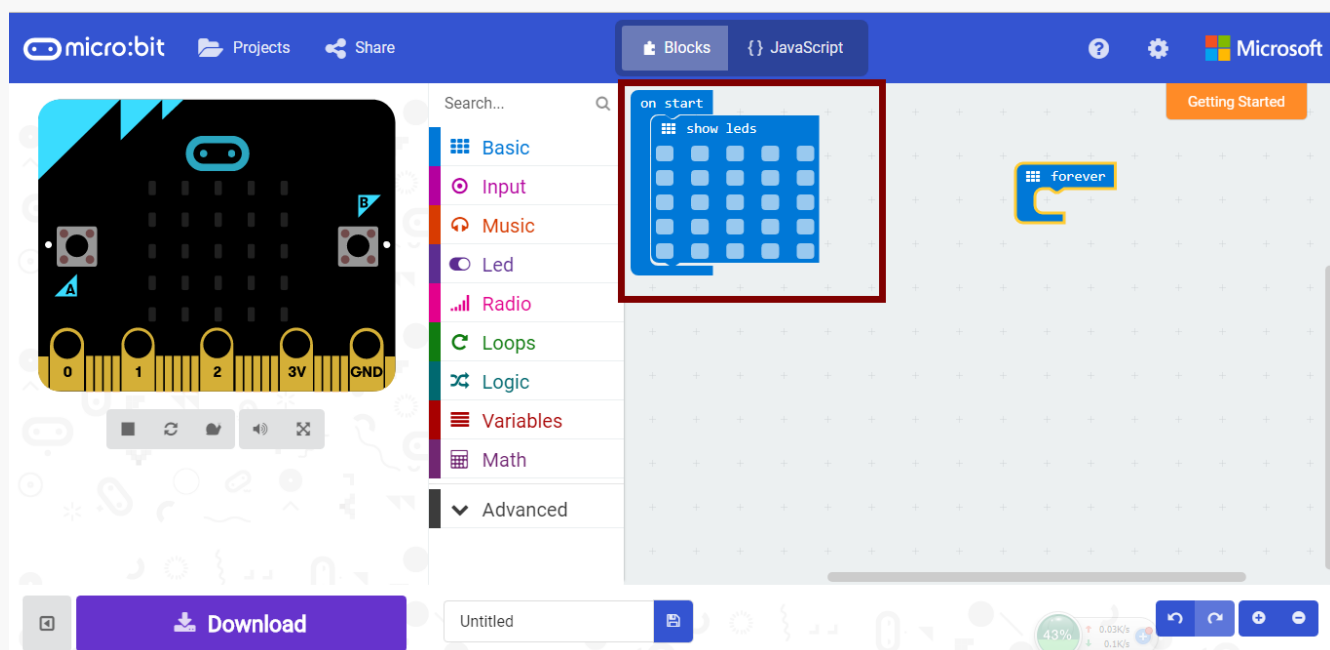
PASO 2: Haga clic en "Basic" en la parte superior del área funcional y encuentre la función "show leds".



PASO 3: El módulo "on start" aparecerá automáticamente en el área de programación después de abrir por primera vez la página web online de MakeCode, o puede arrastrar la función "on start" en la parte inferior de "Basic" hasta el área de programación.

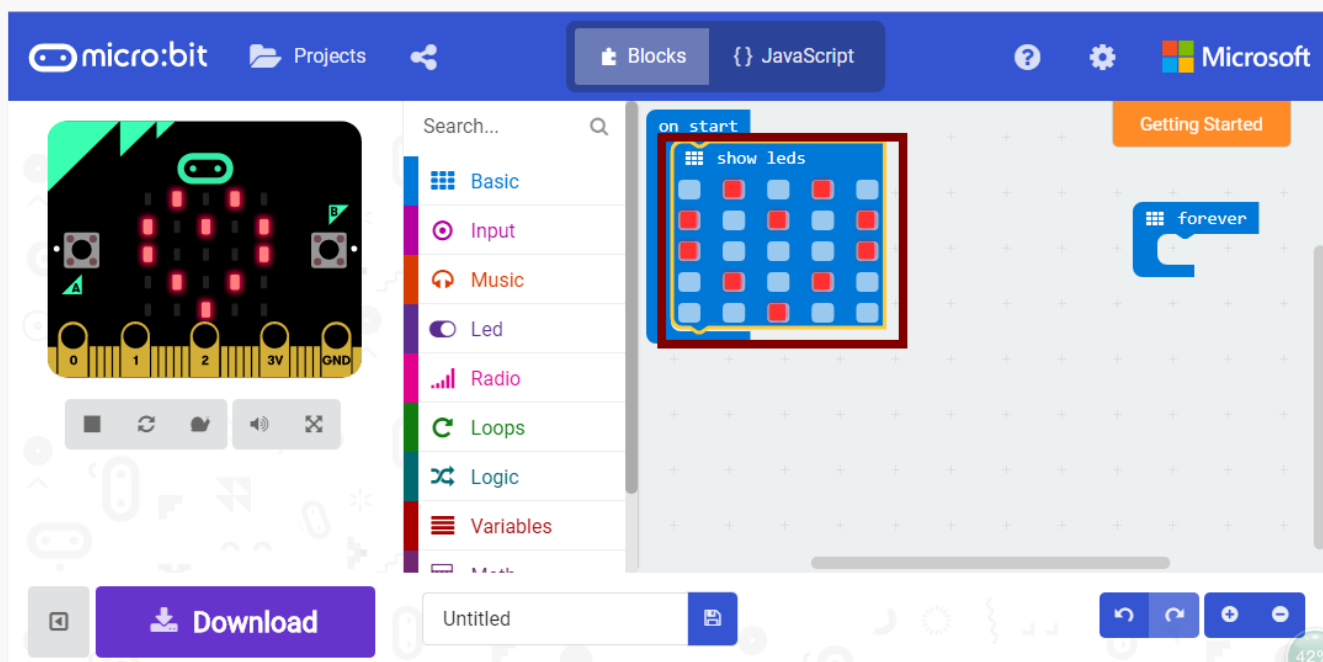


PASO 4: Arrastre la función "show leds" hasta el área de programación y colóquela en "on start".

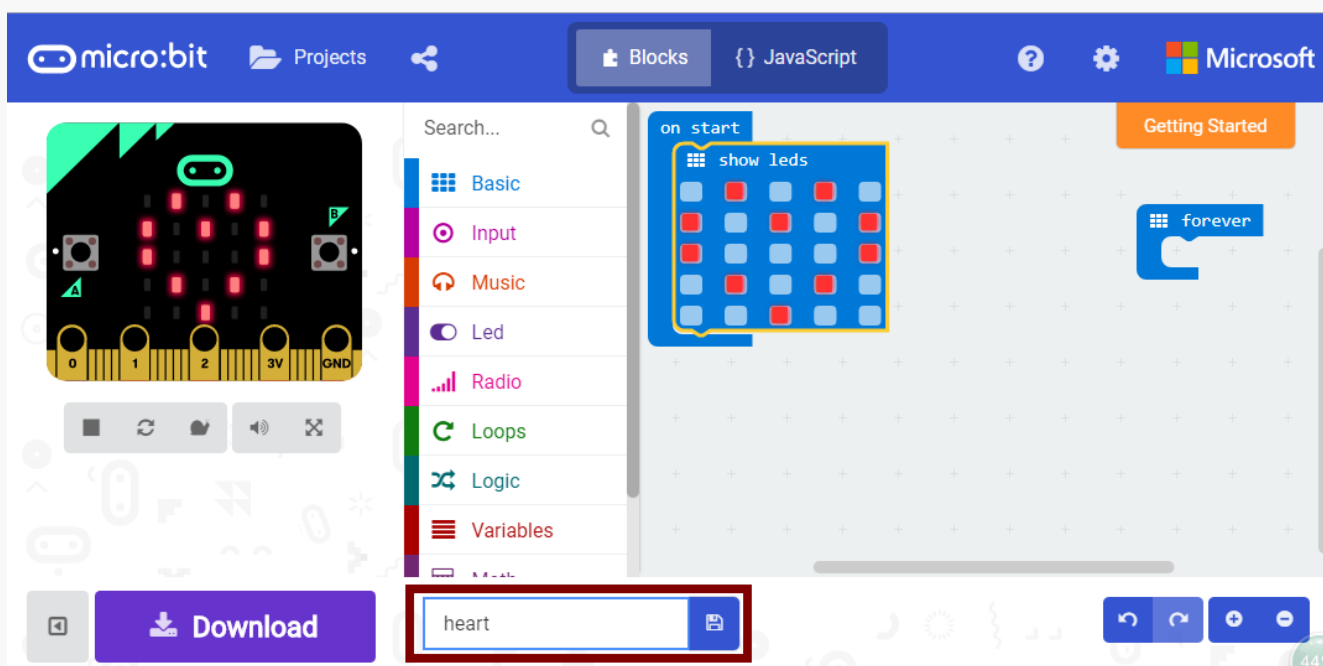


Cuando la función "show leds" está dentro del módulo "on start", puede hacer clic en el punto celeste dentro del bloque para editar el patrón. Además, todo lo que dibujemos dentro del bloque se mostrará en la ventana de simulación.

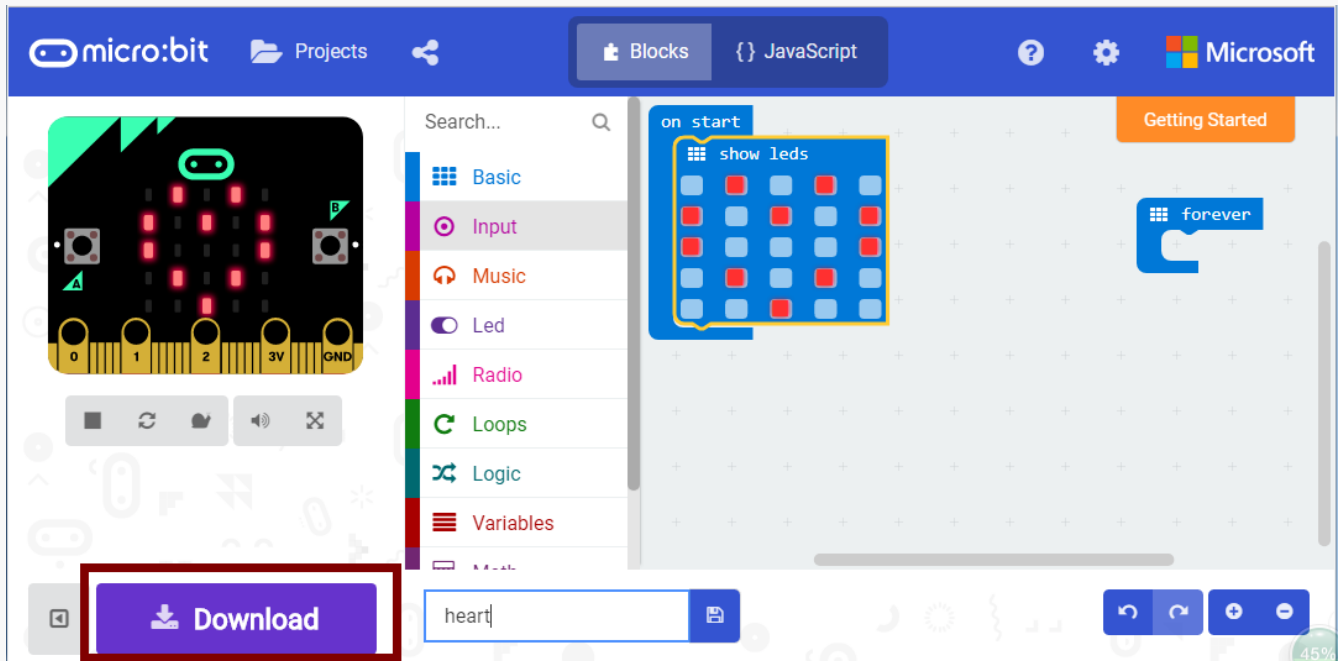




PASO 5: Aquí, dibujamos un corazón en el bloque. Para ayudarnos a recordar, podemos renombrarlo como "corazón".



PASO 6: Haga clic en "Download" para guardar el archivo en MICROBIT en el ordenador, o puede guardarlo directamente en Micro:bit. Si todo va bien, el Micro:bit mostrará entonces un emoji "corazón". (Si olvida cómo cargar el código, vuelva atrás y compruebe el PASO 4 del Capítulo 1).



## Ejercicios

Intente diseñar un emoji que pueda cambiar entre dos patrones diferentes.

## Proyecto 2: Luz LED intermitente

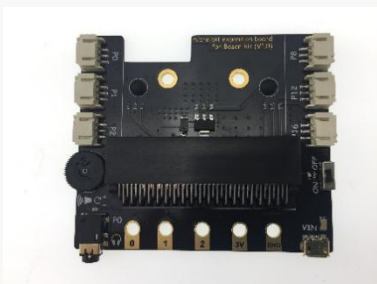
En la parte anterior, hemos aprendido a programar el panel de LEDs para hacer nuestros propios emojis. En este capítulo, conectaremos un módulo Led externo y aprenderemos cómo encenderlo y hacer que parpadee.

### Lista de componentes

1 × micro:bit



1 × Tarjeta de expansión Boson



1 × Modulo LED



1 × Cable USB

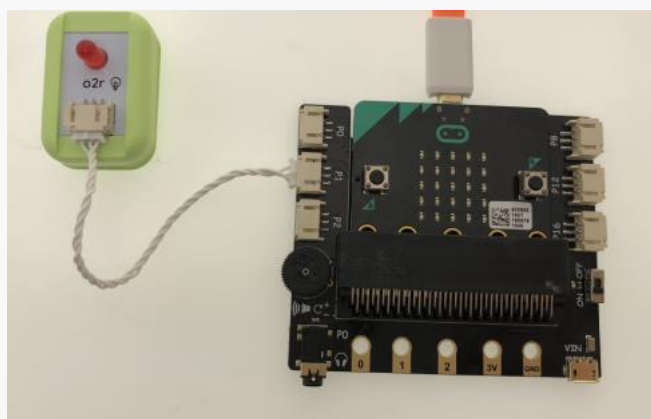
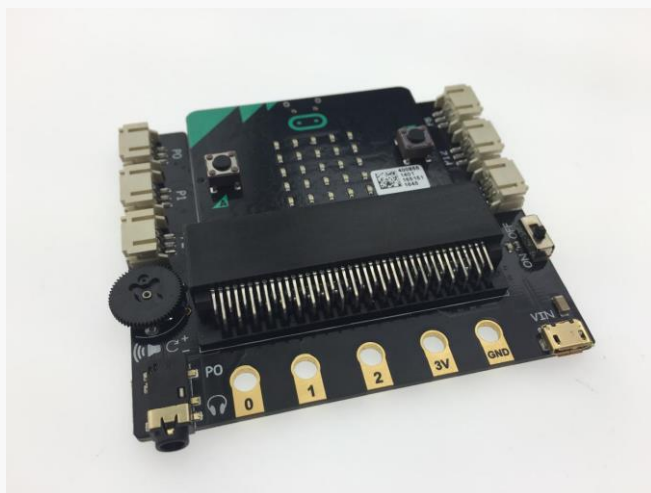


## Conexión

- Inserte el micro:bit en la tarjeta de expansión.

Aviso: micro:bit permanecerá dentro de la tarjeta de expansión en todos los capítulos siguientes. Saltaremos este paso en todos los capítulos siguientes.

- Conecte el módulo LED a P1

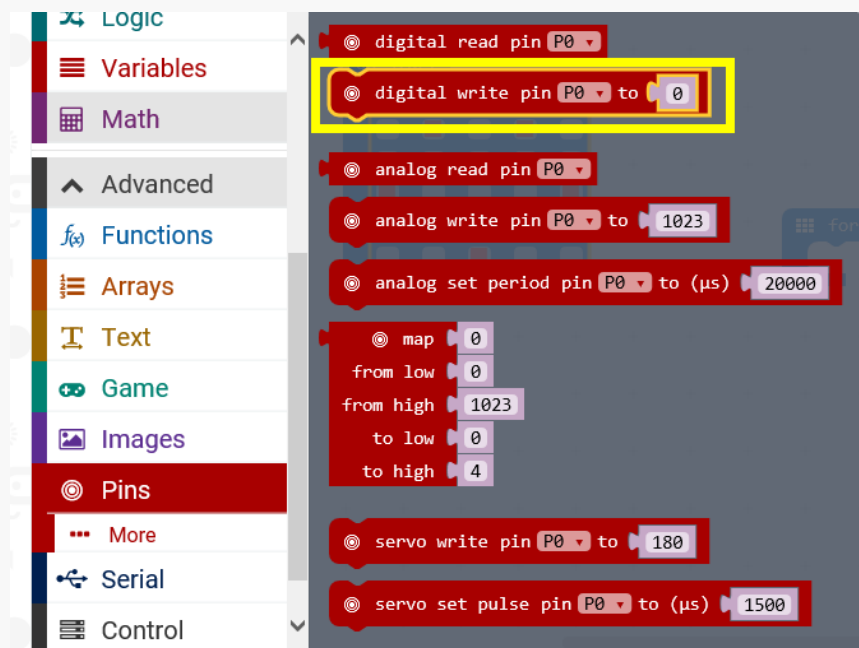


## Programa

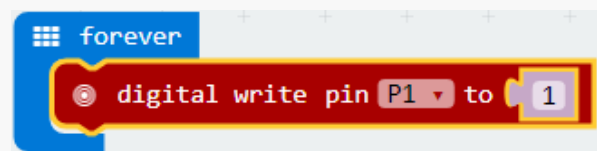
### Tarea 1: Encienda el LED

PASO 1: Abra el sitio web de MakeCode: <https://makecode.microbit.org/> y comience un nuevo proyecto.

PASO 2: Esta vez, ya que el módulo LED está conectado al P1 del Micro:bit como componente externo, para controlar el LED, tendremos que ajustar el estado de la clavija a la que se conecta el LED. El bloque de función "Digital write" es el que realiza el trabajo. "Digital write" puede hacer que la salida de pines sea de alto voltaje (presentado como 1), o de bajo voltaje (presentado como 0), que a su vez enciende y apaga el LED. Primero, encendemos el LED. Para ello, vaya a "Advance" -> "Pins", y ajuste la clavija de escritura digital p1 a HIGH (1).



PASO 3: Ponga la función "Digital write" en bucle "forever" (el bucle "forever" aparecerá en el área una vez que cree un nuevo programa, también se encuentra en "Basic") y el LED conectado a P1 se iluminará. El acabado será el siguiente:

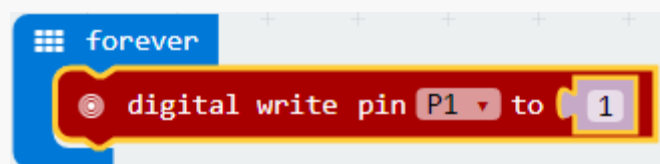


## Tarea 2: LED intermitente

Objetivo: Después de saber cómo encender un LED, daremos un paso más para aprender a hacer que parpadee.

PASO 1: Comience un nuevo proyecto. Siempre puede volver al PASO 3 del capítulo 1 si olvida cómo hacerlo.

PASO 2: Configure el estado de las clavijas del Micro:bit para encender y apagar el LED. Los LEDs de este experimento pertenecen a la salida digital. Haga clic en "Advance"->"Pins", ajuste "digital write pin p1" a (0). El valor del pin sólo puede ser 0 o 1. Cuando es 1, la luz permanece encendida. Cuando es 0, la luz permanece apagada.

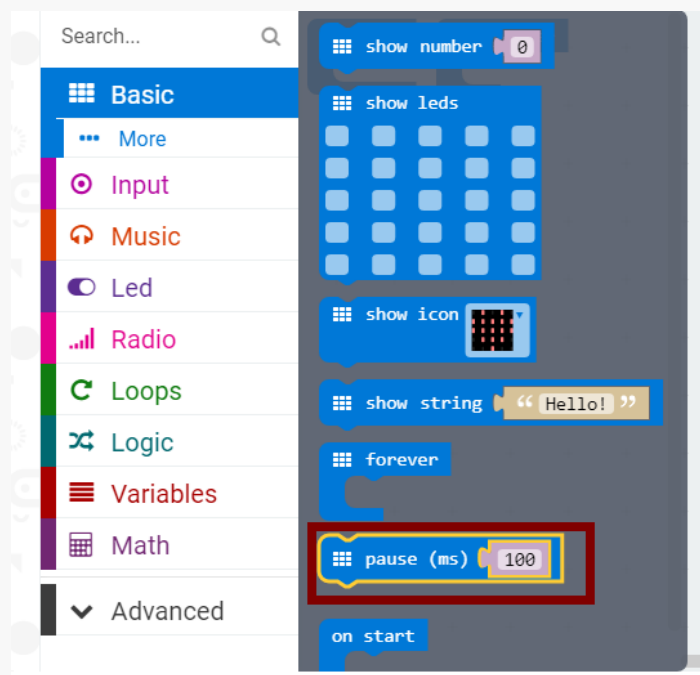


Ajustar el valor de la clavija P1 a "high" (1) y el LED se iluminará.



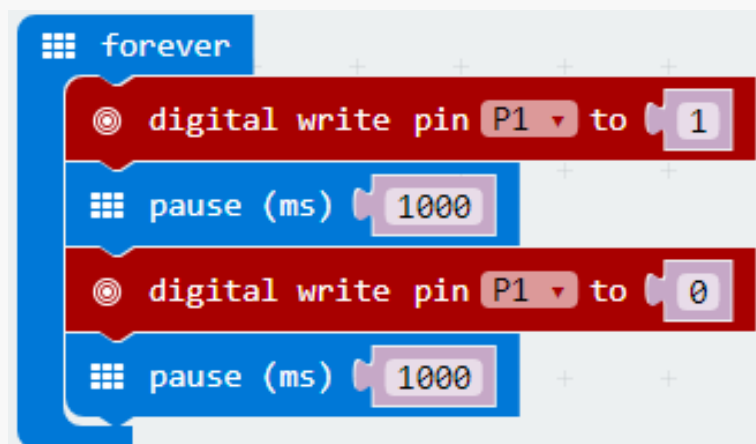
Ajustar el valor de la clavija P1 a "low" (0) y el LED se apagará.

PASO 3: También necesitará mantener el LED externo encendido por 1 segundo y apagado por otro segundo en una secuencia. Para ello, vaya a "Basic" y seleccione "Pause (ms) (100)". La función "Pause" mantiene el mismo estado durante un cierto periodo de tiempo.



PASO 4: Ponga la función "Pause" en el bucle "forever" y ajuste los números.

Combine todos los bloques de función listados arriba, tendremos el programa final como se muestra a continuación. En esta imagen: ajustar "pin P1" como 1 (LED encendido) y "pause(ms)" como 1000, pin P2 como 2 (apagado) y "pause(ms)" como 1000, y el programa dentro del bucle "forever" se ejecutará repetidamente en secuencia.



## Ejercicio

Puede que hayas oído la señal de emergencia SOS. ¿Por qué no invitar a tus compañeros de clase a aprender e intentar programarlo?

## Proyecto 3: Luz de notificación

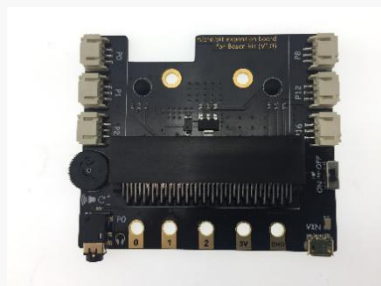
Como ha aprendido a encender un LED externo a través de micro:bit, ahora aplicaremos un módulo de botones externos para controlar el LED. Además, aprenderemos a ajustar el brillo de la luz usando una perilla.

### Lista de componentes

1 × micro:Bit



1 × Tarjeta de expansión BOSON



1 × Modulo LED



1 × Módulo Botón



1 × Modulo Perilla



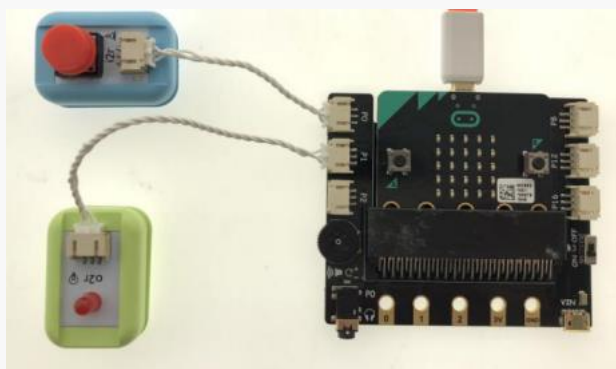
1 × Cable USB



## Conexión

Conectar el módulo de Botón a P0

Conectar el módulo LED a P1



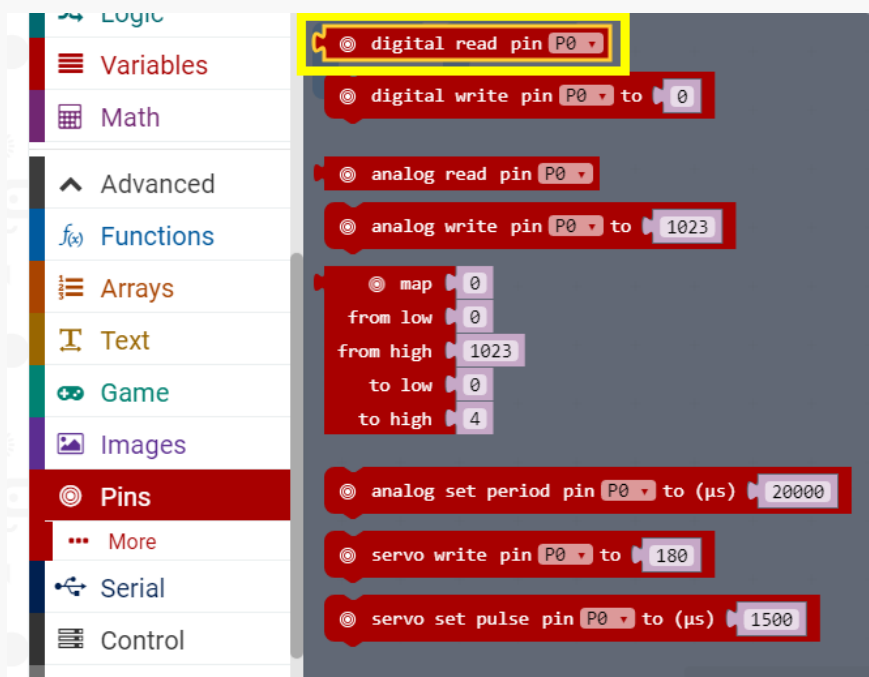
## Programa

### Tarea 1: Lámpara controlada por botones

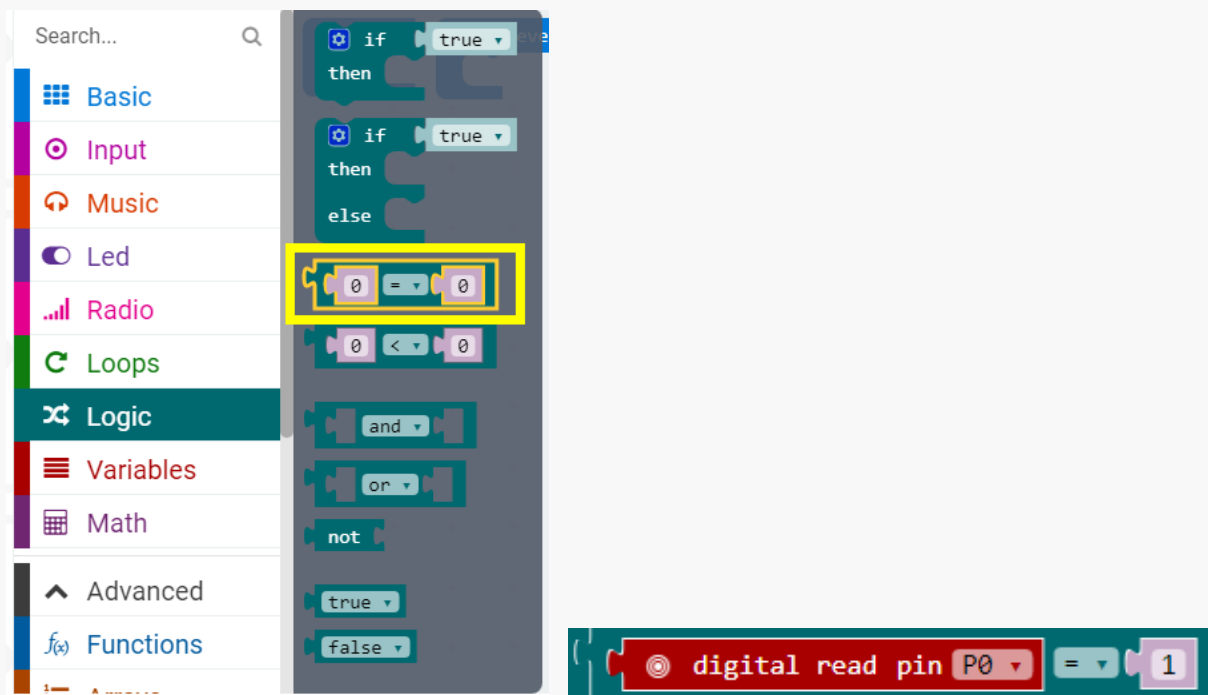
Objetivo: Cuando se pulsa el botón, el módulo LED se enciende. Cuando se suelta el botón, el LED se apaga.



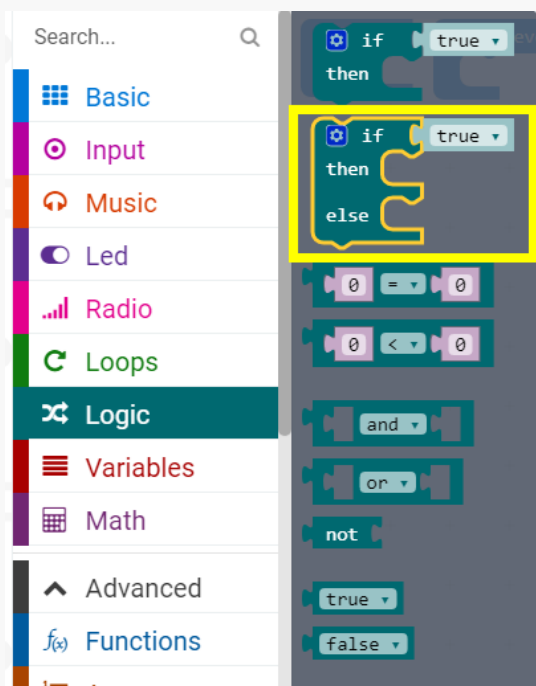
PASO 1: Conectar el módulo de botón a P0. Leyendo el valor de P0, Micro:bit obtendrá el estado del botón. Para vincular el botón con el LED, podemos escribir un programa sencillo Cuando se pulsa el botón Micro:bit recibe un "1", por lo que ajustaremos el valor del LED (pin 1) como 1 y el LED se encenderá. Cuando se suelte, ajuste el valor de la clavija a 0 y el LED se apagará. Este módulo de botones es un módulo de entrada digital y la función "digital read" se encuentra en "Advance" -> "Pins".



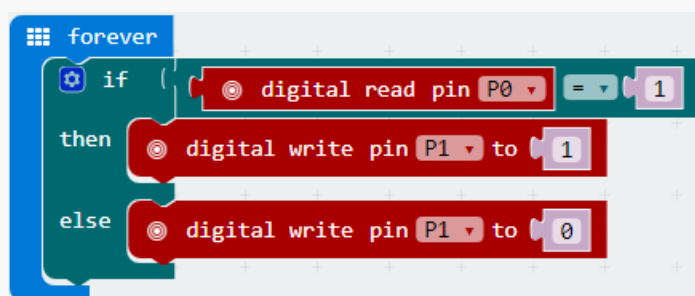
PASO 2: El operador de condición "=" bajo "Lógica" se utiliza para determinar si se pulsa o no el botón. Si el estado del botón es igual a 1, entonces sabremos que el botón está siendo presionado y viceversa.



PASO 3: Si se pulsa el botón, el LED externo se ilumina; en caso contrario, permanece apagado. Utilizaremos otro operador de condiciones, la función "if-else". La función también está en "Logic"



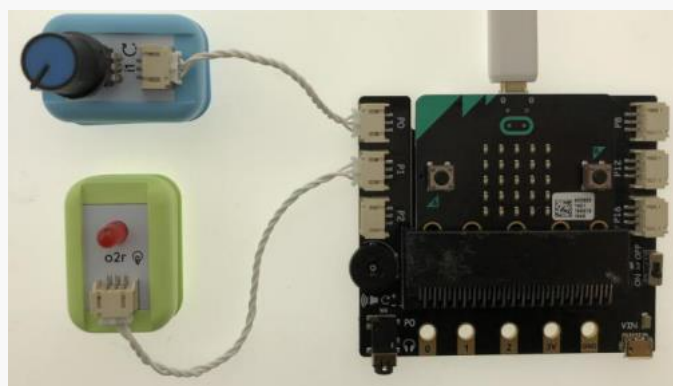
PASO 4: Al juntar todas las funciones mencionadas anteriormente, tendremos el siguiente programa. El programa detecta el estado del botón y enciende el LED cuando se pulsa el botón.



## Tarea 2; LED controlado por perilla

Objetivo: en la pieza, no sólo controlaremos el LED de encendido a apagado, sino que también utilizaremos la perilla para cambiar su brillo.

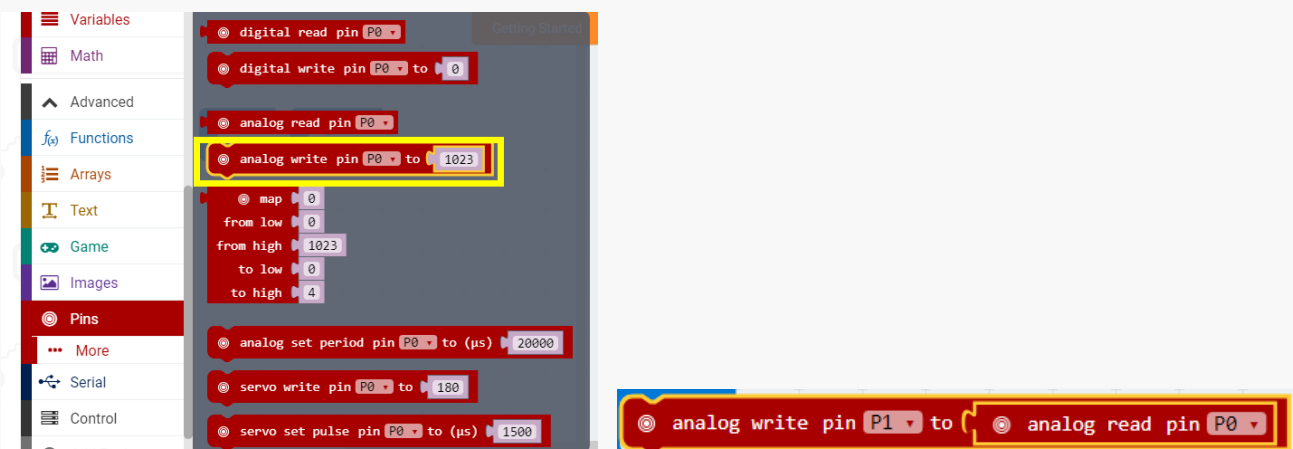
PASO 1: Conecte el circuito como se indica a continuación. El módulo de botones utilizado en la tarea 1 se sustituye ahora por el módulo de perilla.



PASO 2: El brillo del LED está vinculado al valor de entrada de la perilla. Esta es un módulo de entrada analógica, lo que significa que genera un rango de señales de 0 - 1023, mediante el cual se controla correspondientemente la luminosidad del LED. La función "lectura analógica" se encuentra en "Pins". Además, recuerde ajustar el pin a P0.



PASO 3: El LED se utiliza ahora como módulo de salida analógica. Del mismo modo, la señal de salida analógica también está comprendida entre 0-1023. "La función "Analog write" se encuentra en "Pins".



PASO 4: Ahora, todo lo que necesitamos es conectar la entrada directamente a la salida para que el LED esté bajo control del mando. Ponga la función de "analog read" dentro de la función de "analog write", luego póngalos todos dentro del bucle "forever" y listo. El programa final se muestra a continuación.



## Ejercicio:

Haga que su LED se parezca más a una lámpara real!

## Proyecto 4: Ventilador eléctrico

En esta parte, aprenderemos a construir un ventilador eléctrico que funcione a diferentes velocidades.

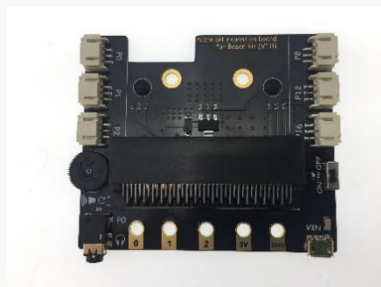
Antes de empezar, tendremos una rápida revisión de cómo encender o apagar el ventilador con el módulo de botones. A continuación, aprenderemos a controlar la velocidad de un ventilador con el módulo de perilla.

### Lista de componentes

1 × micro:bit



1 × Tarjeta de expansión BOSON



1 × Modulo ventilador



1 × Módulo botón



1 × Modulo perilla

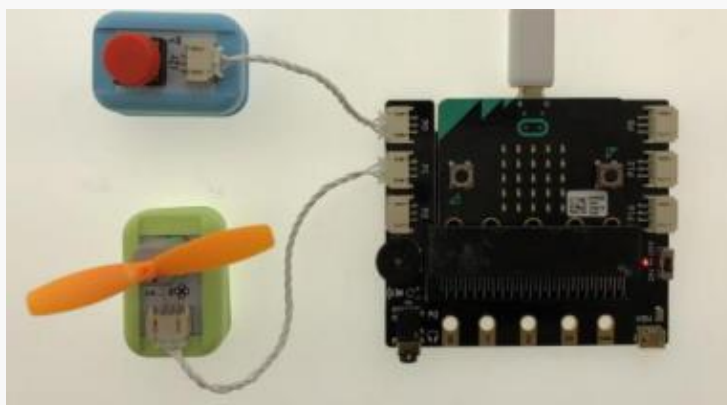


1 × Cable USB



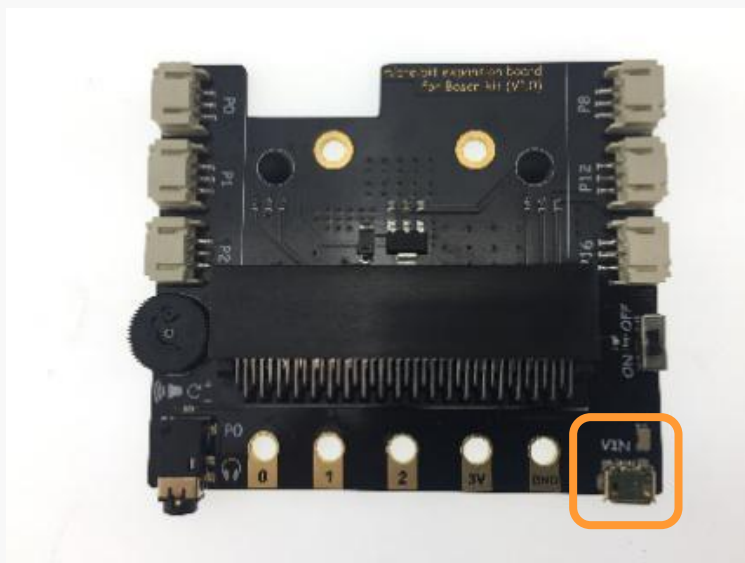
## Conexión

Conecte el módulo de botón (módulo de perilla en la segunda parte) al P0.  
Conectar el módulo de ventilador al P1.



## ¡El ventilador eléctrico "devora" la energía!

Al conectar los módulos Boson, debemos tener en cuenta que, debido al consumo de energía de módulos como ventiladores o servos, será necesaria una fuente de alimentación externa. Por lo tanto, primero usaremos el cable USB para programar el Micro:bit. Cuando terminemos, cambiaremos el cable USB a la fuente de alimentación externa de la tarjeta de expansión.



## Programa

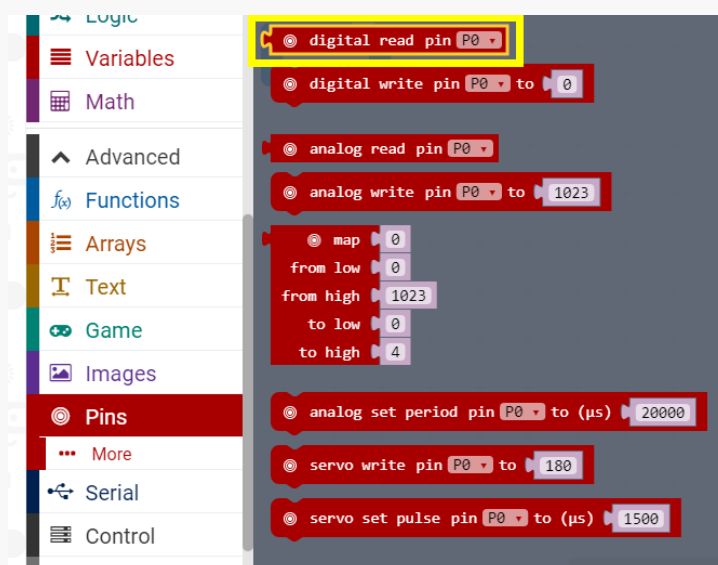
### Tarea 1: Control el ventilador con el módulo de botón

Objetivo: cuando se presiona el botón, el ventilador comienza a funcionar; cuando se suelta, el ventilador se detiene.

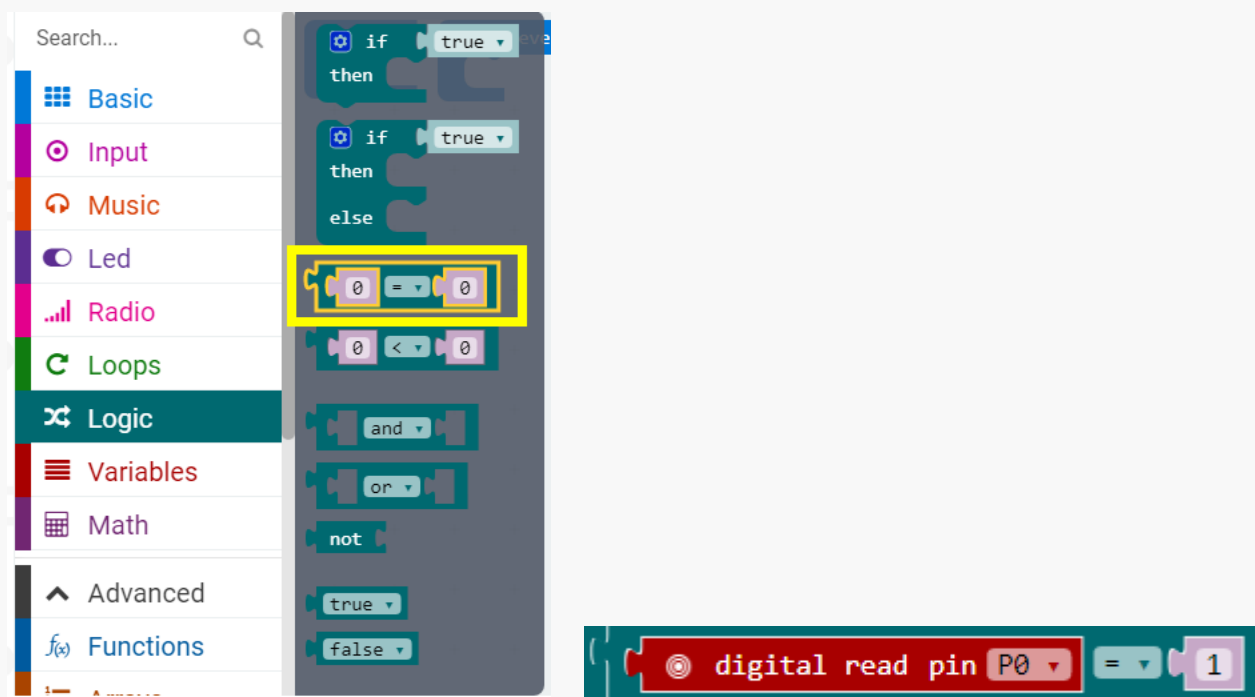
PASO 1: El módulo de botón se conecta a P0, lo que significa que tendremos que utilizar la función de "digital read" en P0 para detectar primero el estado del botón.

El módulo del ventilador se conecta a P1. Asimismo, para controlar el ventilador, utilizaremos la función de "digital write" en P1 para conmutarlo entre ON y OFF.

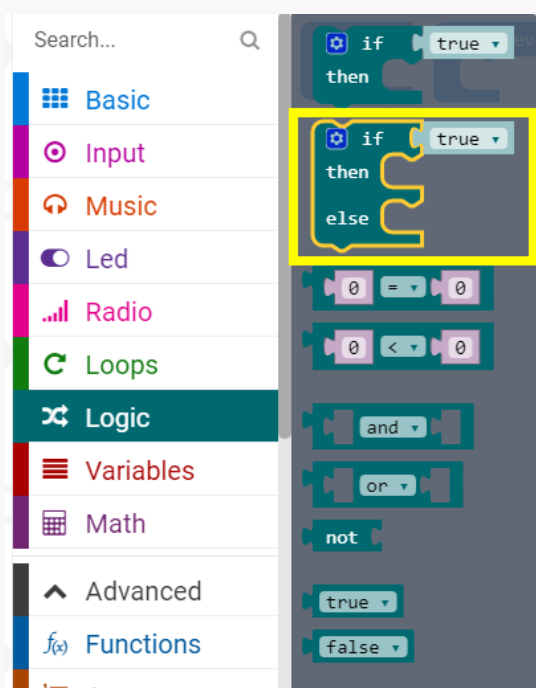
Las funciones de "digital read" y "digital write" se encuentran en "Pins".



PASO 2: El operador de condición "=" bajo "Lógica" se utiliza para detectar si se pulsa o no el botón. Pulsando el botón, la función de "digital read" devuelve "1", que cumple la condición de "1" que pre programamos al principio, por lo que se ejecutará el programa bajo la sentencia. Sin embargo, cuando devuelve "0", se omitirá el programa bajo la sentencia.

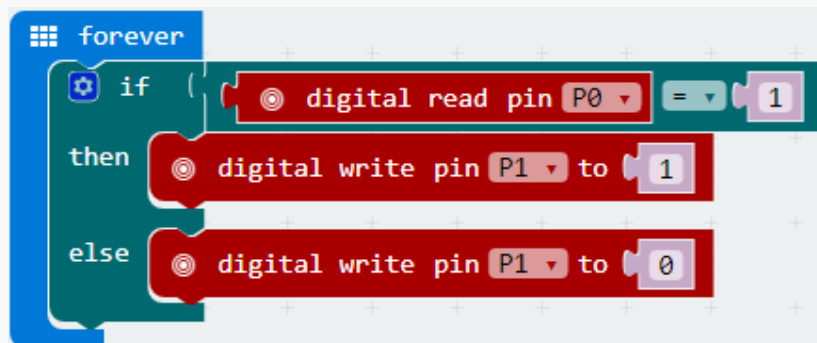


PASO 3: Si se presiona el botón, el ventilador se encenderá; de lo contrario, permanecerá apagado. La función "if-else" bajo "logic" nos ayudará a hacer la elección en función del estado del botón.





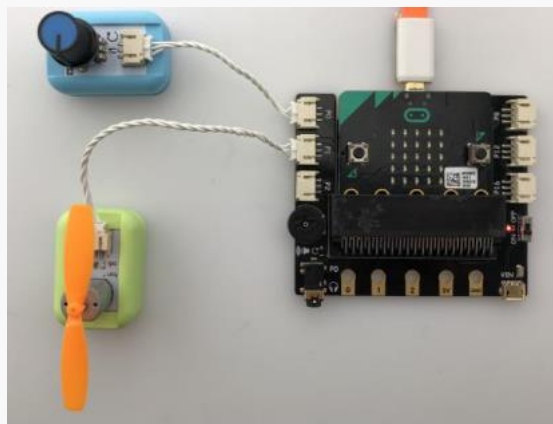
PASO 4: Al juntar los bloques de función mencionados anteriormente, debemos terminar con un programa con la siguiente función: cuando se presiona el botón, se enciende el ventilador, cuando se suelta, se apaga el ventilador.



## Tarea 2: Ajustar el ventilador a una velocidad diferente

Objetivo: En esta tarea, aprenderemos a usar un módulo de perilla para controlar la velocidad del ventilador. Cuando se alcanza un valor de entrada analógica mayor, cuanto más rápida es la velocidad a la que funciona el ventilador, y viceversa. La velocidad del ventilador cambia continuamente cuando el mando gira.

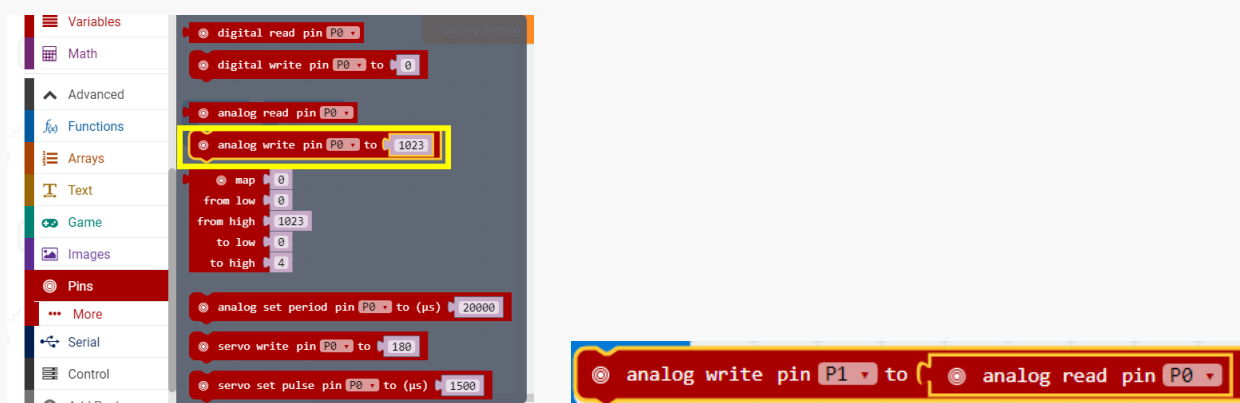
PASO 1: Conecte el circuito como se indica a continuación. El módulo de botón utilizado en la tarea 1 debe ser sustituido por un módulo de perilla.



PASO 2: Volveremos a utilizar el módulo de ventilador. Para hacer la velocidad del ventilador controlada por la perilla, necesitamos construir un enlace entre la velocidad y el valor de entrada de la perilla. La perilla es un módulo de entrada analógica, lo que significa que genera una señal que oscila entre 0 y 1023, mediante la cual se controla la velocidad del ventilador. Para leer la señal, conecte el módulo de perilla a P0, utilice la función "analog read" bajo "Pins"



PASO 3: El módulo de ventilador se utiliza ahora como un módulo de salida analógica, y su valor también oscila entre 0-1023. La función "analog write" se encuentra bajo "Pins".



STEP4: Poner la función de "analog read" en "analog write", y luego colocarlas todas en el bucle "forever". El programa final es el siguiente :



## Capítulo 4: Un poco más allá

Arriba hemos aprendido a usar el módulo de botón y el módulo de perilla para controlar un ventilador eléctrico. Pero todavía hay un largo camino por recorrer, usted puede estar ansioso por explorar más sobre el micro: bit. En este capítulo, se introducirán más sensores para que su dispositivo sea más inteligente y funcional.

### Proyecto 1: Vela Electrónica

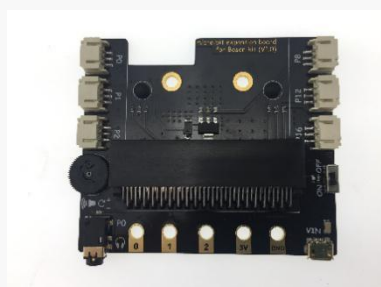
En este proyecto, vamos a hacer un dispositivo que simula una vela para reducir las emisiones de carbono. Además, para que se parezca más a uno real, el fuego debería poder apagarse soplando aire sobre él. ¡Hagámoslo posible!

#### Lista de componentes

1 × micro:bit



1 × Tarjeta de expansión BOSON



1 × Modulo LED



1 × Modulo Sensor de Sonido

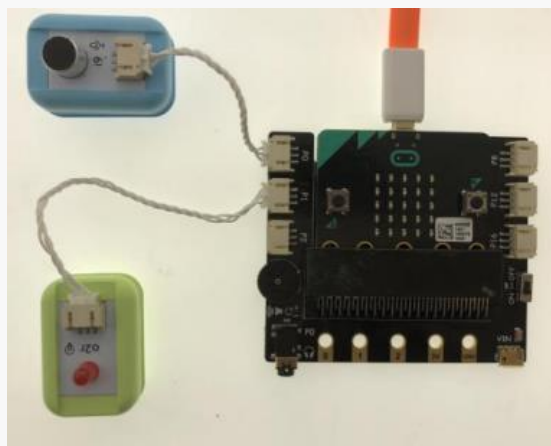


1 × Cable USB



## Conexión

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



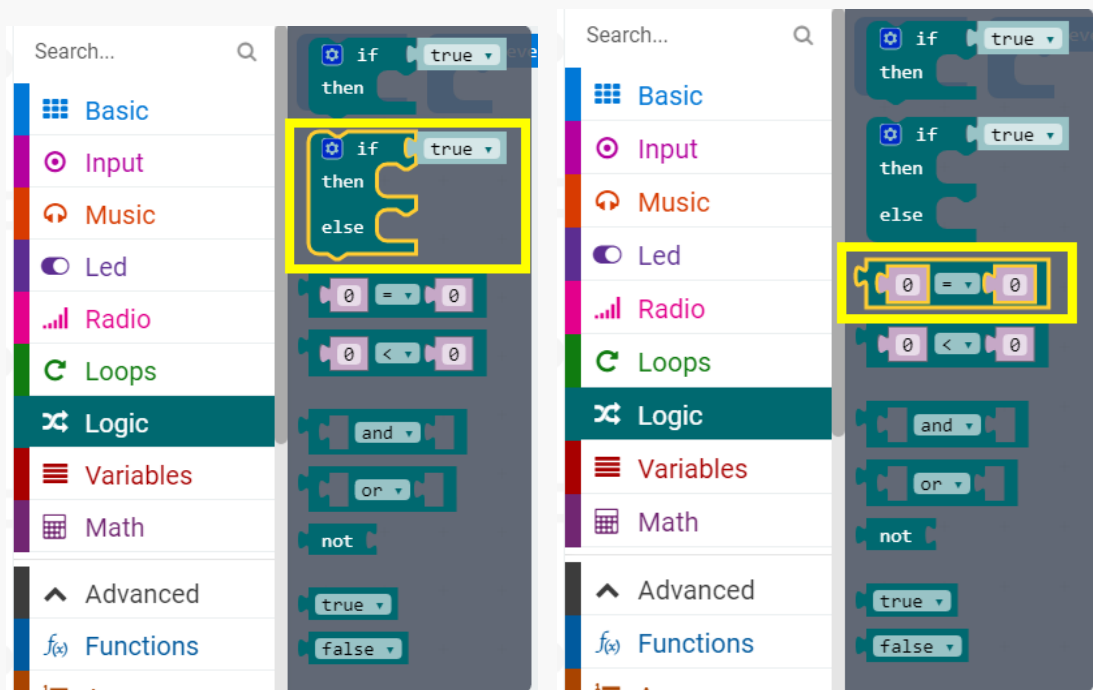
## Programa

Para construir una vela electrónica, primero debemos saber cómo funciona el sensor de sonido. Podemos conectar el sensor de sonido al Micro:bit y ver cuán fuerte es nuestro sonido mirando el número que aparece en el panel LED.

**Tarea 1: Medir el volumen**

Objetivo: Cuando el valor analógico del sensor de sonido es inferior a 50, el LED de la placa muestra un número "1", indicando que el volumen es bajo; cuando es superior a 50, el LED de la placa muestra un número "2", indicando que el volumen es alto.

PASO 1: Dejaremos el módulo LED a un lado por un momento y nos centraremos en la parte lógica. En esta parte se aplicarán la función "if else" y el operador "=". Ambos deben ser colocados en bucle "forever". La función "if else" se puede utilizar para juzgar qué número se debe mostrar. Lo que significa que, si se cumple la sentencia lógica de la parte derecha, debe ejecutarse la frase clave a la derecha de "then", mientras que la frase a la derecha de "else" debe ejecutarse si no se cumple la sentencia lógica.



PASO 2: El sensor de sonido está conectado a P0, lo que significa que utilizaremos la función "analog read" para obtener la lectura de P0. Como se ha explicado anteriormente, la señal analógica va de 0 a 1023.



PASO 3: Ahora, tendremos que dejar que el programa determine el nivel de la voz. El operador "<" bajo "Logic" hace el trabajo. Como se muestra en la imagen siguiente, Cuando el valor analógico del sensor de sonido es inferior a 50, los LEDs muestran un número "1", indicando que el volumen es bajo; cuando es superior a 50, el LEDs muestran un número "2", indicando que el volumen es alto.



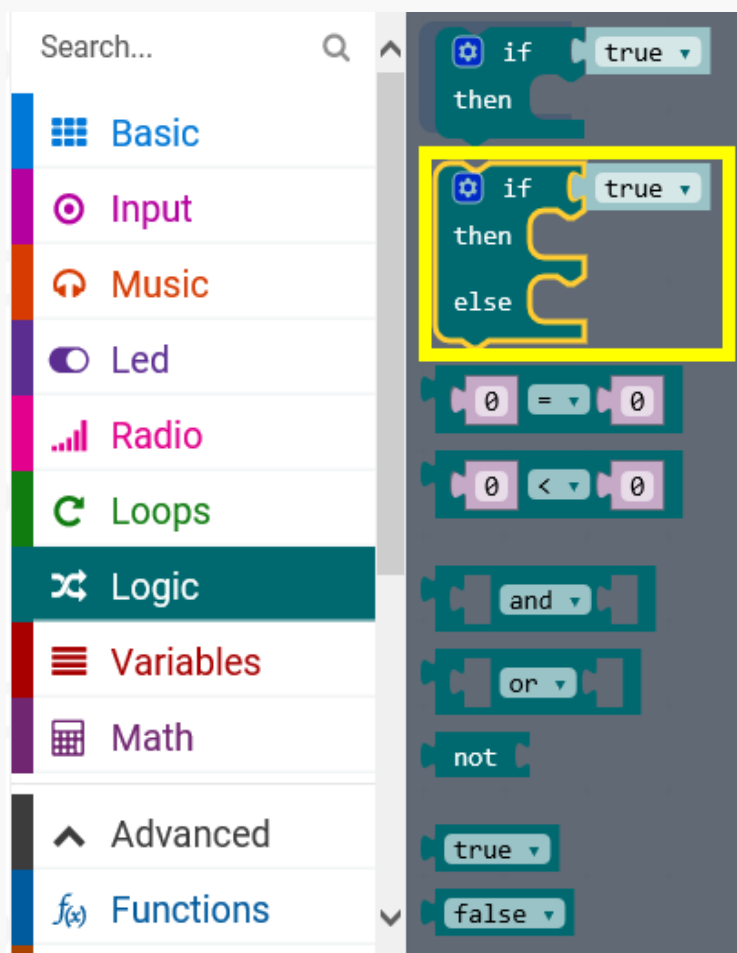
Haga algún ruido alrededor del sensor, o simplemente sople un poco de aire para probar si el Micro:bit es capaz de mostrar un número diferente con respecto al volumen.

Además, si el sensor está siendo demasiado sensible, puede ser necesario cambiar el "50" después del operador "<". También funciona de la misma manera si el sensor no responde.

## Tarea 2: Vela electrónica

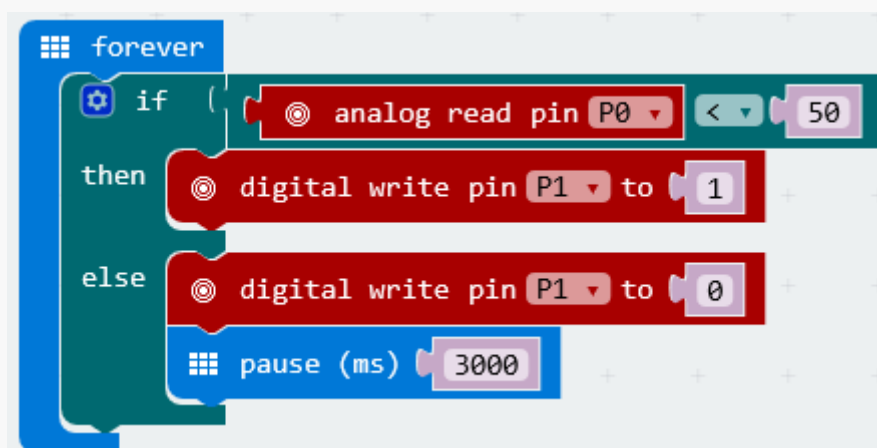
Objetivo: El Micro:bit puede medir el volumen usando el sensor de sonido, en base a eso, queremos usarlo para controlar la vela. Esto es lo que vamos a programar: una vez que soplamos aire hacia el sensor, la vela se apaga por un segundo.

PASO 1: Volveremos a utilizar la función "if else" en nuestro programa. Sin embargo, en lugar de controlar un panel de LED, sólo tenemos que controlar el LED que se conecta a P2. La lógica es muy sencilla: una vez que el volumen supera el valor preestablecido, la luz se enciende y viceversa. Ya que queremos que la vela siga encendida, el programa debe ser puesto dentro del bucle "forever".



PASO 2: Sin embargo, jeso no es suficiente! Cuando soplamos aire al sensor, el volumen sólo excederá el valor preestablecido por un tiempo muy corto. A veces sucede tan rápido que no seremos capaces de notar el cambio. Para solucionar este problema, necesitaremos añadir una función "Pause" para extender la duración. Digamos que si la duración ajustada es 3000 (3000ms = 3 segundos), una vez que soplamos la vela, la luz se apagará durante 3 segundos, y luego volverá al principio del bucle "forever" y se volverá a encender.

Así es como se ve el programa.



## Proyecto 2: Puerta automática

Las puertas automáticas se encuentran muy a menudo en nuestra vida cotidiana. Las puertas automáticas se encuentran muy a menudo en nuestra vida cotidiana. Cuando estamos a punto de entrar en una tienda, la puerta se abre automáticamente y a veces junto con el sonido "welcome~".

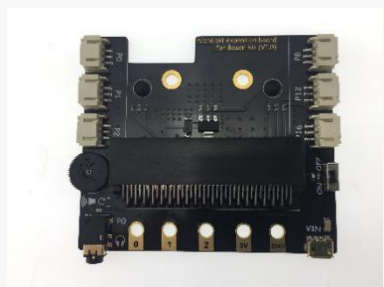
En esta lección, vamos a aprender a construir una puerta automática usando un sensor de movimiento, que detecta si alguien se acerca, y un servo para simular la acción de abrir la puerta. ¡Ahora comencemos!

### Lista de Componentes

1 × Micro:bit



1 × Tarjeta de expansión BOSON



1 × Módulo Botón



1 × Modulo Sensor de movimiento





1 × Modulo Servo



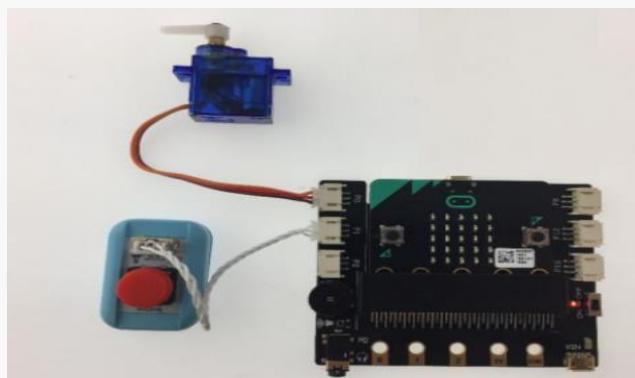
1 × Cable USB



## Conexión

Conecta el servo a P0

Conectar el módulo de botón (sensor de movimiento en la segunda parte) a P1

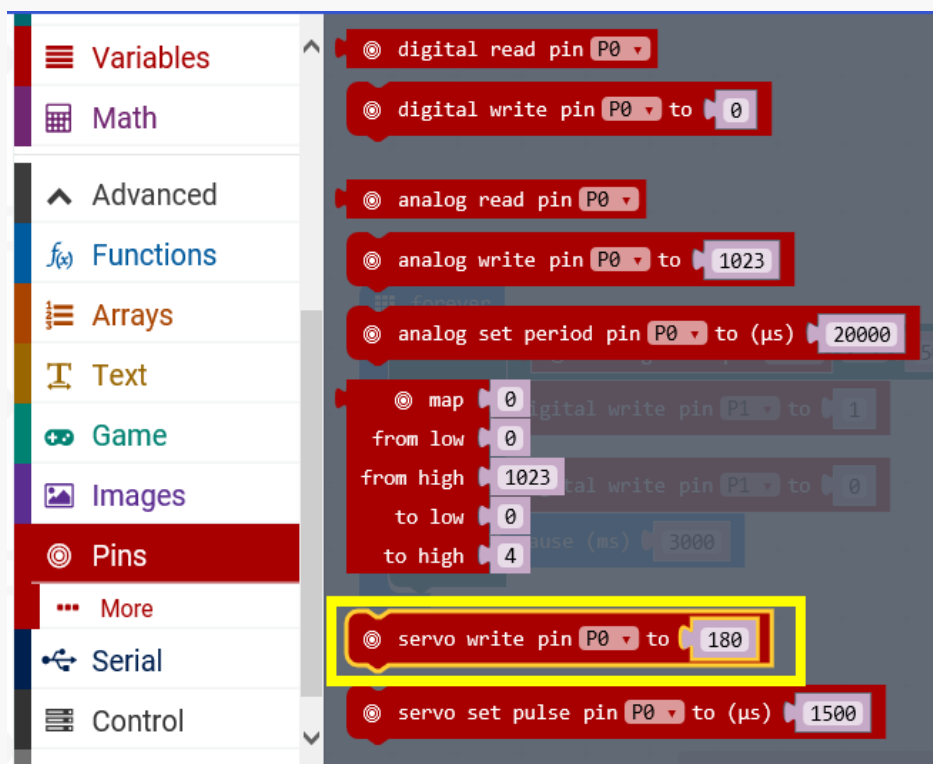


## Programa

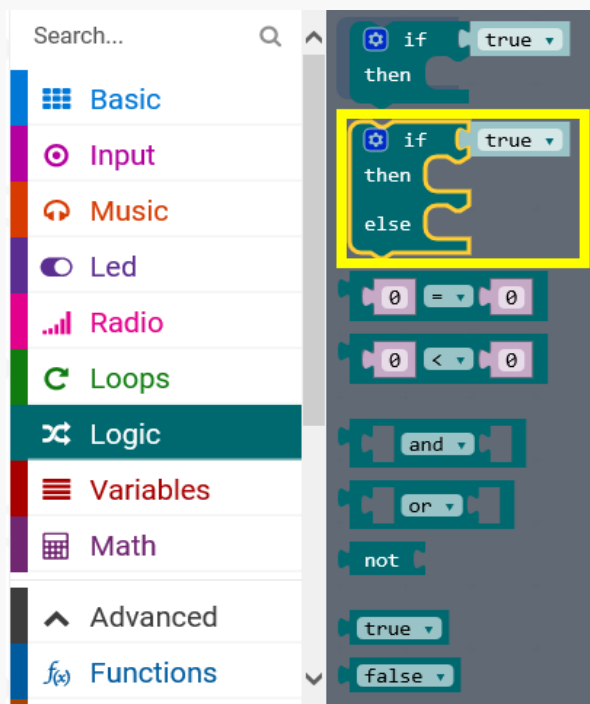
Tarea 1: Utilice el módulo de botón para controlar el servo

Objetivo: Cuando se presiona el botón, el servo gira a 100°, y el panel LED del Micro:bit muestra "O"; cuando se suelta, el servo permanece en el 0°, y el panel LED del Micro:bit muestra "X".

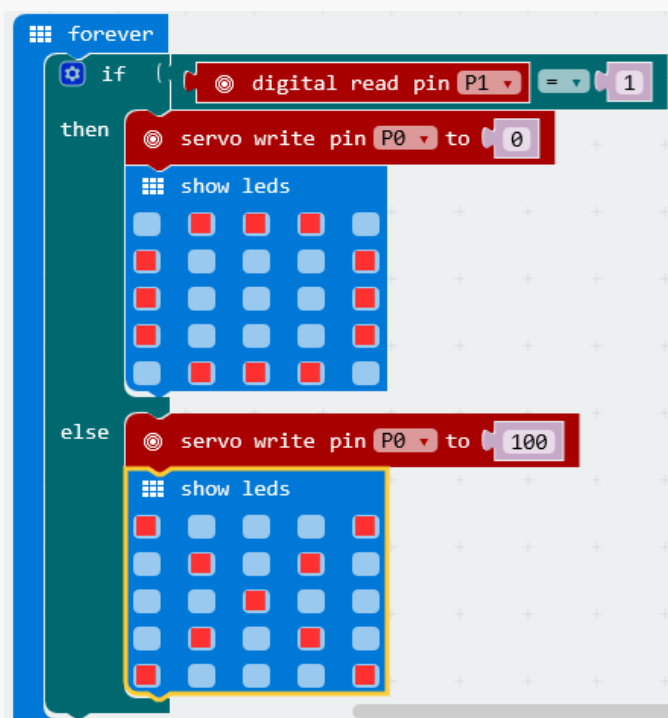
PASO 1: La función "servo write" bajo "Pins" se usa para controlar el servo. Como el servo está conectado a P0, dejaremos el "P0" tal cual. El número "180" es el ángulo que queremos que alcance el servo. Cambiando este valor podemos ajustar el servo para que gire entre 0°~180°.



PASO 2: Ahora, usa la función "if-else" para determinar la posición del servo basado en el estado del botón. Cuando se presiona el botón, el servo va a 100°, de lo contrario se mantiene a 0°. Además, no olvide añadir la "O" y la "X" para indicar el estado del botón.



PASO 3: Al juntar todo, terminaremos con el programa que se muestra a continuación.

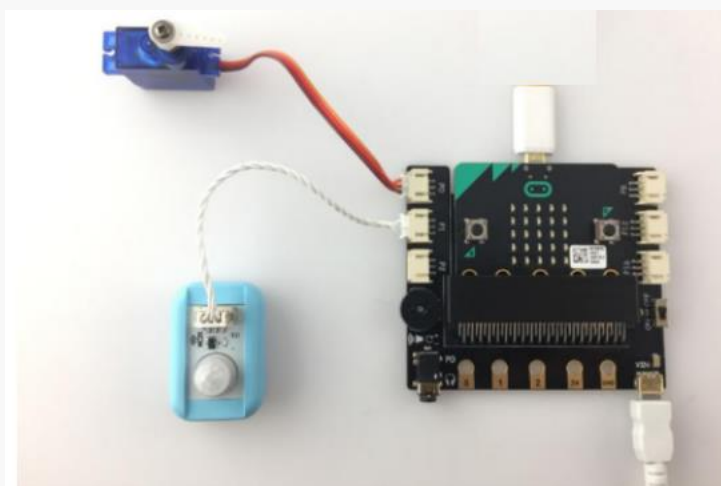


Hay que tener en cuenta que el servo tiene un consumo de energía relativamente grande, por lo que tendremos que conectarlo a la fuente de alimentación a través del puerto de alimentación USB externo.

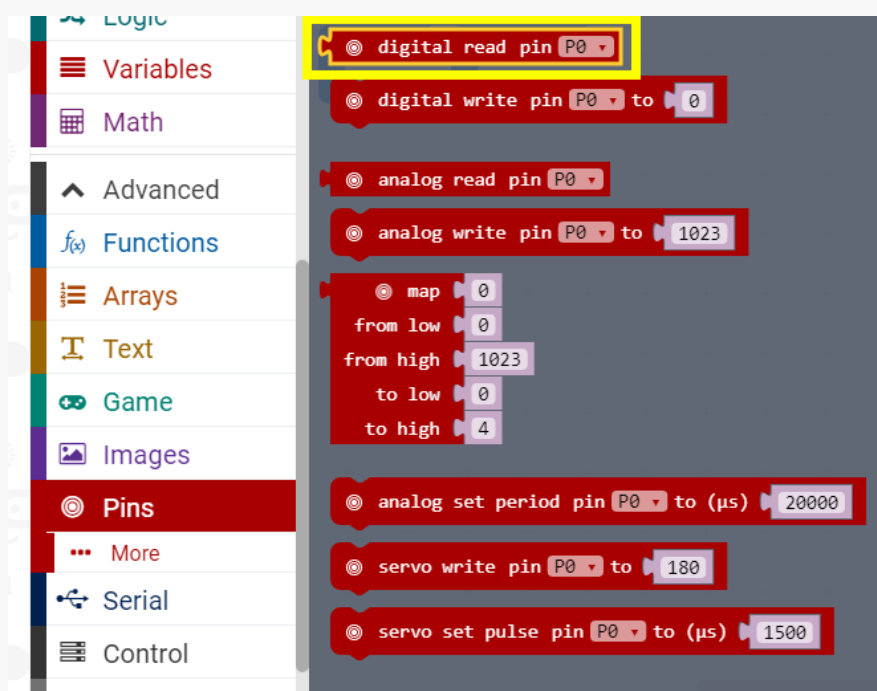
## Tarea 2: Puerta con control de movimiento

Objetivo: reemplazaremos el botón por un sensor de movimiento para controlar el servo. Cuando alguien se acerca, la puerta se abre. De lo contrario, permanecerá cerrada.

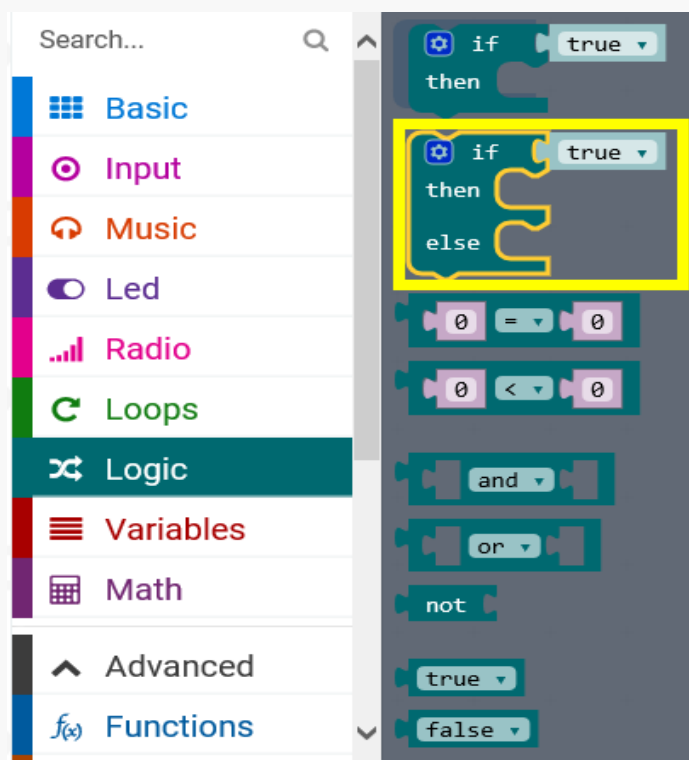
PASO 1: Sustituya el módulo de botón por el sensor de movimiento.



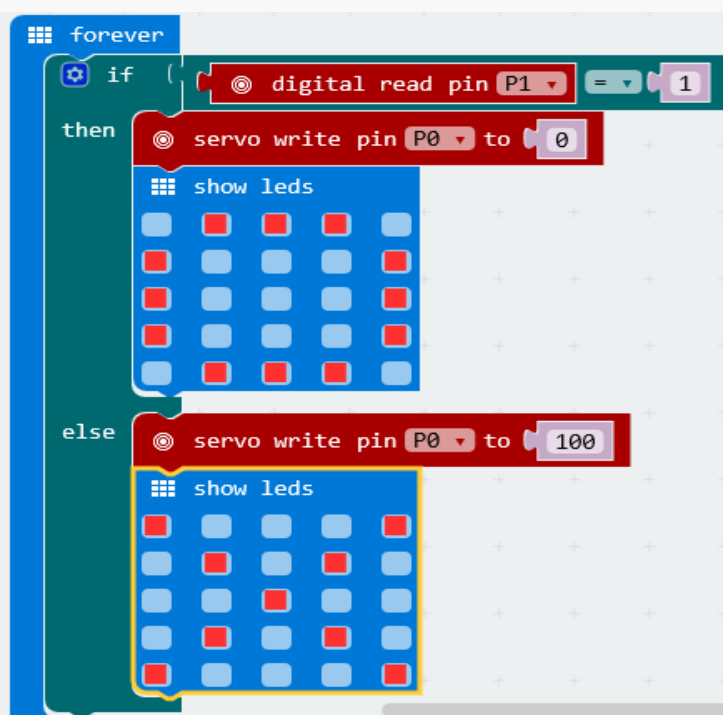
PASO 2: El sensor de movimiento está conectado a P1. Así, podemos leer el valor del sensor de movimiento. La función utilizada en esta parte es: "Pins"=>"digital read". Recuerde reemplazar el valor por defecto P0 por P1.



PASO 3: Cuando alguien es detectado por el sensor de movimiento, el servo gira a 100°, el Micro:bit muestra "O"; de lo contrario, el servo gira a 0° y el Micro:bit muestra "X". El bloque de función utilizado en este paso es "if else".



PASO 4: Poniendo todas las funciones mencionadas anteriormente, tendremos el programa final de la siguiente manera:



## Proyecto 3: Caja musical

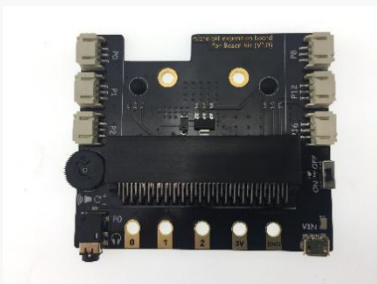
Varias cajas musicales llamativas de diferentes formas siempre han sido populares en las tiendas de regalos. Usted debe haber sido tocado profundamente por sus apariencias adornadas y diseños ingeniosos. En esta lección le ayudaremos a construir una caja musical inteligente. Ahora, comencemos.

### Lista de Componentes

1 × micro:bit



1 × Tarjeta de expansión BOSON



1 × Auriculares



1 × Modulo Sensor de movimiento



1 × Cable USB

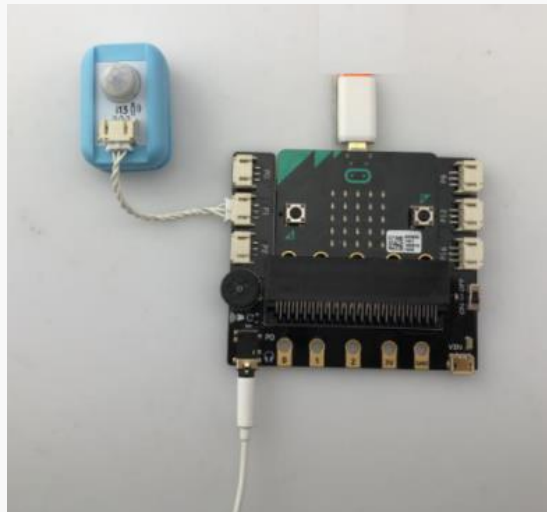


## Conexión

Conecte el Micro: bit al ordenador y conecte el auricular a la clavija.

Conectar el módulo del sensor de movimiento a P1.

Nota: como P0 está ocupado por la toma de audio, ya no estará disponible para conectar otros módulos.

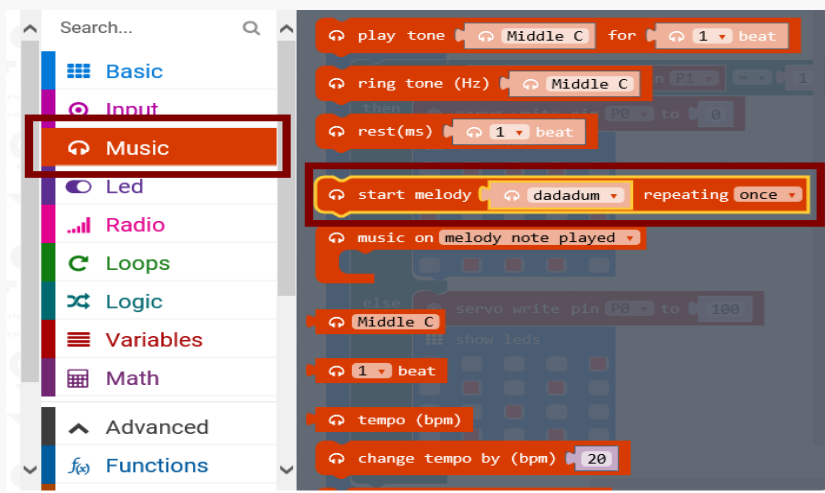


## Programa

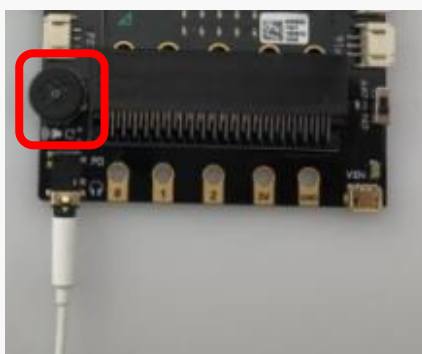
### Tarea 1: Caja musical electrónica

Objetivo: En esta parte, aprenderemos a dejar que Micro: bit reproduzca una pieza musical.

PASO 1: En este paso utilizaremos la función "Music". Mueva la función "start melody 'dadadum' repeating once" de "Music" al bucle "forever". También puedes elegir otras canciones que estén disponibles en la lista.



PASO 2: Dado que el "dadadum" dura 4 tiempos, la función "rest (ms)" de "Music" es necesaria para esta parte. El programa final es el siguiente:



## Tarea 2: Caja musical inteligente

Objetivo: Construiremos una caja musical inteligente en esta tarea. Cuando alguien se acerca, el Micro:bit reproducirá una pieza musical de su elección; cuando nadie está cerca, se queda en silencio...

PASO 1: Editar la canción "Twinkle, Twinkle Little Star".



En primer lugar, intentaremos editar la primera frase de la "Twinkle, Twinkle Little Star". Se requieren diferentes tonos y notas para editar la frecuencia vocal.

The image shows the musical notation for the first phrase of "Twinkle, Twinkle Little Star" in G major. The title "小星星" (Xiao Xing Xing) is at the top. The key signature is 1=C (one sharp, F#). The notes are numbered 1 through 6, corresponding to the solfège scale: 1 (Do), 2 (Re), 3 (Mi), 4 (Fa), 5 (Sol), 6 (La). The notes are: 1 (Do), 2 (Re), 3 (Mi), 4 (Fa), 5 (Sol), 6 (La), 5 (Sol), 4 (Fa), 3 (Mi), 2 (Re), 1 (Do). The lyrics "一闪一闪亮晶晶，满天都是小星星，" are written below the notes. Red arrows point from the labels "Tono" (Tone) and "Notas" (Notes) to the key signature and the numbered notes respectively.

"Twinkle, Twinkle Little Star".

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

not 1 2 3 4 5 6 7  
as

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

En este paso se utilizará la función "play tone Middle C for 1 beat" de "Music". Aquí, usted puede fijar el tono y los pulsos.

The image shows the Scratch Music block editor. The "Music" category is selected in the left sidebar. The main workspace shows a sequence of blocks: "play tone Middle C for 1 beat", "ring tone (Hz) Middle C", "rest(ms) 1 beat", "start melody dadadum repeating once", "music on melody note played repeating once", "rest(ms) 4 beat", "Middle C", "1 beat", "tempo (bpm)", and "change tempo by (bpm) 20".

The image shows a Scratch Music block editor with a "forever" loop. Inside the loop, there are several "play tone" blocks: "262 for 1 beat", "Middle A for 1 beat", "Middle A for 1 beat", and "Middle G for 1 beat". A piano keyboard is shown below the blocks, with the notes corresponding to the blocks highlighted. A label "play the 'Do' sound" is placed above the keyboard.

Cuando las funciones mencionadas anteriormente se combinan en el bucle "forever", nuestro micro:bit es capaz de reproducir la primera frase de la "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 "if else", la caja musical inteligente cobrará vida.

PASO 2: Cuando se conecta el sensor de movimiento y se añade la función "if else" al programa, poniendo en conjunto todas las funciones enumeradas anteriormente, el programa final de la caja musical inteligente tiene el siguiente aspecto:



## Proyecto 4: Tira LED de colores

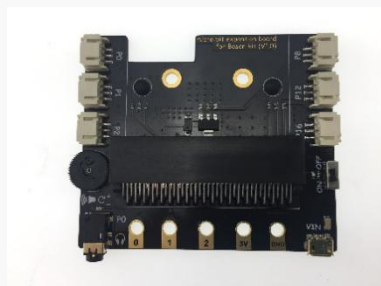
Después de haber aprendido a construir una caja musical inteligente, es posible que desee explorar más sobre el micro: bit. Las cajas musicales en nuestra vida diaria siempre llevan luces de colores. Por lo tanto, el proyecto para hacer la tira de LEDs colorida será introducido aquí para ayudarlo a construir una caja de música distintiva. Ven y únete a nosotros para entrar en el fantástico mundo de la luz!

### Lista de componentes

1 × micro:bit



1 × Tarjeta de expansión BOSON



1 × Tira de LED RGB



1 × Modulo Sensor de Sonido

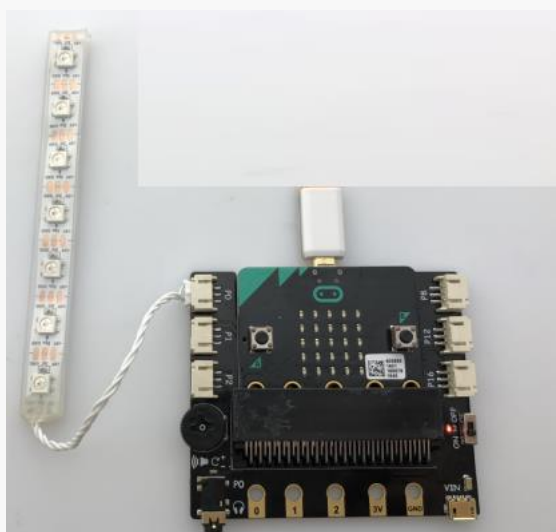


1 × Cable USB



## Conexión

Conecte la tira de LEDs RGB a P0.

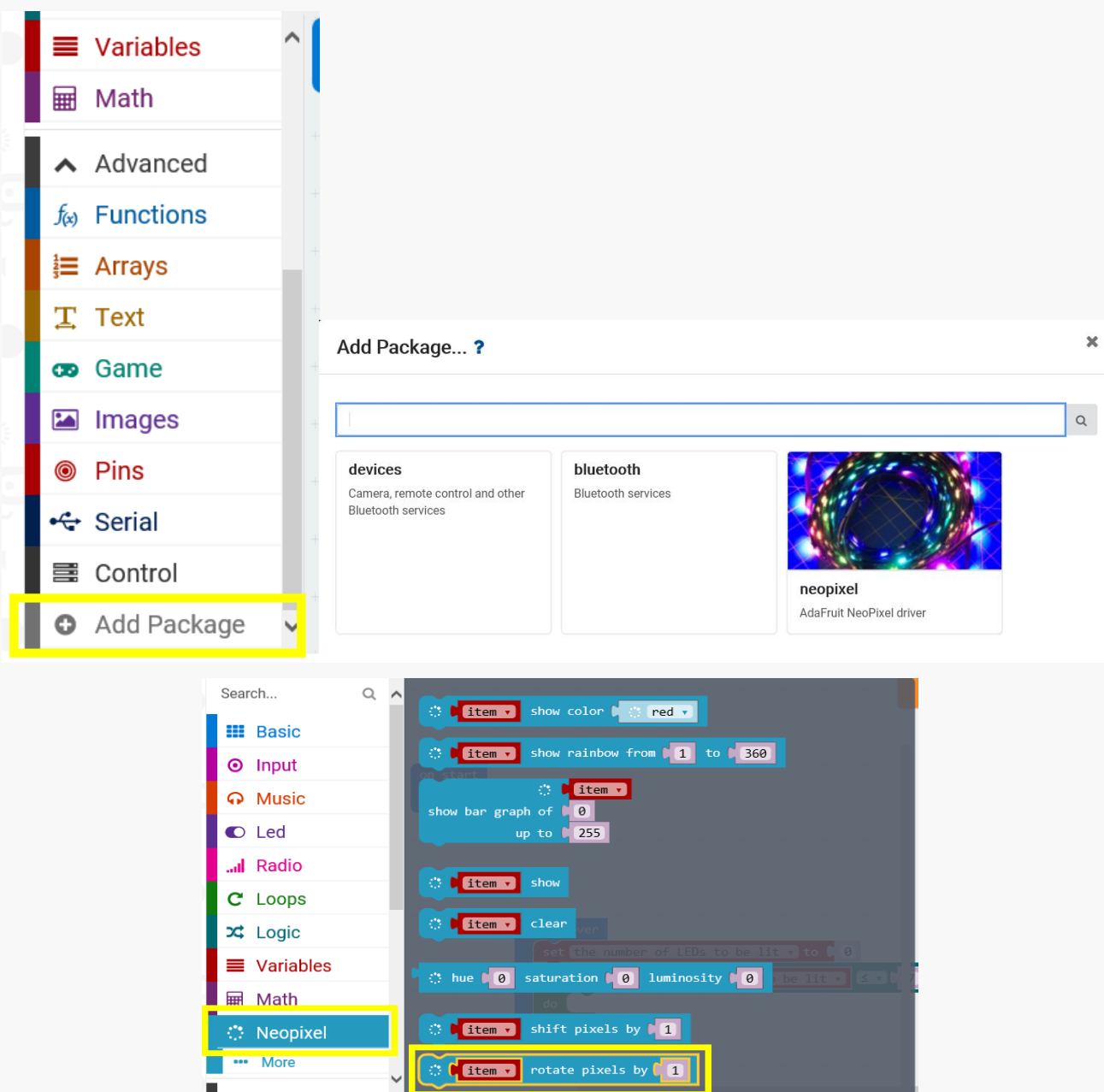


## Programa

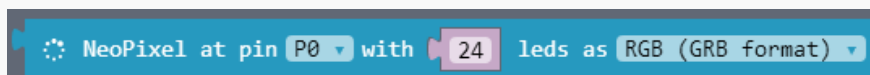
### Tarea 1: Encienda la tira de LEDs RGB

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

PASO 1: El bloque de funciones para el control de la barra de luces RGB no aparece en la caja de herramientas cuando iniciamos un nuevo proyecto. Para añadir esta función, necesitamos importar manualmente el "Neopixel" desde la parte inferior de la página. Haga clic en "Add Package" y para abrir el administrador de funciones de extensión, seleccione "Neopixel". El Neopixel aparecerá en la lista de funciones.



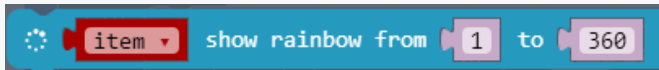
PASO 2: Para empezar a utilizar la función Neopixel, primero configuraremos cuántos LEDs están incluidos en la tira de luz y a qué pin se conecta. Necesitará poner estos dos parámetros en el bloque de función de la siguiente manera:



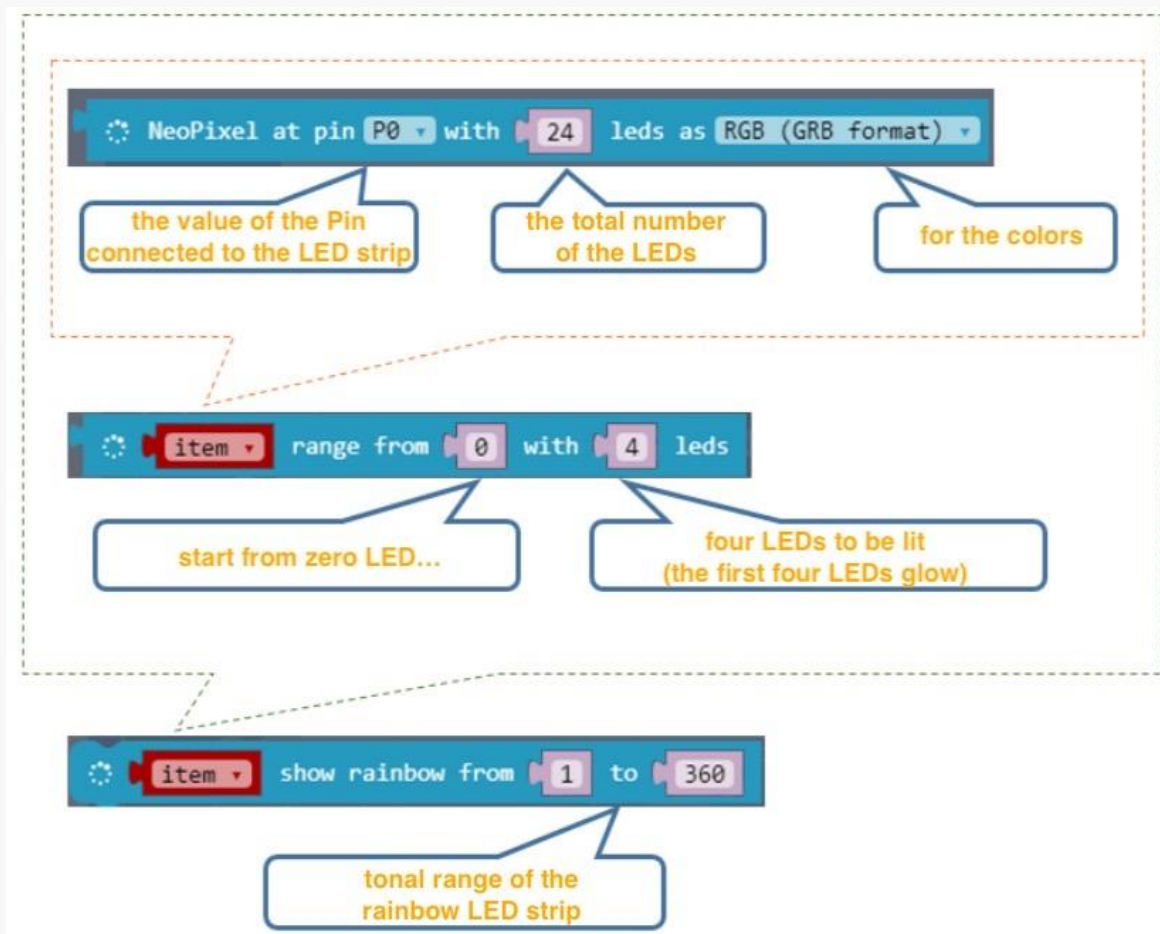
Después de esto, necesitamos configurar cuántos LEDs (del número total de LEDs que físicamente se incluyen en la tira) estarán bajo control del programa:



A continuación, utilice la función "show rainbow" para configurar el efecto de luz. Este efecto sólo se aplicará a los LEDs que definimos en nuestro último paso. El parámetro dentro del bloque de función se utiliza para establecer el rango de gradiente de color, 1 representa el rojo mientras que 360 representa el azul, cuanto más cerca estén estos dos números, menor será el cambio de gradiente:



El diagrama le ayudará a tener una comprensión general del significado de cada parámetro: (trate de ajustar el parámetro y ver cómo cambia el efecto de luz)



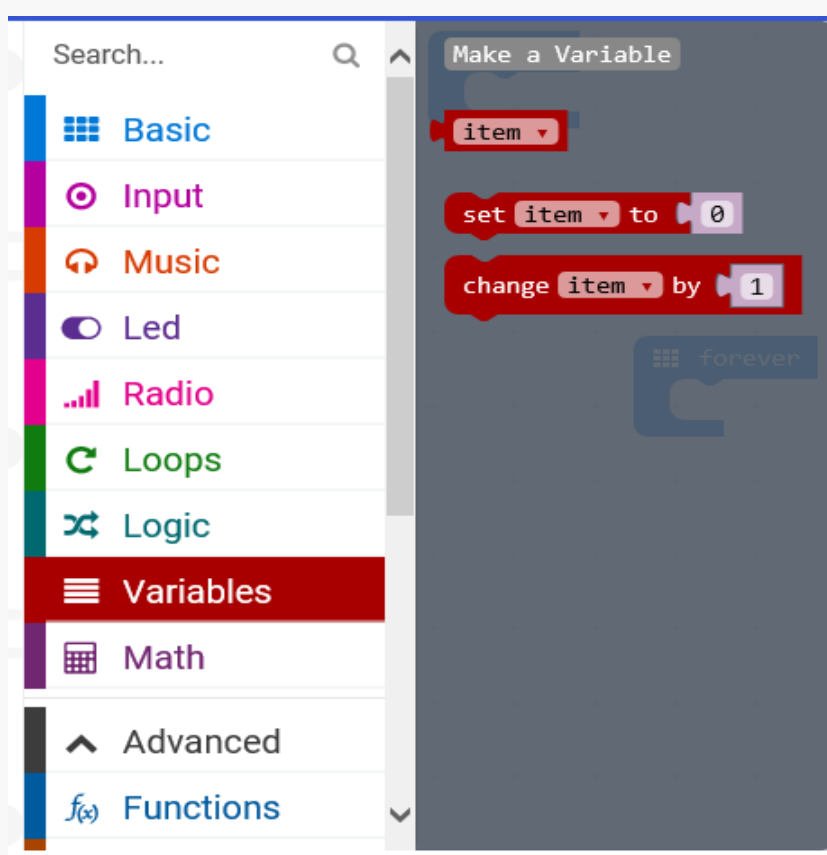
PASO 3: El pin P0 conectado a la tira viene con 7 LEDs. Para añadir un efecto de luz arco iris a la tira, necesitaremos establecer el rango de color de 1-360 (el rojo cambia gradualmente a azul). Luego, combinamos todos los bloques de función y los colocamos dentro del bucle "forever", y nuestro programa final tiene el siguiente aspecto:



## Tarea 2: Encender las luces del arco iris

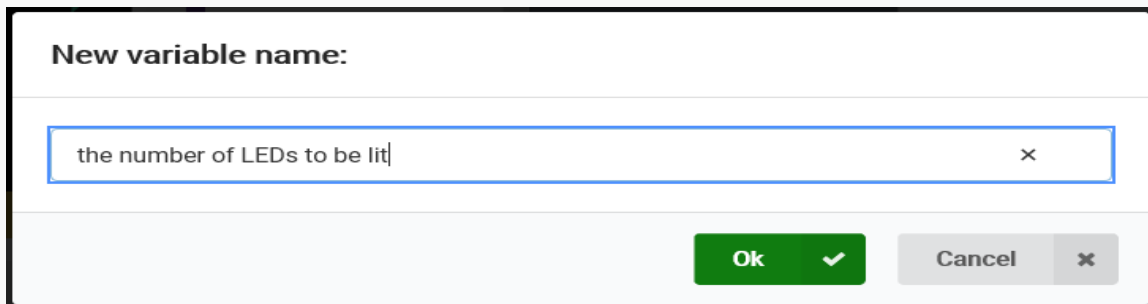
Objetivo: Hemos aprendido a encender y ajustar el efecto de luz de la tira de LEDs. Pero, ¿podemos hacer una simple animación de luz, digamos, hacer que los LEDs se enciendan en una secuencia?

PASO 1: La clave para crear una animación con la tira LED es cambiar el parámetro del efecto de luz. Crearemos una variable que cambie a lo largo del tiempo y la usaremos para reemplazar el parámetro de constante preestablecida. Para configurar una variable de este tipo, vaya a: "Variables" => "Make a Variable", y establecer el nombre de la variable (para que quede claro, nombramos la variable como "el número de LEDs a iluminar"), Luego haga clic en OK.



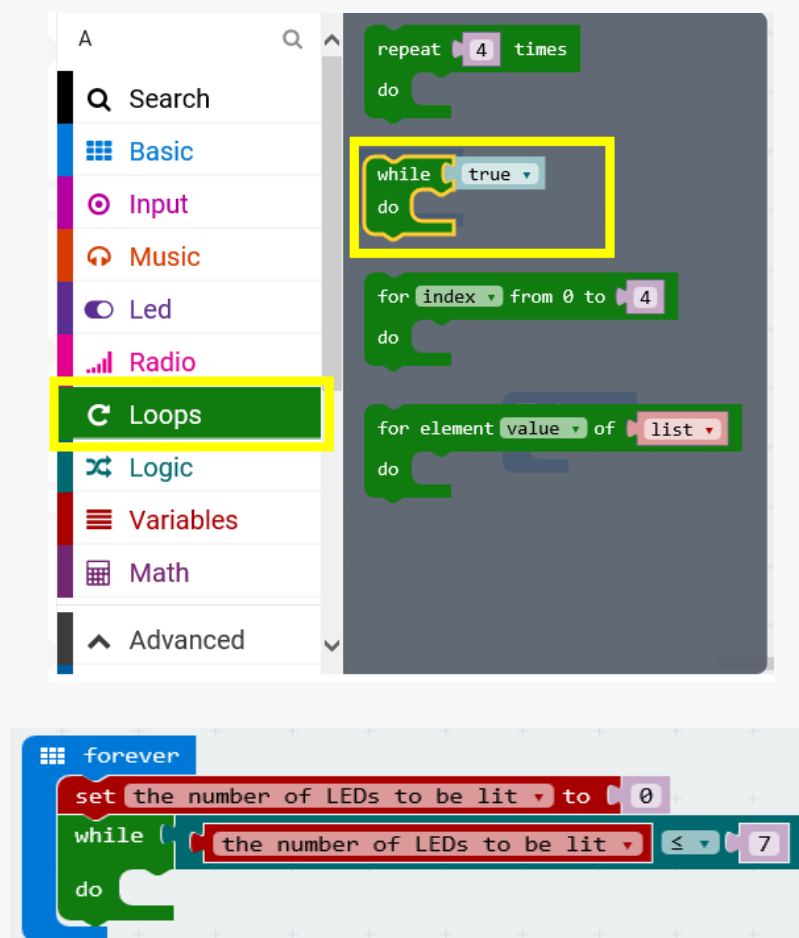
New variable name:

Ok ✓ Cancel ✕



PASO 2: Todos los LEDs arrancan apagados, lo que significa que el valor inicial de la variable "the number of LEDs to be lit" (el número de LEDs a iluminar) debería ser 0. Luego, usaremos una función de bucle para hacer que la variable se agrande en 1 en cada bucle. Además, para asegurarnos de que el número total del LED nunca exceda de 7, necesitaremos configurar la condición inicial usando la función "while do".

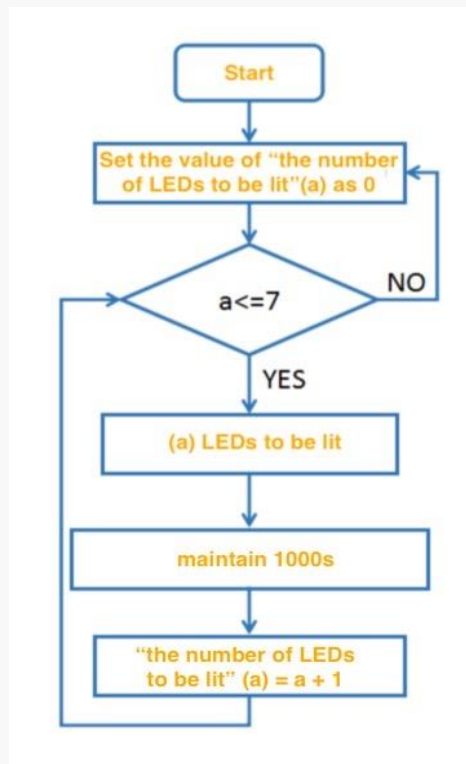
Utilizando el operador lógico "<" para ajustar la condición, tendremos: cuando "el número de LEDs a iluminar" sea inferior a 7, el bucle "while do" hará que el LED se ilumine 1 a 1 hasta que se enciendan todos los LEDs. Al juntar todas las funciones listadas arriba, el programa final va como:



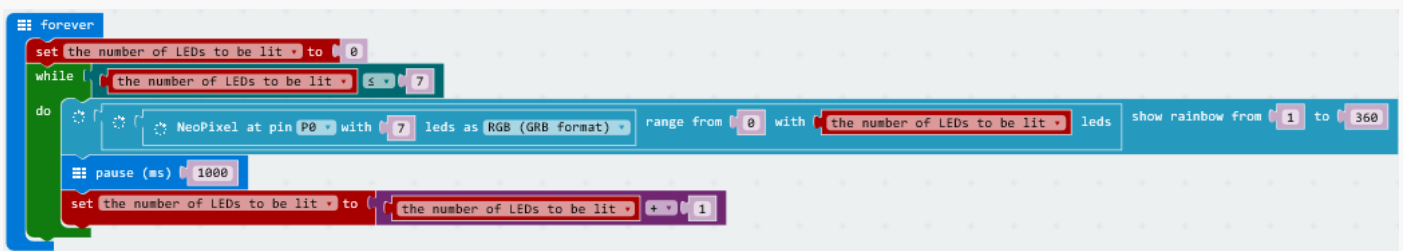
PASO 3: Ahora, el resto que necesitamos hacer es hacer que el parámetro se haga más grande por 1 en cada bucle. Sin embargo, no queremos que esto suceda demasiado rápido, una función



"rest" puede ayudarnos a ralentizarlo. El diagrama de flujo que se muestra a continuación puede ser útil para ayudarnos a determinar el aspecto del programa:



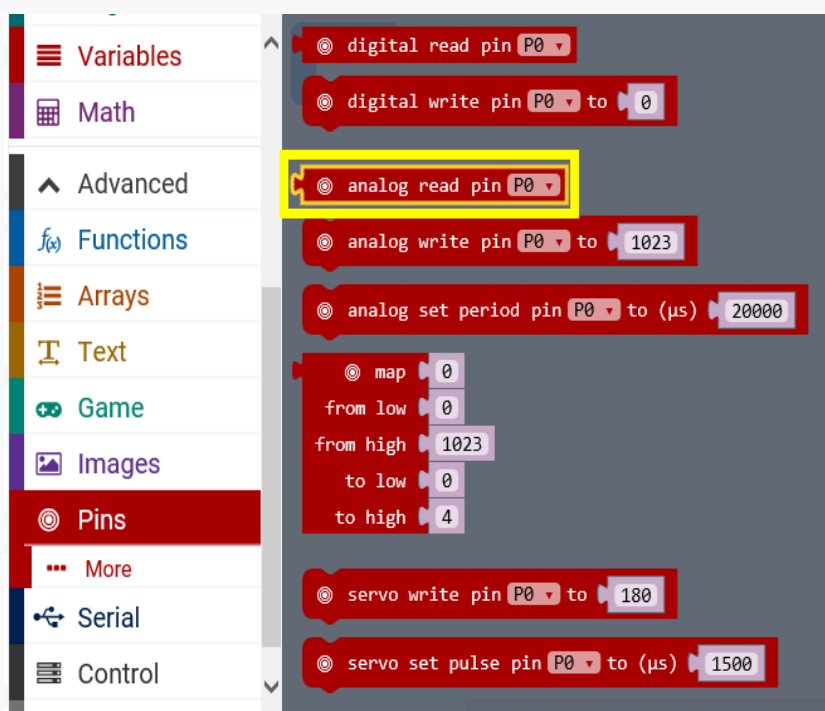
PASO 4: Para que el LED se ilumine de 1 en 1 cada 1 segundo, tendremos que poner la variable "the number of LEDs to be lit" dentro de la función "LED range from", y poner una "pause 1000ms" en el bucle. Y esto es lo que tenemos.



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

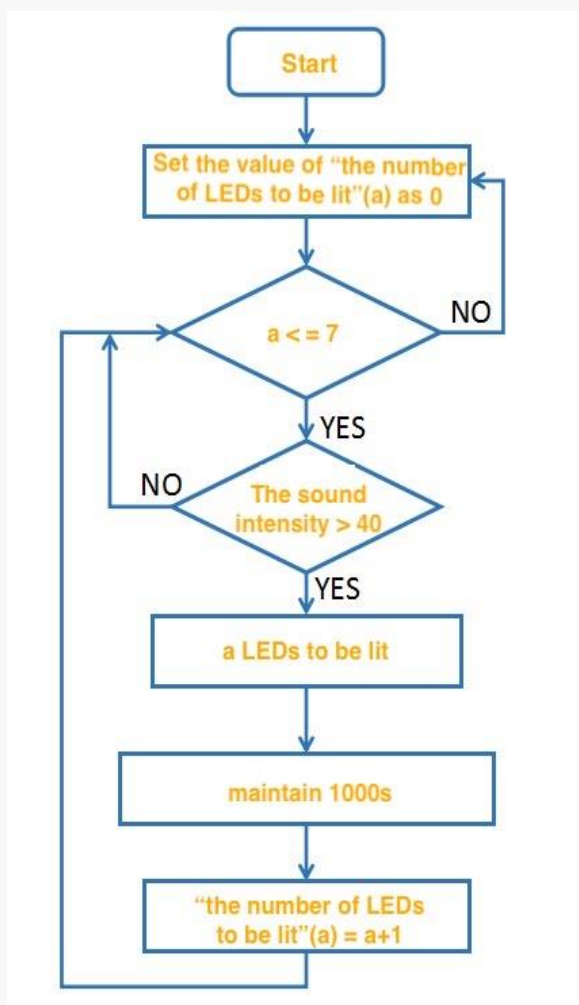
Objetivo: Lo que aprenderemos en esta parte es utilizar un sensor de sonido para controlar la tira de LEDs: cuando se detecta sonido, 1 LED más se enciende en la tira.

PASO 1: Primero necesitamos configurar el sensor de sonido tal como lo hicimos en los capítulos anteriores. Recuerde que en nuestro caso, el sensor está conectado a P1.



PASO 2: Cuando el valor es superior a 40 y el número total de LEDs iluminados es inferior a 7, se enciende uno más. El sensor de sonido detectará continuamente el valor.

El diagrama de flujo que figura a continuación puede ser útil para comprender el mecanismo:



PASO 3: Poner la condición "if the analog P1 > 40" en el programa que construimos en la última parte, tendremos el programa final como se indica a continuación:



## Capítulo 5: Conviértase en un experto

Hemos construido tantos trabajos divertidos con diferentes tipos de módulos en capítulos anteriores. Ahora puede tener una comprensión más profunda del micro: bit. ¿Es suficiente para la exploración? La respuesta es sin duda NO! Aguanta un poco más y daremos un paso más en el micro:bit.

### Proyecto 1: Estabilizador Electrónico

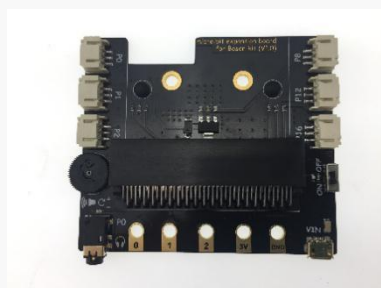
¿Has oído hablar del acelerómetro? El acelerómetro mide la aceleración en 3 direcciones y ahora es extremadamente popular en todo tipo de aparatos electrónicos. Por ejemplo, si giramos el teléfono o el portátil 90 grados, su página también girará 90 grados automáticamente, lo que ahorra mucho tiempo en la configuración de la dirección. En esta parte, vamos a jugar con el acelerómetro con el Micro:bit!

#### Lista de componentes

1 × micro:bit



1 × Tarjeta de expansión BOSON



1 × Modulo servo

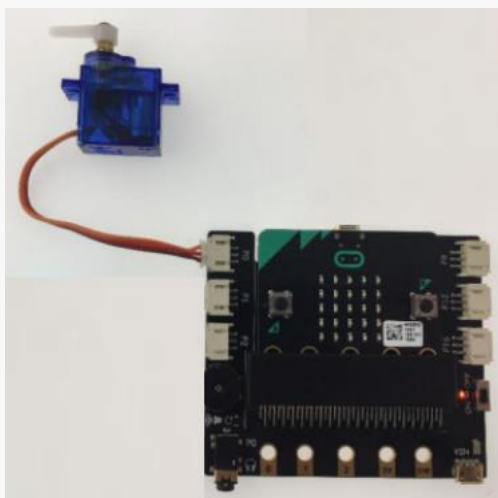


1 × Cable USB



## Conexión

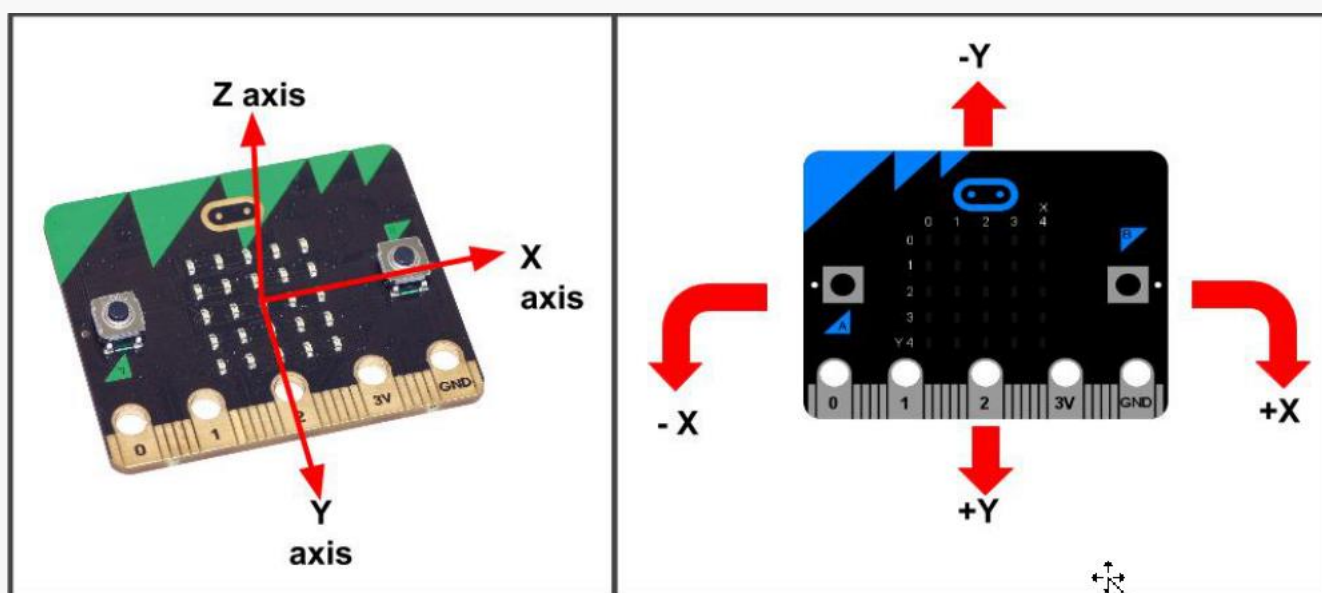
Conecta el servo a P0



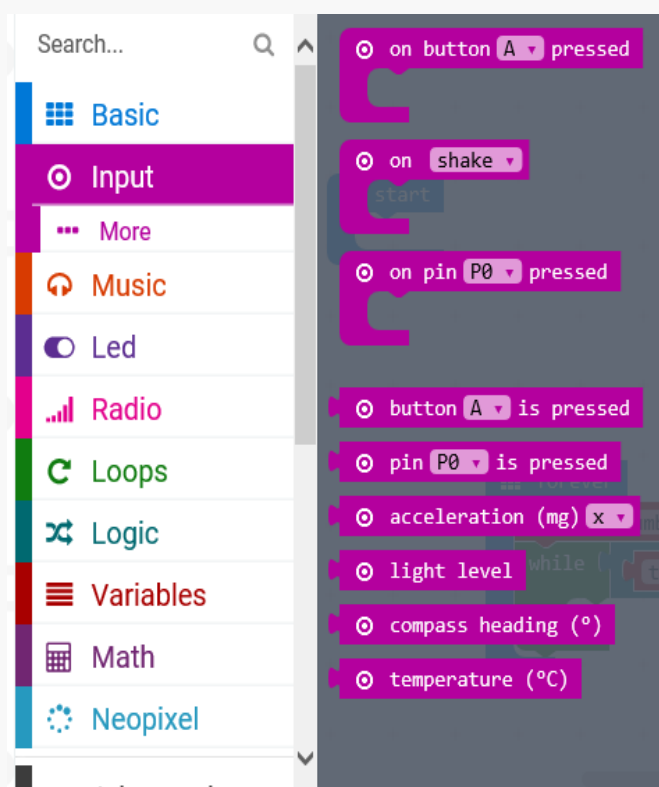
## Programa

Objetivo: Fijar el servo en el Micro:bit, el servo seguirá apuntando verticalmente sin importar como lo hagamos.

Micro:bit detecta el gesto automáticamente a través del acelerómetro. La fuerza de gravitación es también una forma de aceleración, y el micro:bit puede detectar la fuerza de gravitación a lo largo de las direcciones X, Y, y Z, mientras que X está a lo largo de la dirección izquierda-derecha, Y está a lo largo de la dirección adelante-atrás, Z es perpendicular a la tarjeta y a lo largo de la dirección arriba-abajo.



Los valores de los ejes X, Y y Z se obtienen mediante la función " acceleration " bajo " Input ".

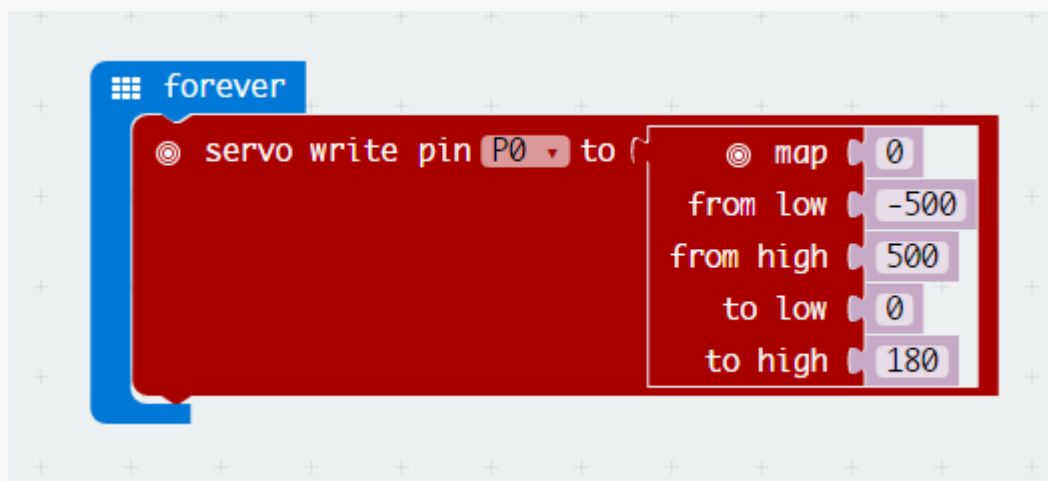


Cuando la placa base está plana sobre el escritorio, la fuerza gravitacional sube y baja, de modo que los valores de los ejes X e Y están cerca de cero y el eje Z está alrededor de 1000 ( $\text{mm} / \text{s}^2$ ), lo que equivale aproximadamente a la aceleración gravitacional.

Cuando inclinamos la placa, los valores de X e Y cambian. Cuando se inclina hacia la izquierda, el valor de X es negativo o, por el contrario, positivo.

PASO 1: Cuando la placa está inclinada hacia la izquierda, el acelerómetro producirá un valor de salida negativo a lo largo de la dirección X. Giraremos el Micro:bit hacia 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 -500 y 500. Podemos hacer que el -500 ~ 500 corresponda a 0 ~ 180 grados del servo a través de la función "map".



PASO 2: Fije el servo y la tarjeta de expansión con cinta adhesiva para que estos dos permanezcan relativamente quietos. Tenga en cuenta que la orientación del servo debe ser la misma que la del puerto USB, de modo que el eje del engranaje sea perpendicular al eje X y el grado de rotación del servo sea complementario al grado de inclinación del Micro:bit.

## Ejercicios

Intenta construir una brújula usando la brújula electrónica integrada en Micro:bit.

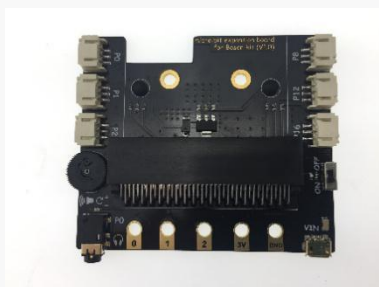
## Proyecto 2: Panel de DJ

### Lista de componentes

1 × micro:bit



1 × Tarjeta de expansión BOSON



1 × Tira de LED RGB



1 × Modulo perilla



1 × Cable USB

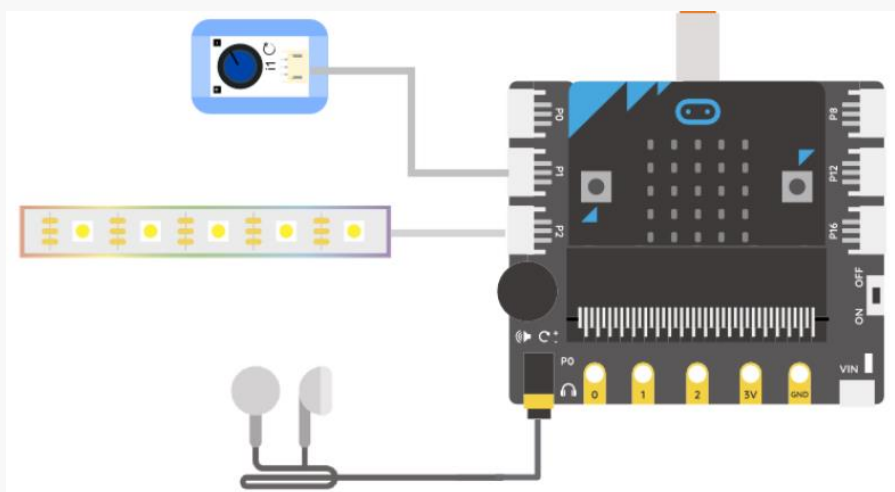




## Conexión

Conecte el módulo de perilla a P1;

Conecte la tira de LEDs RGB a P2.



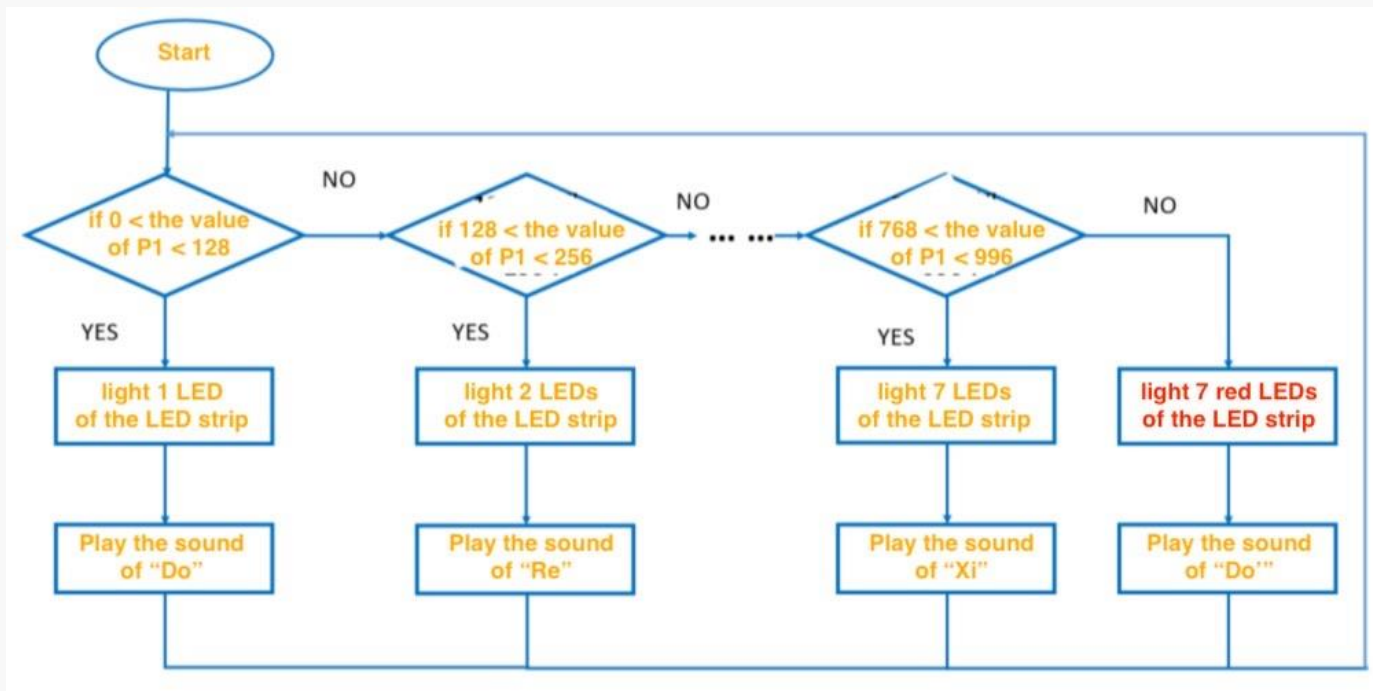
## Programa

Objetivo: En este proyecto, la caja musical inteligente y la tira de LEDs se combinan para realizar el funcionamiento de la música con la tira de LEDs al ajustar manualmente el ángulo de rotación de la perilla.

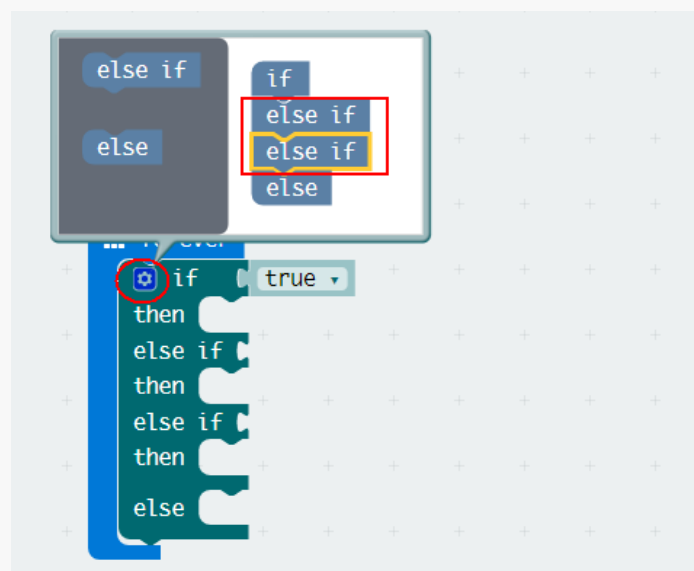
Girando la perilla a mano, el valor de P1 cambia de 0 a 1023 y se divide en 8 pequeños rangos, correspondientes a una octava "Do ~ Do'" y el número de LEDs a iluminar.

Valor de entrada de P1	Sonido	Tira de LED
0-128	Do	Enciende un LED
128-256	Re	Enciende dos LEDs
256-384	Mi	Enciende tres LEDs
384-512	Fa	Enciende cuatro LEDs
512-640	So	Enciende cinco LEDs
640-768	La	Enciende seis LEDs
768-996	Si	Enciende siete LEDs
996-1023	Do'	Enciende siete LEDs en rojo

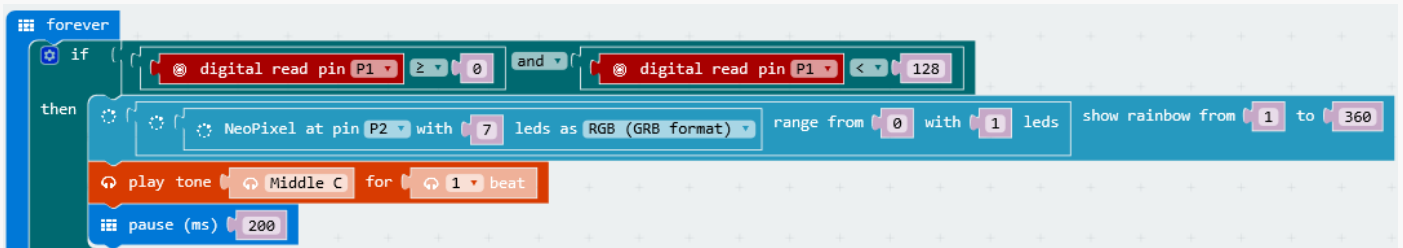
PASO 1: El siguiente diagrama lógico le ayudará a construir una idea general sobre cómo realizar un PANEL DE DJ.



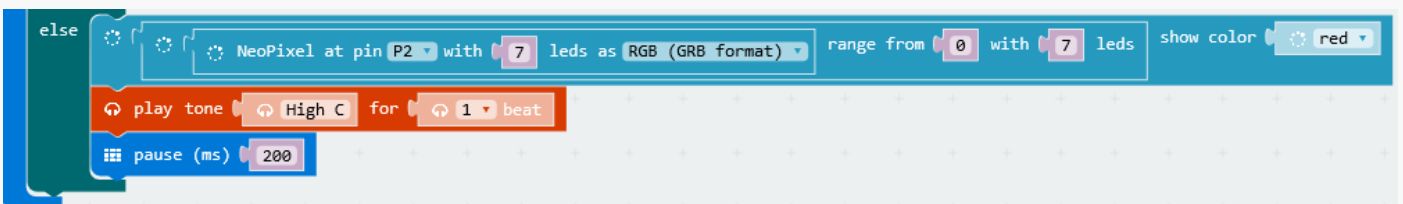
PASO 2: En este paso aprenderemos a editar la función "if else". En primer lugar, coloque el "if else" de "Logic" bajo el bucle "forever". Como podemos ver desde abajo, en la esquina superior izquierda de la función "else if" hay un icono azul con forma de engranaje (marcado en rojo como abajo). Haga clic en él y arrastre la frase "else if" desde el lado izquierdo a la parte central entre el "if else" del lado derecho. El siguiente diagrama le ayudará a finalizar este paso.



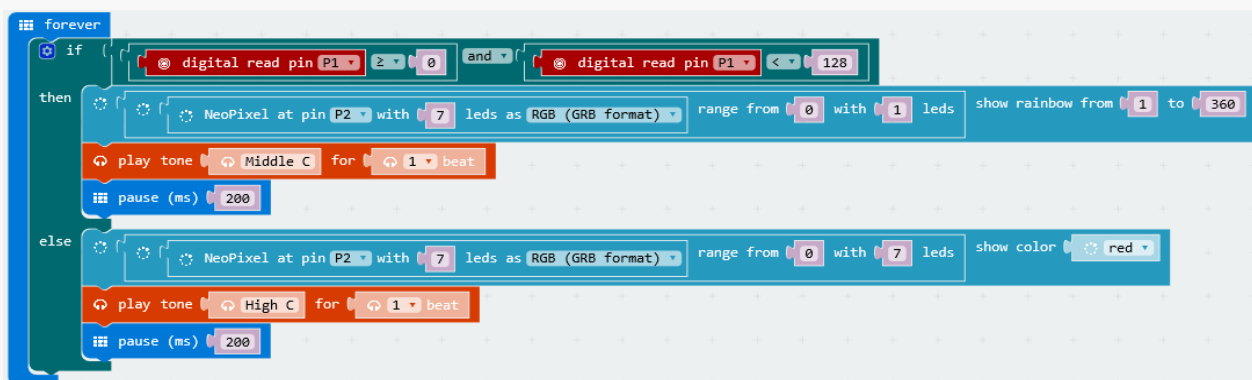
PASO 3: En este paso, la función "y" y el operador "<" bajo "Logic" serán aplicados para juzgar el rango de valores de entrada de P1. Las funciones de "NeoPixel" se utilizarán para controlar el efecto de iluminación de la tira de LEDs GRB. Y las funciones de "Music" y "pause (ms)" de "Basic" serán tomadas para controlar la música. El siguiente es el programa terminado en este paso:



PASO 4: Como sólo hay 7 LEDs, el octavo sonido "do" puede expresarse con un efecto luminoso diferente.



PASO 5: Combine todas las funciones mencionadas anteriormente, el programa final es el siguiente:



## Ejercicios

Usted debe haber notado que hay una matriz LED de 5×5 en el micro:bit. ¿Por qué no hacer un uso completo de ella! Intente usarlo para mostrar las notas musicales correspondientes cuando hayan sido reproducidas.



## Proyecto 3: Timbre Remoto

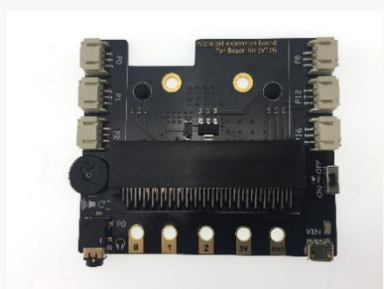
Para aquellos que tienen un timbre en sus casas, puede ser un problema bastante común que a veces el timbre no sea lo suficientemente alto como para llegar a las habitaciones. Para resolver este problema, una sirena inalámbrica puede ser útil. En el capítulo, aprenderemos cómo hacer que el timbre de la puerta controle un LED remoto que podamos llevar a todas partes.

### Lista de Componentes

2 × micro:bit



1 × Tarjeta de expansión BOSON



1 × Modulo LED



1 × Modulo Sensor de movimiento



1 × Módulo botón

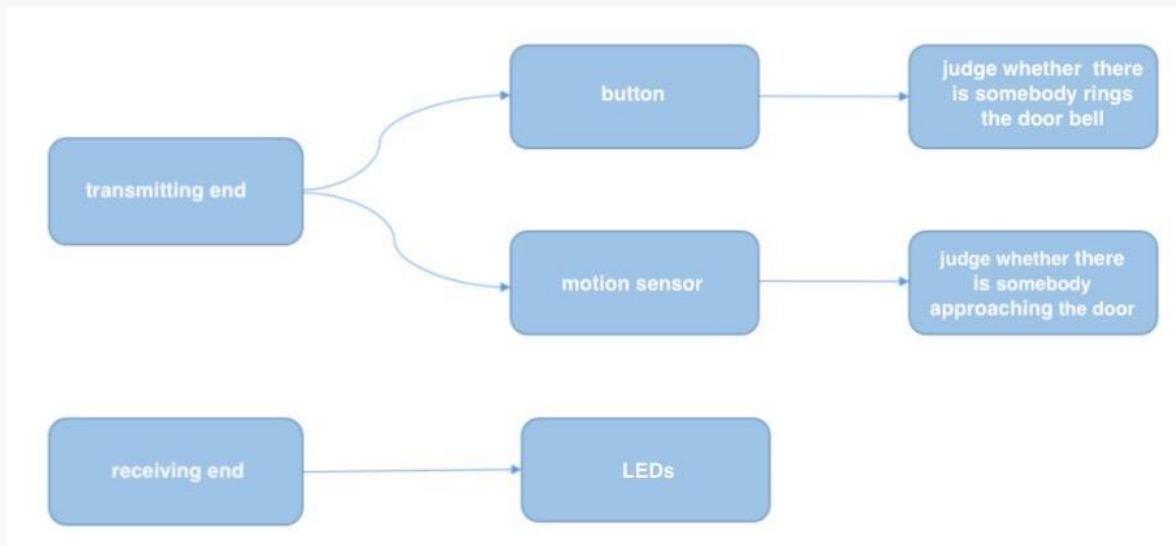


1 × Cable USB

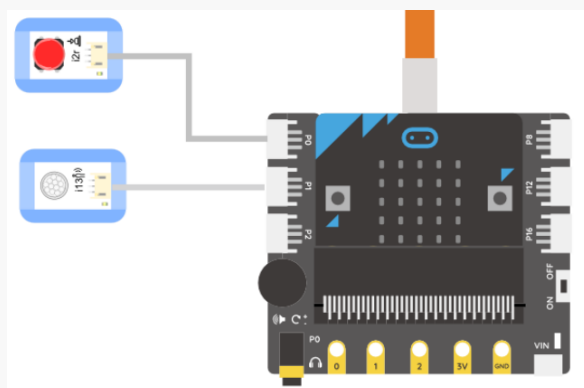


## Conexión

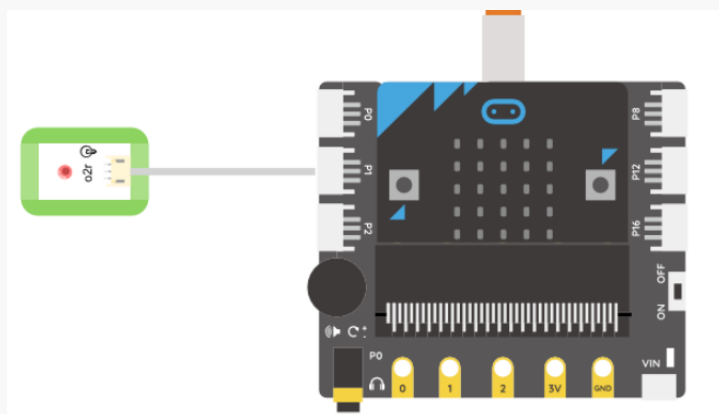
Necesitaremos dos Micro:bit en este proyecto, uno está conectado al timbre de la puerta y envía el mensaje (extremo transmisor), el otro recibe la señal y controla el indicador LED (extremo receptor). El Micro:bit se comunica por radio.



Esquema de conexión del transmisor Micro:bit:



Esquema de conexión del receptor Micro:bit:

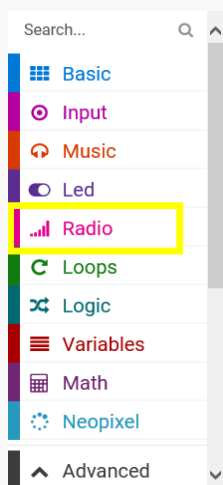


## Programa

### Tarea1: Programar el transmisor

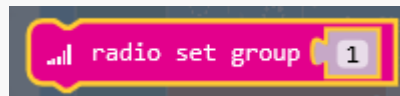
PASO 1: Hemos estado bastante familiarizados con el módulo de botones y el sensor de movimiento, sin embargo, ¿cómo podemos enviar su estado a través de la radio? Para lograr esto, aprenderemos cómo funciona exactamente la radio en la siguiente sesión.

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



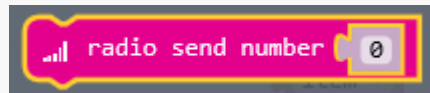
Primero necesitamos entender las siguientes 3 funciones más comúnmente usadas bajo "Radio":

"set group":



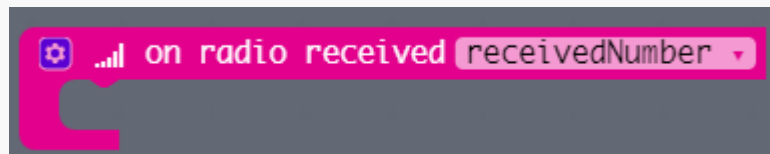
Descripción de las funciones: Hay un total de 255 canales de radio diferentes. El transmisor y el receptor deben estar en el mismo grupo para poder comunicarse.

"send out number":



Descripción de la función: enviar un número por radio.

"on radio received":

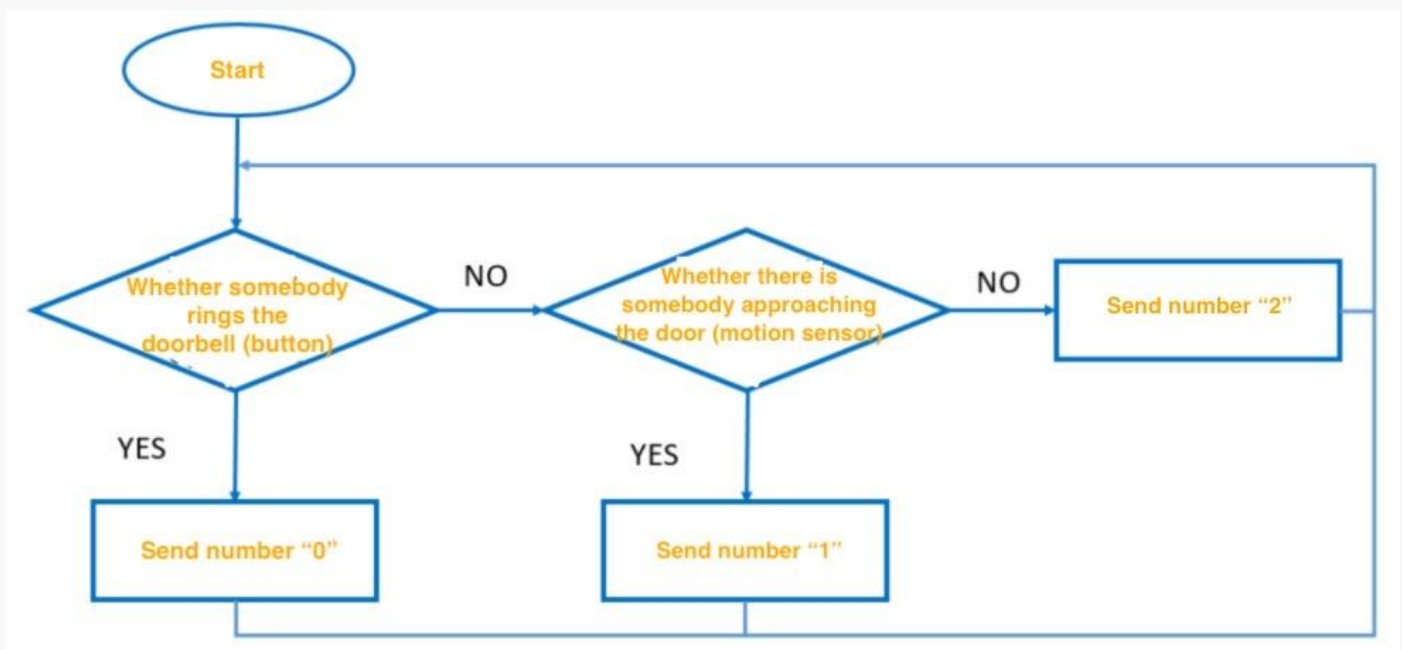


Descripción de las funciones: Al recibir un número, éste ejecutará el programa dentro del bucle "on radio received".

También tendremos que programar el receptor para que haga la reacción correspondiente basándonos en el número recibido.

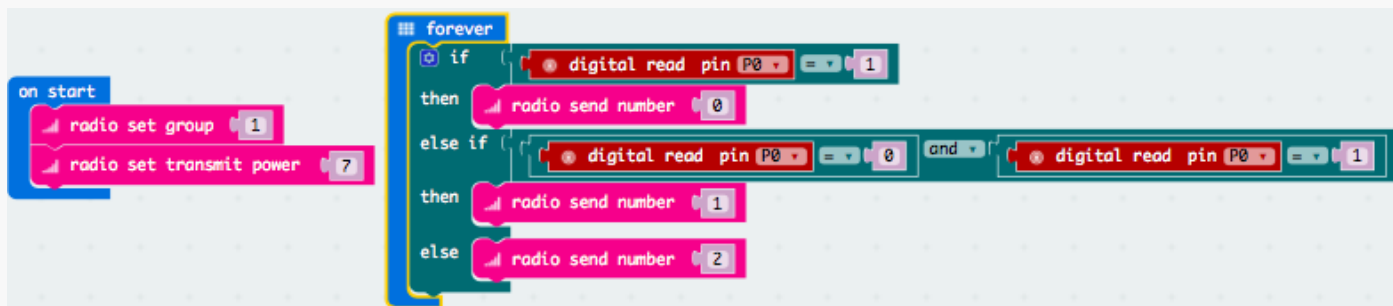
Nota: la variable "receivedNumber" dentro de la función "on radio received" representa el número recibido.

PASO 2: Después de saber cómo utilizar la función de radio, es el momento de utilizar los módulos de entrada para activar la señal de radio. Para hacerlo un poco más desafiante, intentaremos programar el micro:bit enviar un mensaje cuando alguien toque el timbre, y enviar otro mensaje cuando haya alguien pasando por aquí. Para lograrlo, necesitaremos el módulo de botones y un sensor de movimiento. El siguiente diagrama de flujo le da una idea de cómo escribir el programa.



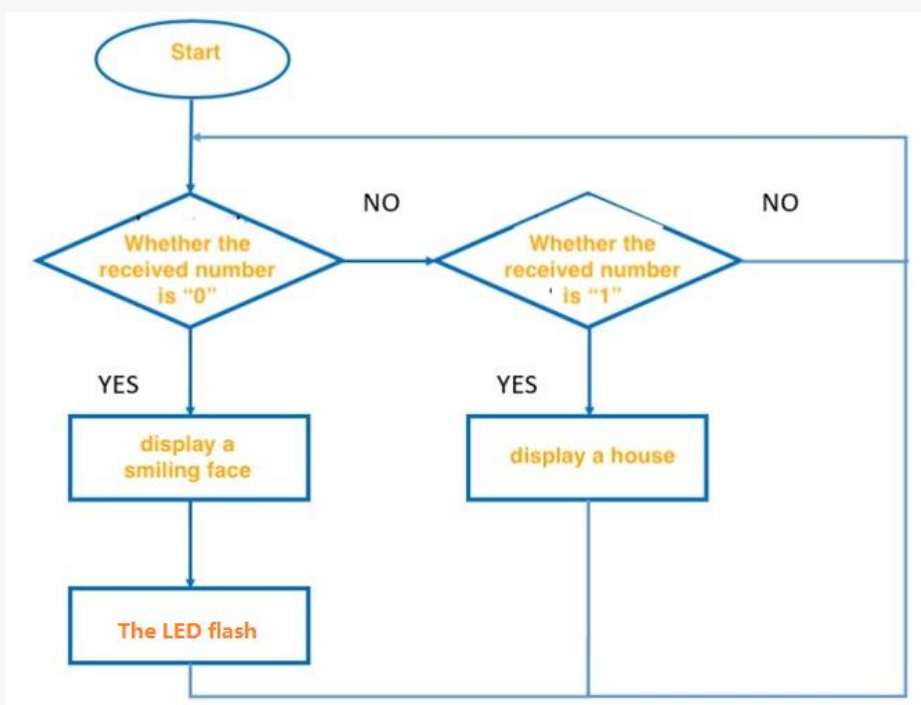


PASO 3: Juntar los bloques de función. El programa completo tendrá el siguiente aspecto.

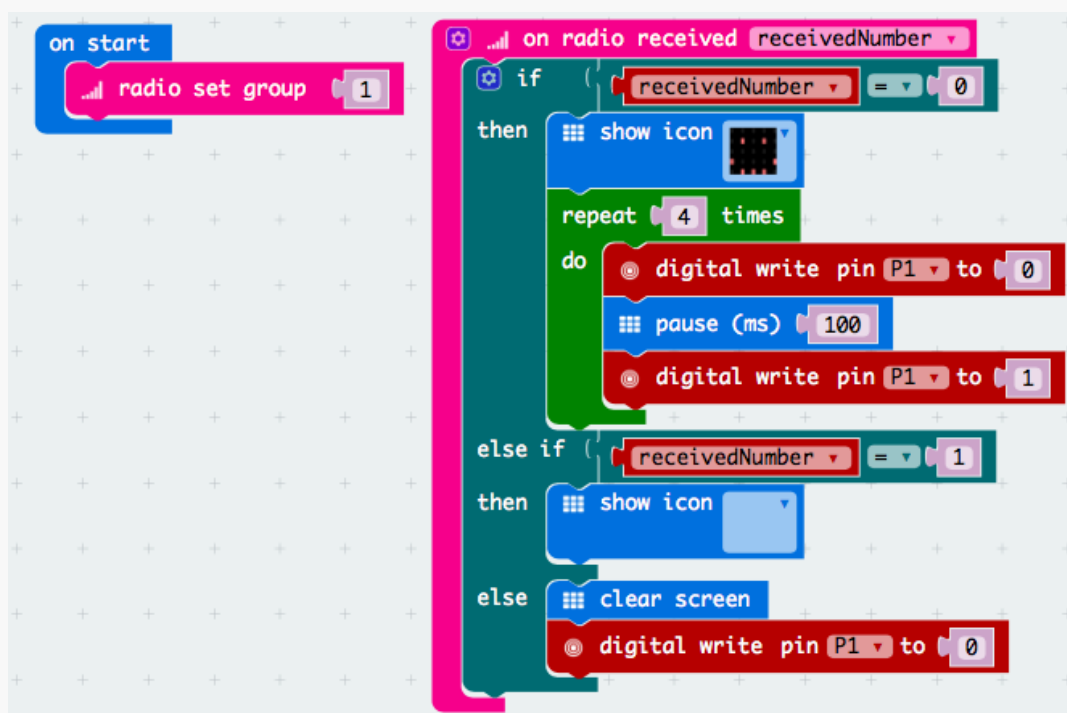


## Tarea 2: Programar el receptor

PASO 1: El receptor es capaz de ejecutar diferentes acciones basadas en la señal recibida. El siguiente diagrama de flujo le ayudará a construir el programa.



PASO 3: Juntar los bloques de función. El programa completo será el siguiente.



## Ejercicios

Añada un aviso de "Por favor, espere" al timbre de la puerta cuando alguien toque el timbre.

## Proyecto 4: Escape el laberinto

Micro:bit está atrapado dentro del laberinto con una bomba de tiempo! El micro:bit no puede moverse por sí solo. Sin embargo, envía una señal a través del panel LED para guiar el camino. Tenemos que ayudarlo a huir del laberinto antes de que la bomba explote!

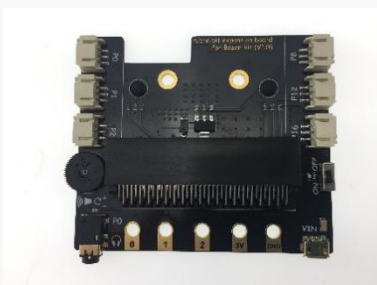


### Lista de componentes

1 × micro:Bit



1 × Tarjeta de expansión BOSON



1 × Tira de LED RGB

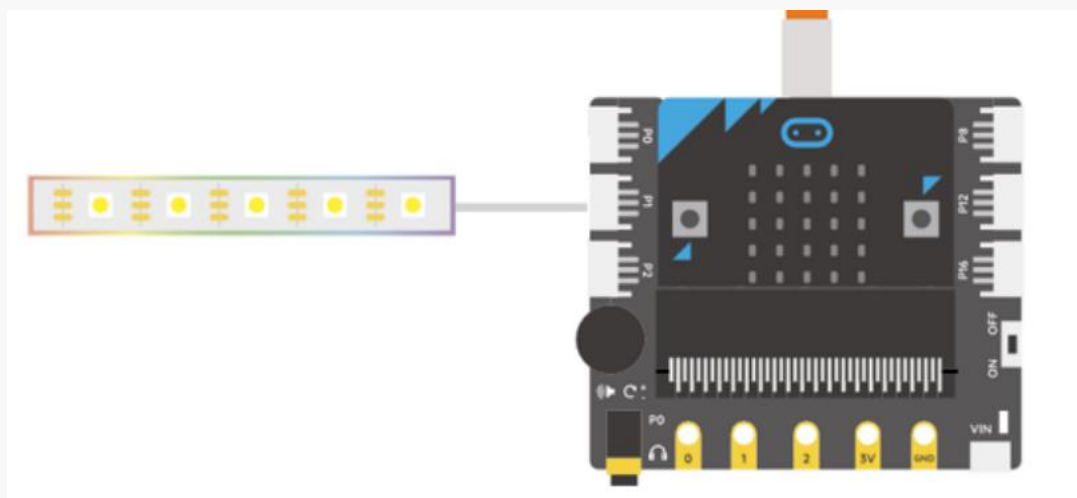


1 × cable USB



## Conexión

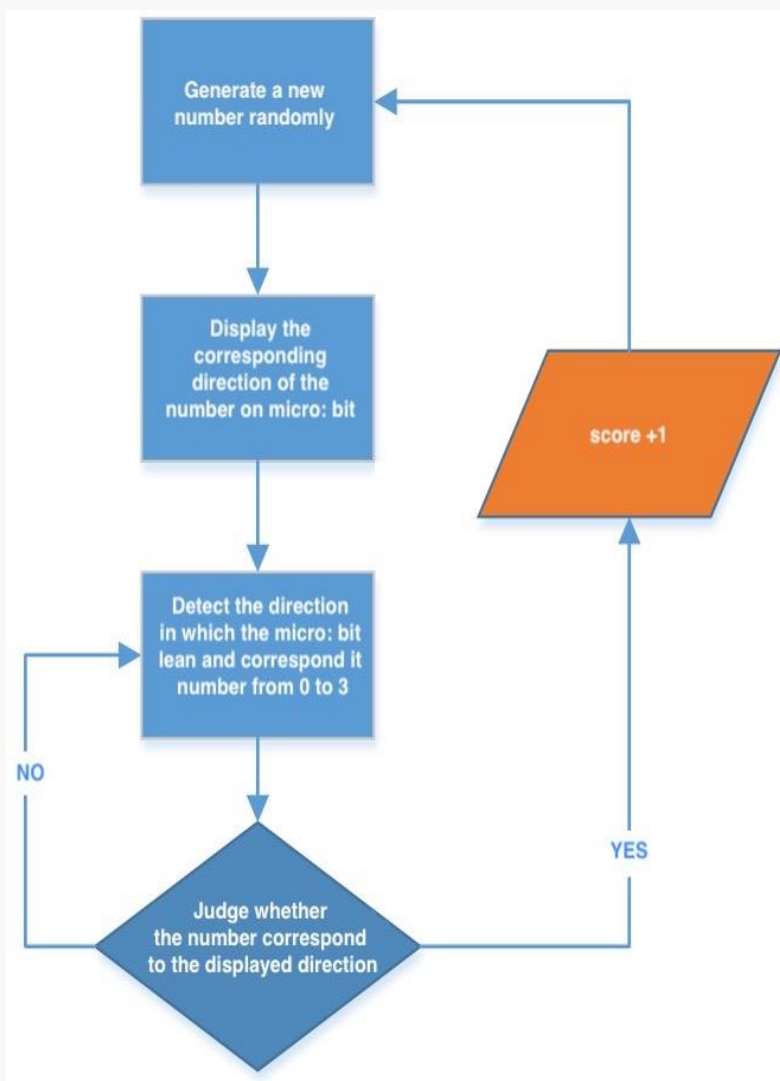
Conecte la tira de LEDs RGB a P1.



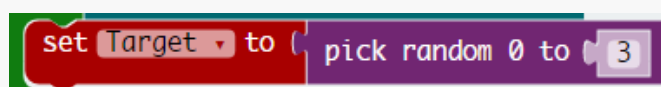
## Programa

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

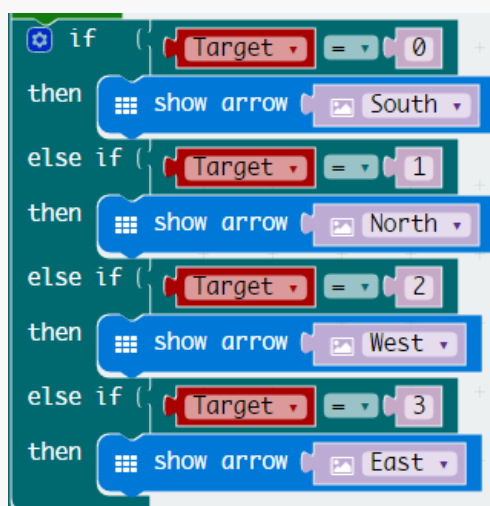
El siguiente diagrama de flujo puede ser útil para explicar cómo funciona el programa.



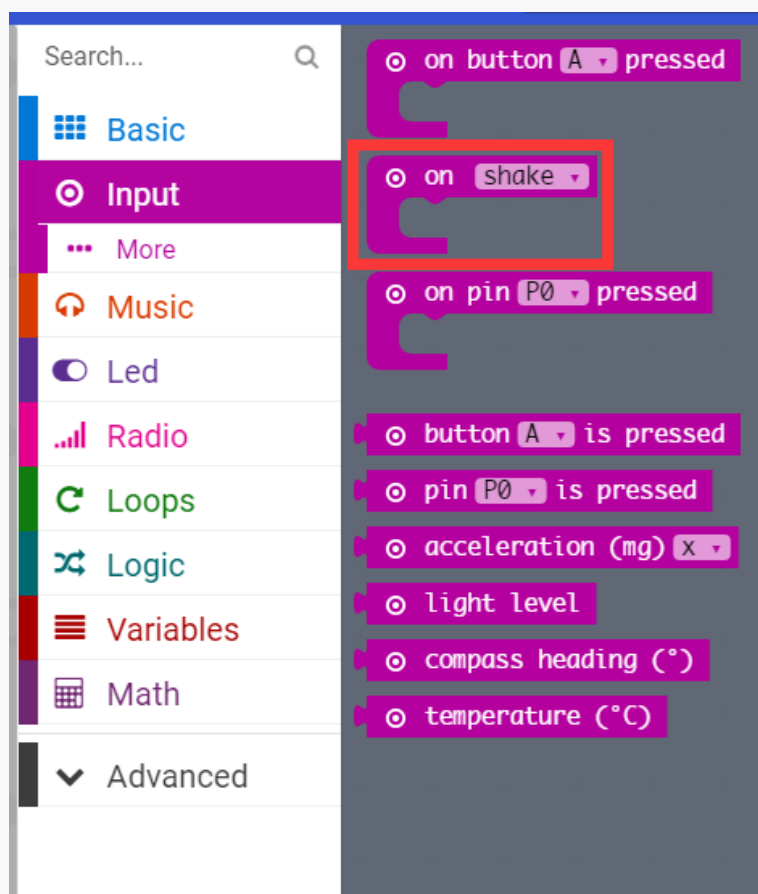
Usaremos el número 0 ~ 3 para representar cuatro direcciones objetivo respectivamente. La función "pick random" bajo "Math" nos ayuda a obtener un número aleatorio de 0 ~ 3

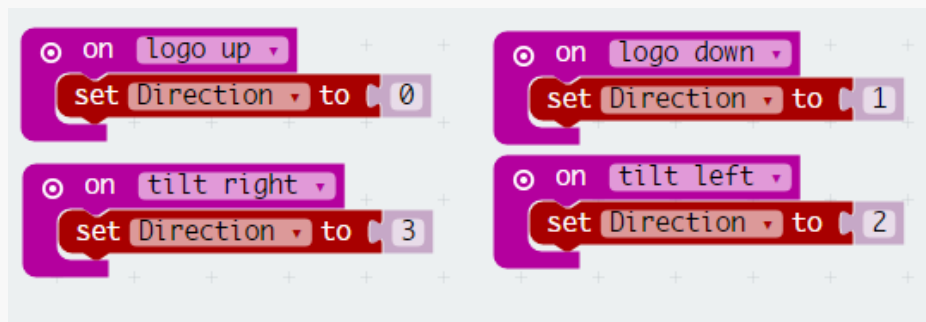
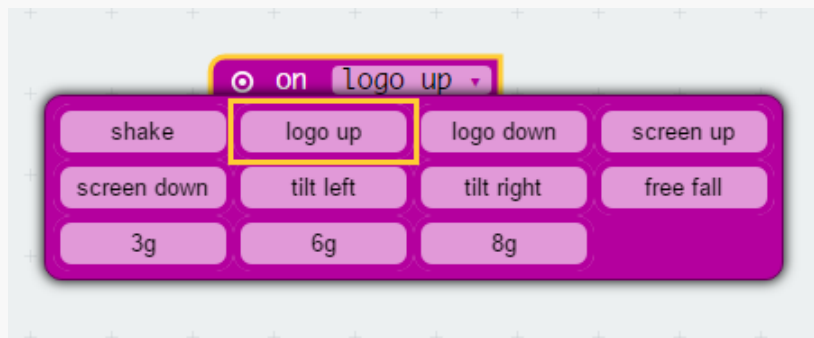


PASO 2: Muestre la imagen con la flecha en el panel de LEDs. La flecha debe corresponder a la dirección del objetivo generada aleatoriamente que obtuvimos desde arriba. Una vez más, la función "if-else" será nuestra elección para hacer el trabajo.

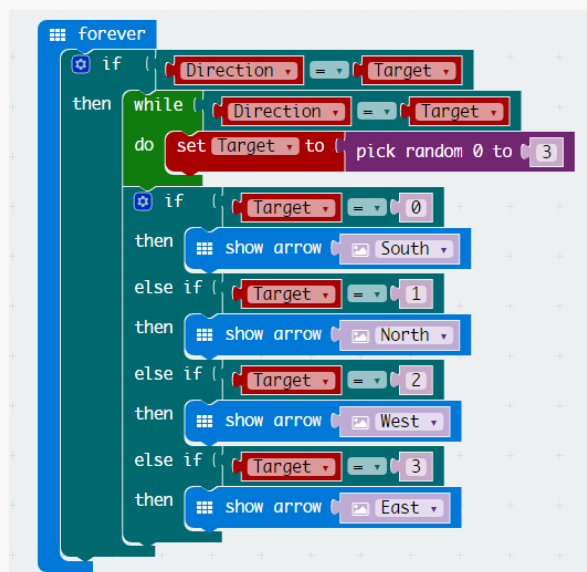


PASO 3: Utilice la función "on xxx gesture" bajo "input" para saber en qué dirección se inclina el Micro:bit. El número dentro del bucle de función representa cuatro direcciones diferentes.





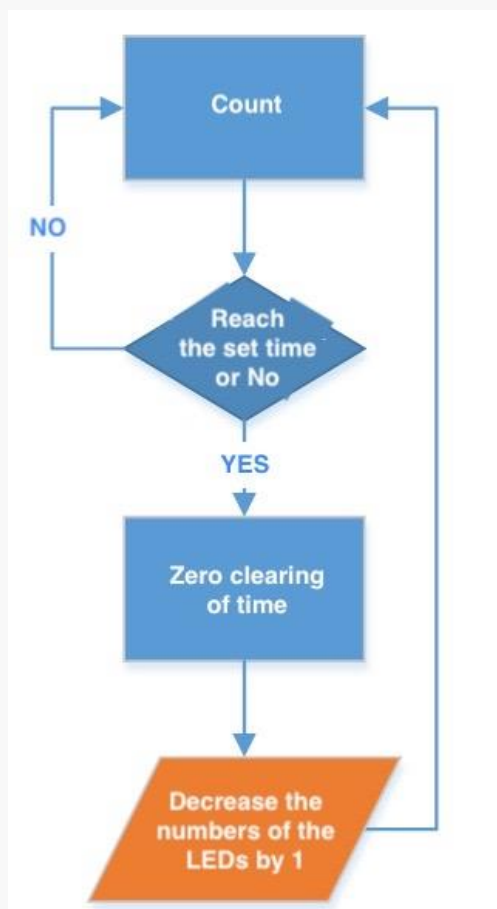
PASO 4: Ponga todo lo que mencionamos arriba dentro de un bucle "if-then" para comparar si las dos direcciones coinciden. Además, no olvides el bucle "forever" para mantener el juego siempre en marcha.



Nota: será útil añadir una función "While" dentro del bucle, para evitar que aparezcan direcciones idénticas en una fila.

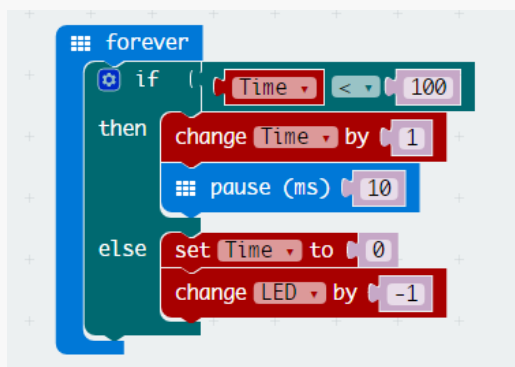
PASO 5: es el momento de agregar la bomba. La bomba empezará a hacer tictac tan pronto como empiece el juego. También, usaremos la tira de Led RGB para simular el fusible. Con el tiempo, los LEDs del fusible se apagarán uno tras otro a un ritmo constante. Si el micro:bit no puede escapar del laberinto antes de que se apague el fusible, el juego ha terminado.

El diagrama de flujo que se muestra a continuación muestra cómo ajustar el LED para que se encienda uno por uno con el tiempo transcurrido

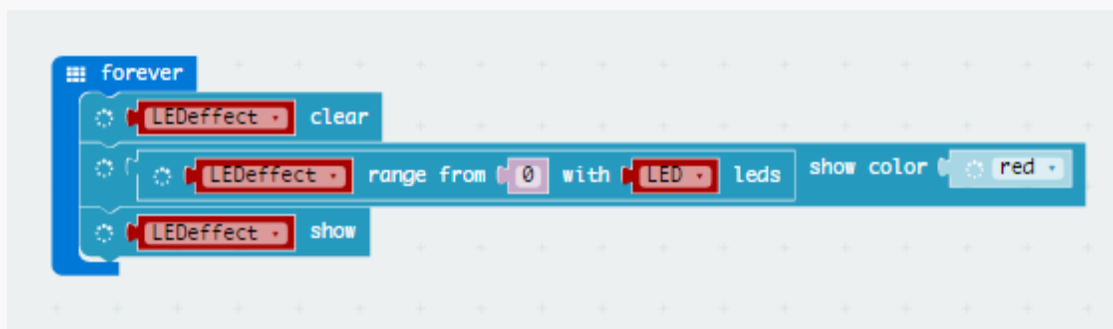


Explicaremos las funciones que necesitamos para lograrlo.

Nota: Micro: el bit necesita tiempo extra para procesar los datos, por lo que el reloj se ejecutará un poco más lento de lo que establecimos.

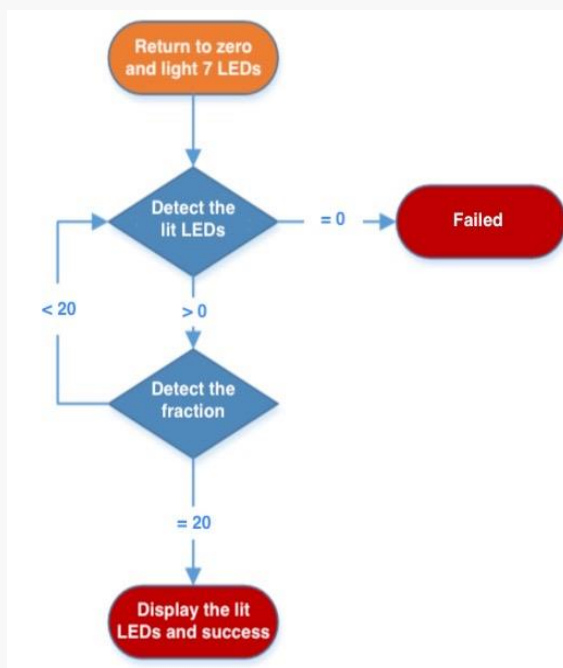


El siguiente programa se utiliza para ajustar el número de LEDs que se encienden.

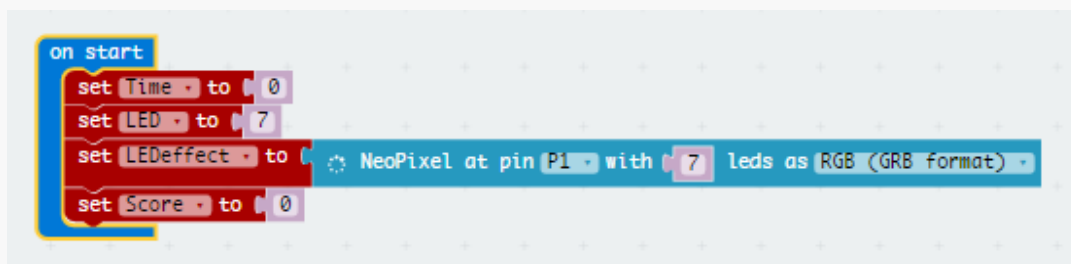




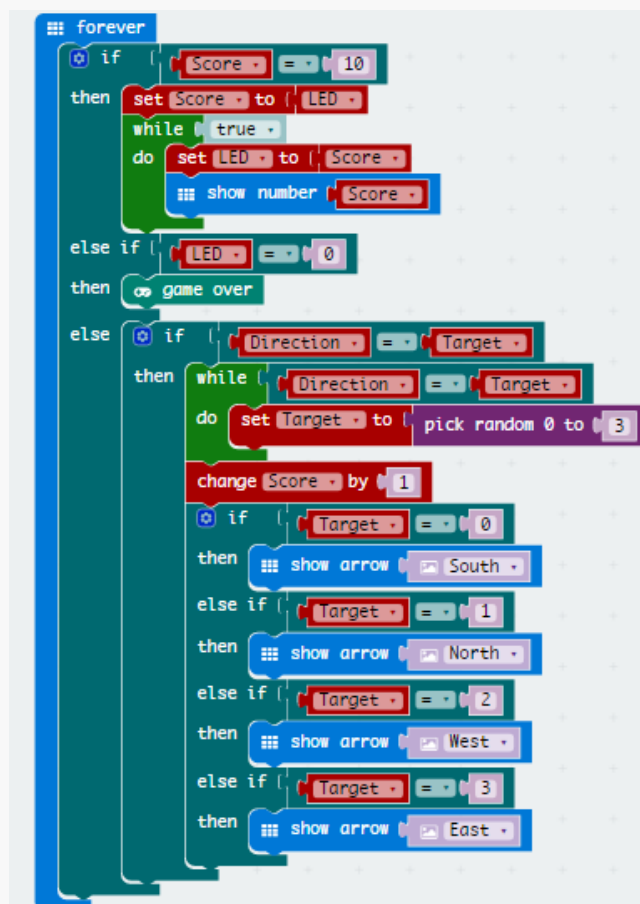
PASO 6: Se necesitará un mecanismo de puntuación para saber si logramos escapar. El siguiente diagrama de flujo le puede dar una idea de cómo funciona.



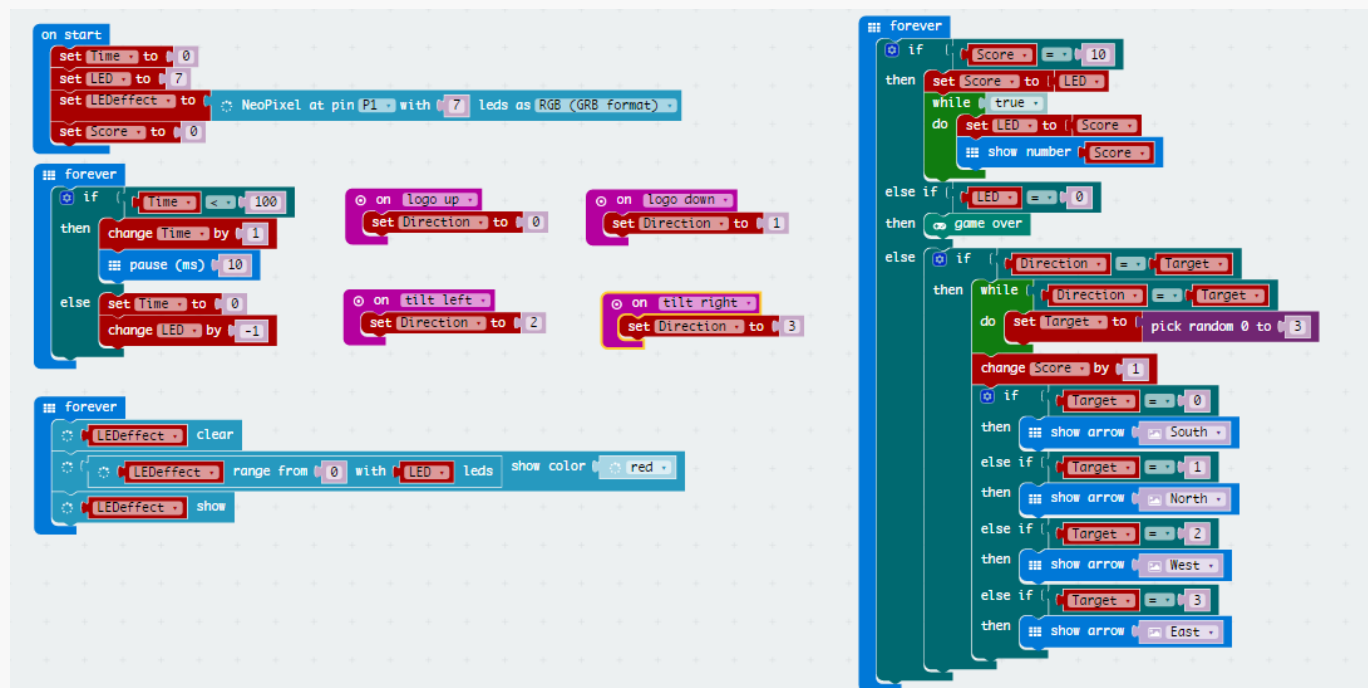
Recuerde reiniciar la puntuación al principio del programa.



PASO 7: Ahora, integre la función de puntuación al programa principal.



PASO 8: Así es como se ve el programa final



## Ejercicio

Intenta hacer que el reloj marque más rápido para aumentar la dificultad;

Programe los LEDs para que parpadeen a un ritmo basado en el tiempo restante;

Un reto poco intuitivo! Establezca diferentes direcciones en parejas y vea si todavía puede terminar el juego. Por ejemplo, necesitará colocar la parte superior para que coincida con una flecha a la izquierda, y la parte inferior para que coincida con una flecha a la derecha..

Tu viaje al micro:bit no se detendría aquí. Si desea compartir sus ideas con nosotros, por favor visite el blog de DFRobot!

**Bienvenido al blog de DFRobot!**

[www.dfrobot.com/blog](http://www.dfrobot.com/blog)