Microbit Y Smartcar

- Introducción Microbit
  - Introducción micro:bit
  - 1.- ¿Que es la micro:BIT?
  - 2.- ¿Cómo se programa la micro:BIT?
  - 3.- Características.
  - 4.- MakeCode. Primeros pasos.
- Programando la micro:bit
  - PROGRAMANDO LA MICRO:BIT
  - 5 Panel LED
  - 6 Sensor de temperatura
  - 7 Sensor de luminosidad
  - Reto 9 Medir la intensidad de la luz ambiente
  - 9 Acelerómetro
  - 10 Brújula
  - 11 Radio
  - 12 Pines de entrada y salida
- SmartCar. Dentro del curso Microbit+SmartCar
  - SmartCar
  - Reto 1 Musica
  - Sensor distancia
  - Sigue-lineas

Copyright 2026

    - Reto 5 Mando a distancia

- Basic kit. Fuera del curso \(de momento...\)
    - Basic kit
    - Reto 1 Movimiento de un servomotor

- Para finalizar...
    - Muro micro:BIT
    - Grupo Robotica Educativa
    - Pensamiento computacional
    - Enlaces de interés
    - Sobre el autor
    - ¿El curso continúa?
    - Créditos

Copyright 2026

Introducción Microbit

Copyright 2026

Introducción Microbit

Introducción micro:bit

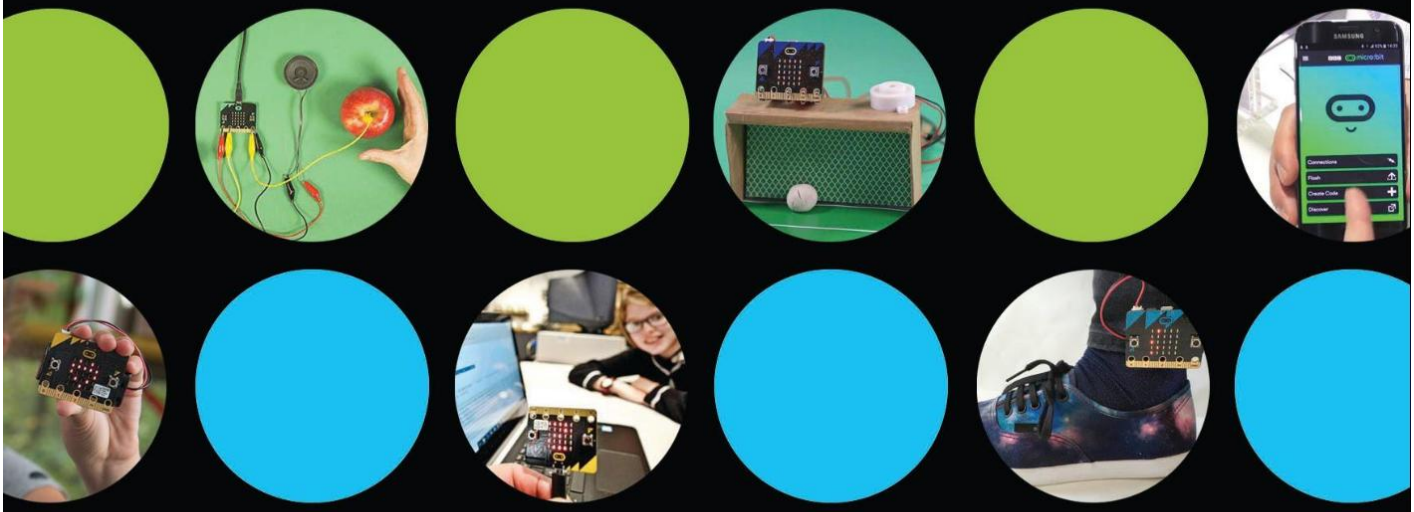
Copyright 2026

Guía de manejo • Resumen online gratuito

micro:es



Manual de Programación micro:bit



MakeCode

SCRATCH

open source hardware

José Francisco Muñoz & Microes.org (Comunidad micro:bit en España)

Copyright 2026

Introducción Microbit

1.- ¿Que es la micro:BIT?

La micro:BIT es una pequeña tarjeta programable, con un costo asequible a cualquier bolsillo (unos 19€, ojo el cable va aparte). Aun cuando su tamaño es muy reducido, incorpora gran cantidad de sensores y actuadores lo que unido a que usa un software Open Source, hacen de la micro:BIT una plataforma ideal para introducirse en el mundo de la programación de robots.

Kit de préstamo CATEDU

Si te matriculas en nuestros cursos, tienes derecho a un kit de préstamo. Ver condiciones en www.catedu.es en Robotica

El kit consta de :

- Placa Microbit
- Cable USB

Pero para este curso sería conveniente conseguir **otro micro:BIT** para hacer el reto **SENSOR INALÁMBRICO**



Te recomendamos ver este vídeo para que veas cómo es esta placa electrónica

https://www.youtube.com/embed/McXf6oF7ueA

A esta microbit se le puede añadir más complementos, como el coche del curso MICROBIT Y SMARTCAR

Introducción Microbit

2.- ¿Cómo se programa la micro:BIT?

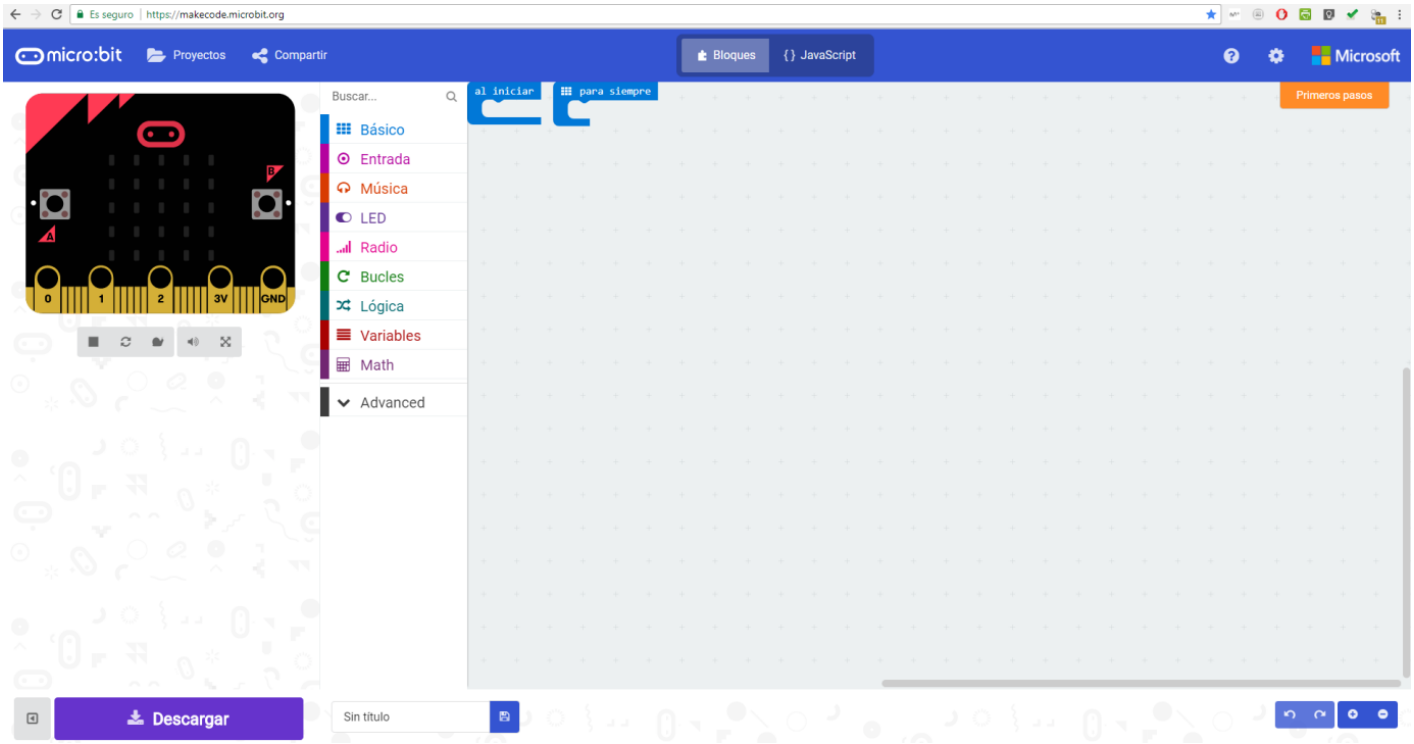
Hay varias plataformas que permiten codificar la micro:BIT, entre ellas destaca MakeCode, tanto en su versión online como offline.

La versión online es accesible desde este enlace:

https://makecode.microbit.org/

La versión offline se puede descargar de estos enlaces:

- Makecode offline portable para Windows, Macc, Android ..: Página Microes.org
- Otro enlace para Makecode offline Portable para Windows



Además la micro:Bit se puede programar con JavaScript, Pyton, Scratch (añadiendo una extensión) y Tickle (aplicación para iPad), para ver otros editores consulta https://microbit.org/es/code-alternative-editors/.



Introducción Microbit

3.- Características.

La micro:BIT incorpora:

- **25 LEDs**. Se pueden programar de forma independiente y permiten mostrar números, letras e imágenes. Si el texto o la cifra no caben en el display se desplazan de forma automática.
- **Sensor de Luz**. Los LEDs también tiene la posibilidad de ser usados como sensor de luz ambiente.
- **Pulsadores**. Existen 2 botones, etiquetados como A y B. Se puede detectar la pulsación independiente de cada uno de ellos así como la pulsación simultánea de ambos.
- **Conectores**. Situados en la parte inferior de la placa, dispone de 25 conexiones que permiten conectar otros sensores y actuadores. 5 de las conexiones (0,1,2 3v. y GND) se encuentran sobredimensionadas, para facilitar la conexión mediante pinzas de cocodrilo.
- **Sensor de temperatura**. Permite conocer a la micro:BIT la temperatura ambiente. Las unidades son los grados Celsius.
- **Acelerómetro**. Activada cuando se mueve la placa, permite conocer aceleraciones y giros a los que se somete la placa.
- **Brújula digital**. Permite conocer la desviación respecto el Norte Magnético. También permite detectar la presencia de campos magnéticos próximos. Al iniciar su uso entra en modo de calibración.
- **Radio**. Permite conectarse inalámbricamente con otras micro:BITs.
- **Bluetooth**. Ideal para conectarse e intercambiar datos inalámbricamente con otros dispositivos (móviles, tablets, ordenadores, etc) que dispongan de este tipo de conexión.
- **USB**. Usado para descargar los programas a la memoria de la tarjeta y para alimentar eléctricamente la micro:BIT.
- **Conector de batería**. Permite suministrar electricidad mediante dos pilas AAA o una batería. La tarjeta carece de interruptor, por lo que cuando se conecta la fuente de alimentación se ejecuta de forma automática el código que haya en memoria.

Guia de accesorios que se le pueden añadir (En CATEDU hemos creado el curso de MicroCar)

Puedes ponerle un protector :

https://www.youtube.com/embed/qGQJnxheZ1s

Copyright 2026

Y puedes poner un portapilas, incluso un protector, los hay baratos en Aliexpress (pon las palabras de búsqueda: *support batterie aaa microbit*)

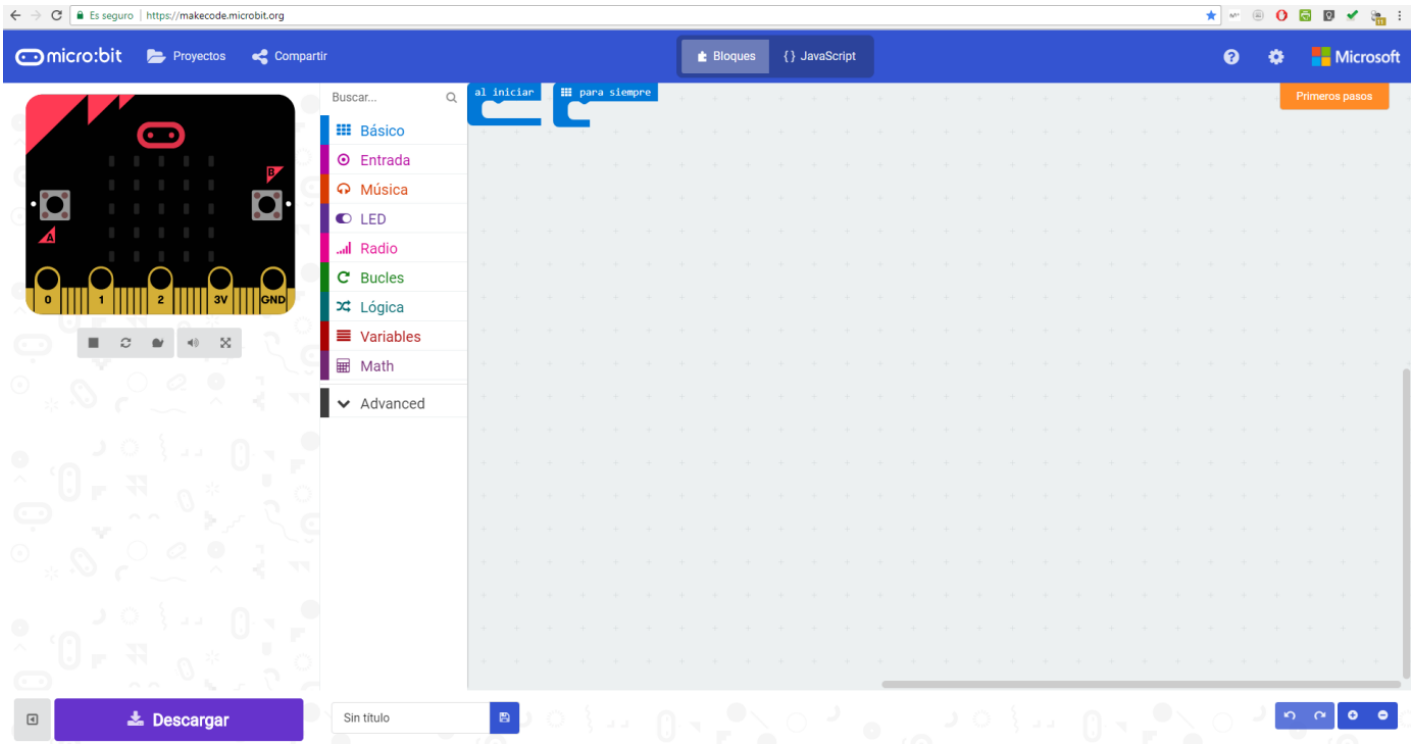


Introducción Microbit

4.- MakeCode. Primeros pasos.

Como se vio anteriormente MakeCode se puede ejecutar online, si está disponible conexión a Internet u offline.

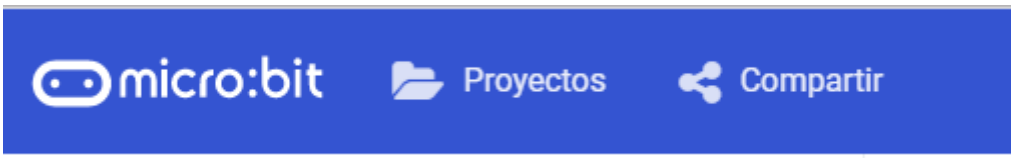
Para acceder directamente a la versión online de MakeCode para micro:BIT se debe introducir en el navegador la siguiente dirección: https://makecode.microbit.org/



A) Descripción General

1. **Simulador**. Situado en el lado superior izquierdo de la pantalla, nos muestra una micro:BIT, dónde se puede ver la ejecución del programa. Esta sección es útil para depurar el código antes de volcarlo a la memoria de la micro:BIT.

2. **Caja de Herramientas.** Situado en una columna vertical, al lado del simulador, allí se ubican los bloques de programación organizados por categorías. Al pulsar sobre el nombre de una categoría se abre, a la derecha, un desplegable donde aparecen los bloques más usados de la sección. Es importante fijarse en que justo debajo del nombre de la categoría, la mayoría de veces, aparece **la opción … Más**, si se pulsa se muestran el resto de bloques de la categoría. Una vez seleccionado el bloque a usar se debe arrastrar al área de programación.
3. **Área de programación.** Situado a la derecha de la Caja de Herramientas, es la zona donde se arrastran los bloques para crear el programa.
4. **Barras de Herramientas.** Situadas en la parte superior e inferior de la pantalla, ofrece atajos a la diferentes funcionalidades. En la parte superior:



Barra de herramientas

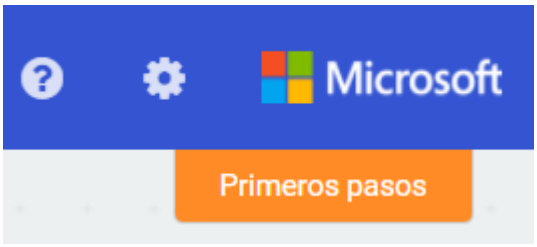
1.- Al pulsar sobre **micro:bit** lleva a la página https://microbit.org/code/ donde encontrar ejemplos de codificación en diferentes lenguajes de programación.

2.- La **sección Proyectos** permite elegir entre: 2.1. *Mis Cosas*. Aquí se puede crear un nuevo proyecto o importar un archivo previamente guardado, además de aparecer un listado de los programas más recientes, ordenados por fecha. 2.2. *Proyectos*. En esta sección se encuentran diferentes tutoriales, paso a paso, donde aprender a codificar la micro:BIT usando MakeCode. 2.3. *Ejemplos*. Se ofrecen varios códigos de ejemplo.

3.- **Compartir**. Permite al usuario compartir los códigos creados. En primer lugar se solicita “Publicar el Proyecto”, tras lo que aparece una nueva ventana, que ofrece un enlace y diferentes opciones de código para poder insertar en una página web.



4.- Permite intercambiar entre la visualización mediante bloques o ver el código escrito en JavaScript.



5.- En el extremo derecho de la Barra de Herramientas se encuentran las secciones de Ayuda, Configuración, acceso a la página principal de MakeCode y en naranja la sección primeros pasos, que ofrece un tutorial de uso. En la barra inferior aparecen los siguientes atajos:

6.- El primer icono de la izquierda, permite ocultar el simulador, lo que permite tener más espacio disponible en el Área de Programación.

7.- **Descargar**. Descarga el código a la micro:BIT. Se usará un nombre por defecto si no se asignó uno previamente.

8.- Pulsando sobre el icono del **disquete**, permite asignar un nombre y descargar el programa al disco duro.



9.- Las flechas permiten deshacer y rehacer los pasos de codificación.

10.- En el extremo derecho inferior aparece el icono “+” que aumenta el zoom del área de programación y el icono “-” que disminuye el zoom.

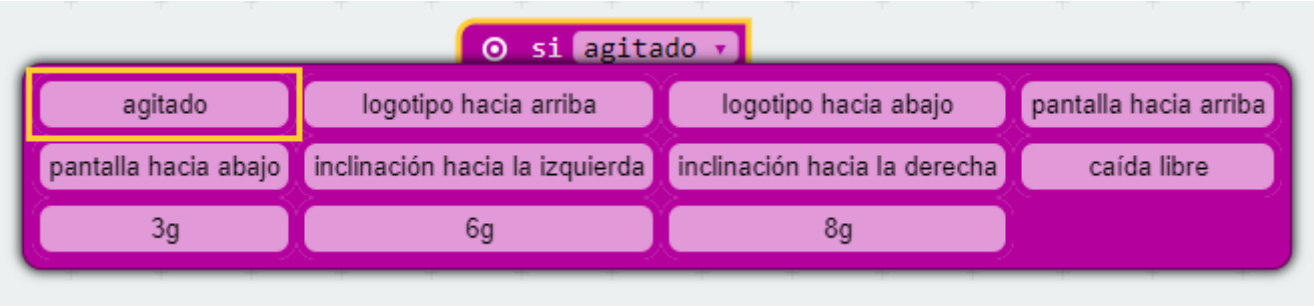


B) Cómo crear un programa con MakeCode

Lo primero que se necesita es decidir qué evento es el que inicia el código.

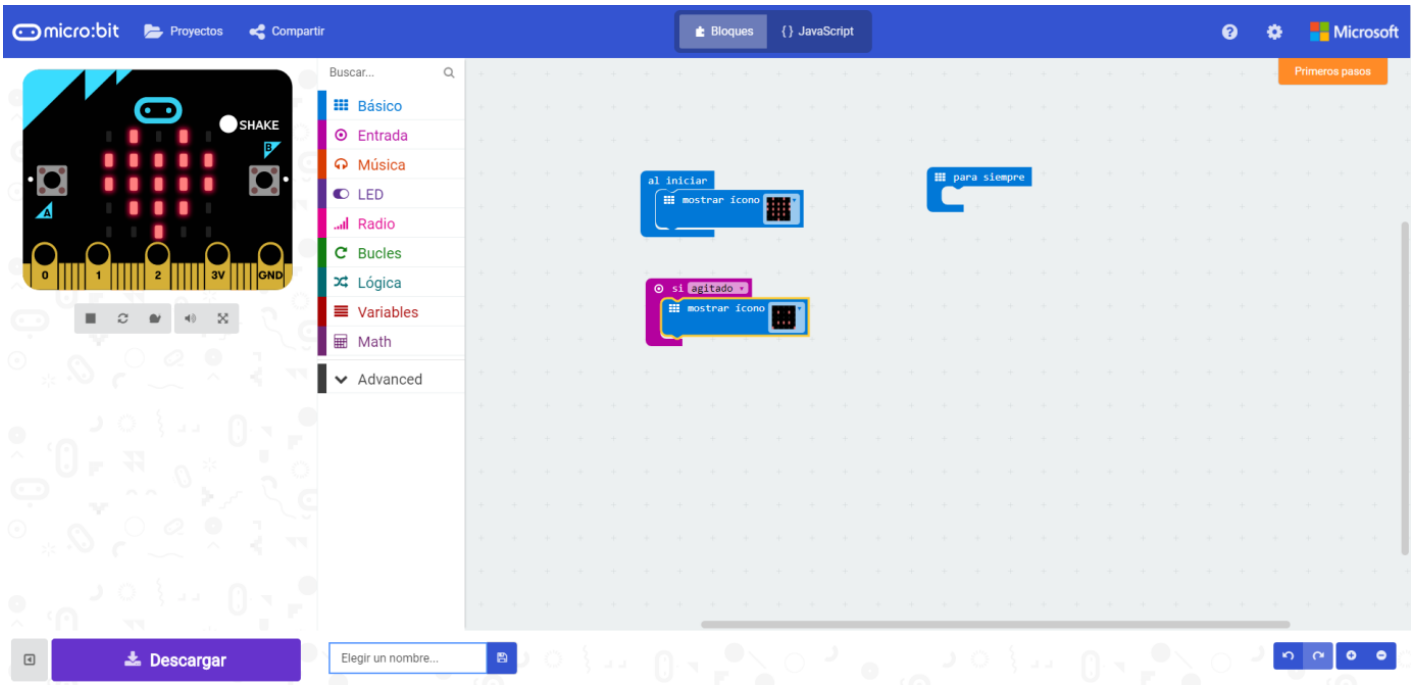
Por defecto, MakeCode, sitúa en el área de programación dos eventos con los que iniciar el código, estos son: al iniciar y para siempre.

También existen otros eventos que permiten iniciar el programa, situados en la categoría Entrada, son: al presionar el botón, al presionar pin o si agitado además permiten diferentes opciones.



Para crear el programa, se deberá ir arrastrando los diferentes bloques al área de programación.

Se podrá comprobar el funcionamiento del programa en el simulador situado a la izquierda. Además según los bloques introducidos, aparecerán diferentes opciones que permiten interactuar con el simulador.



C) Cómo descargar un código a la micro:BIT

Tras conectar la micro:BIT al ordenador, aparece esta como una nueva unidad. Terminado el programa y comprobado su correcto funcionamiento, se debe pulsar sobre el icono de descarga, lo que copia el código al ordenador con un nombre por defecto. Otra opción es introducir el nombre

elegido en la caja situada al lado del icono del disco y pulsar sobre este para descargar. Ya solo queda copiar el fichero con extensión .hex a la unidad que apreció tras conectar la micro:BIT al ordenador.



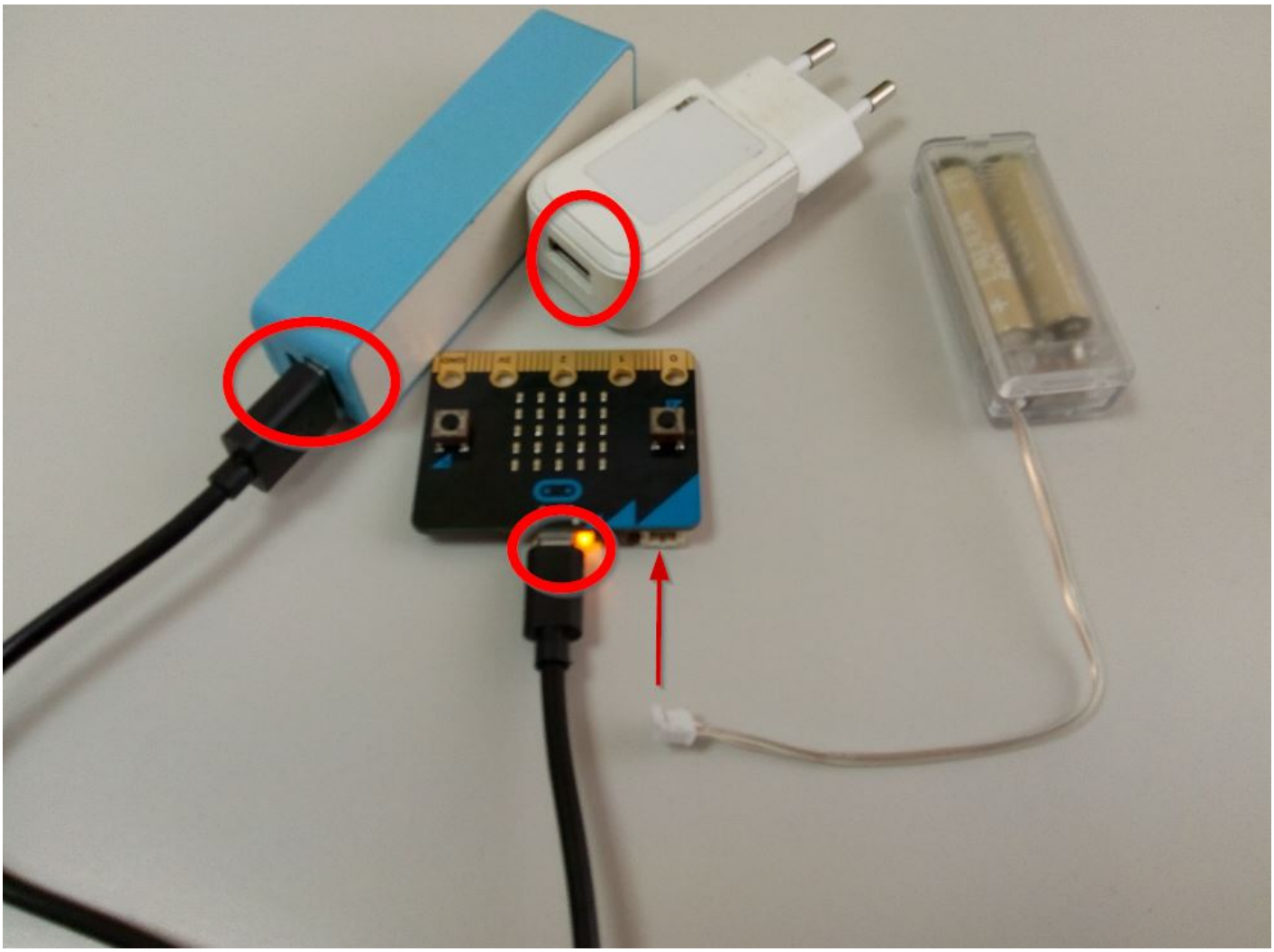
ATENCIÓN PARA USUARIOS LINUX Y VITALINUX

Consultar la wiki

https://wiki.vitalinux.educa.aragon.es/index.php/Rob%C3%B3tica_con_Vitalinux#Microbit

D) Cómo ejecutar el código.

Copiado el código, el programa comienza a ejecutarse de forma automática. Si se quiere usar la micro:BIT desconectada del ordenador, se debe conectar un pack de pilas o batería al conector situado junto al conector microUSB e inmediatamente se ejecutará el código descargado.



Copyright 2026

Programando la micro:bit

Copyright 2026

Programando la micro:bit

PROGRAMANDO LA MICRO:BIT

En este apartado se van a mostrar algunos ejemplos de cómo programar los diferentes sensores y actuadores que componen la micro:BIT. Tal como se explicó en el capítulo anterior, los códigos creados se podrán comprobar inicialmente en el simulador de la página de MakeCode. Posteriormente el código se puede descargar a la memoria de la micro:BIT y ejecutar de forma autónoma.

RPi learning Getting Started with the micro-bit

By Raspberry Pi Foundation - https://www.raspberrypi.org [CC BY-SA 4.0], via Wikimedia Commons

Programando la micro:bit

5 Panel LED

La tarjeta consta de 25 LEDs de color rojo, estos se pueden programar de forma independiente. Permite mostrar números, letras e imágenes. El sistema está preparado para desplazarse de forma automática en el supuesto de que no quepan los caracteres a mostrar.

Micro bit position des DEL By Ugopedia [CC BY-SA 4.0], from Wikimedia Commons

Reto Hola Mundo.

Cuando se aprende a programar, se llama “Hola Mundo” al primer programa que se crea. Este código imprime el texto «¡Hola Mundo!» en un dispositivo de visualización.

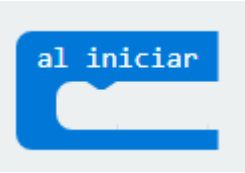
Objetivo.

Al conectar la micro:BIT, se debe mostrar el texto «¡¡¡Hola Mundo!!!» en el panel LED de la tarjeta.

https://www.youtube.com/embed/eqBcjWYTqNE

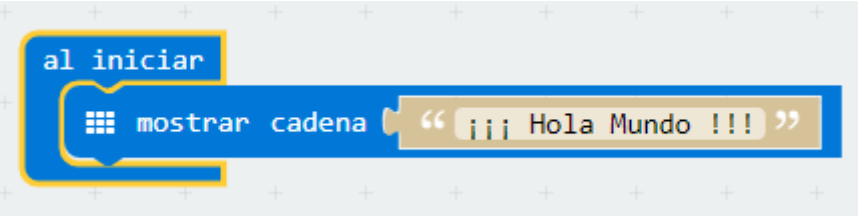
Descripción del código.

Se propone usar el evento al iniciar, situado en la categoría Básico. Este bloque permite ejecutar el código introducido, cuando se inicia la micro:BIT



A continuación se usará el bloque mostrar cadena, también localizado en la categoría Básico e introducir el texto “¡¡¡ Hola Mundo !!!”. Este bloque permite mostrar caracter a caracter, el texto introducido.

El código quedará de la siguiente manera.



De forma inmediata comenzará a leerse en el panel LED la cadena de texto introducida. El texto no cabe en su totalidad por lo que este se desplazará de forma automática.

Simulador

Lo bueno que tiene el editor es que permite compartirlo y embeberlo en una página web. Aquí tienes el resultado de embeber el anterior código:

https://makecode.microbit.org/---run?id=_3bPEot71XXJA

Propuesta.

Modificar el código, para que se muestre el nombre del programador.

Reto Animando los iconos.

La animación es una técnica que logra crear sensación de movimiento a imágenes estáticas. Se consigue mediante una secuencia de imágenes ordenadas, que al ser mostradas consecutivamente, consiguen generar la ilusión visual de movimiento.

Objetivo.

Se propone, que al conectar la micro:BIT, se muestre en el panel LED un corazón que late.

https://www.youtube.com/embed/Cf2ov3A-ICE

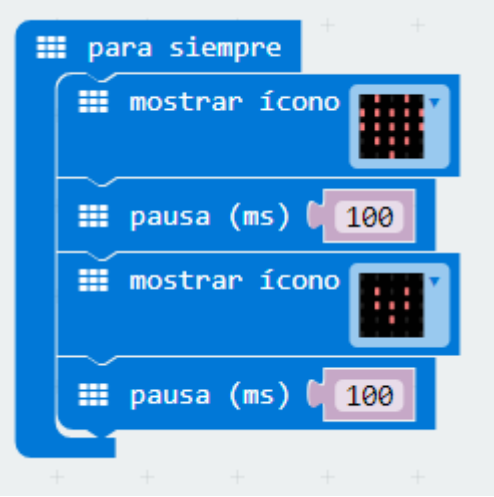
Descripción del código.

Se usará el evento “para siempre”, situado en la categoría Básico..

Situar en el interior el bloque anterior la instrucción mostrar icono, usar el llamado corazón.

Agregar el bloque pausa ms (100) localizado en la categoría Básico.

Repetir los dos bloques anteriores, pero seleccionando el icono “corazón pequeño”.



De forma inmediata se visualiza en el simulador un corazón que simula latir.

Simulador

https://makecode.microbit.org/---run?id=_fxMcfc0KHUmd

Propuesta.

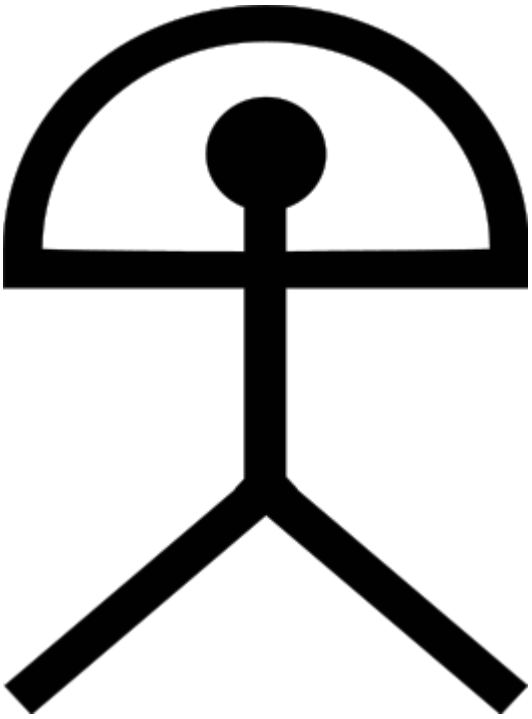
Hacer un código que muestre la animación de una tijera cortando.

Reto 3 Diseñando iconos.

MakeCode permite diseñar de forma gráfica los iconos a mostrar en el display LED. Para este menester dispone de un bloque específico situado en la categoría Básico. El bloque se llama “mostrar LEDs” y permite al usuario diseñar sus propios iconos marcando de forma independiente los LEDs que se quieren iluminar.

Objetivo.

Diseñar un icono con forma de Indalo. Se trata de una figura rupestre encontrado en una cueva de la provincia de Almería y que representa una figura humana con los brazos extendidos y un arco sobre sus manos. Actualmente se considera un símbolo de la provincia de Almería.

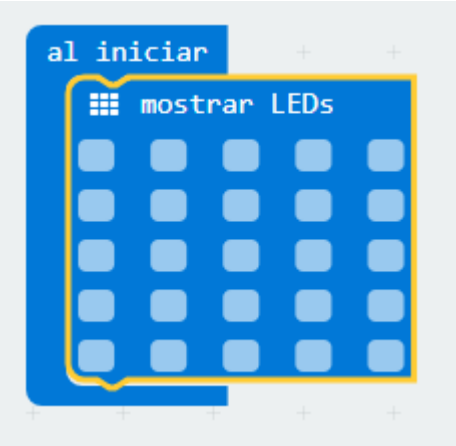


https://www.youtube.com/embed/ag2ulUyaQqg

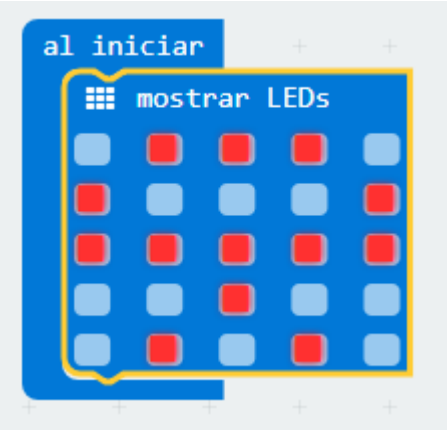
Descripción del código.

Se usará el evento al iniciar, situado en la categoría Básico..

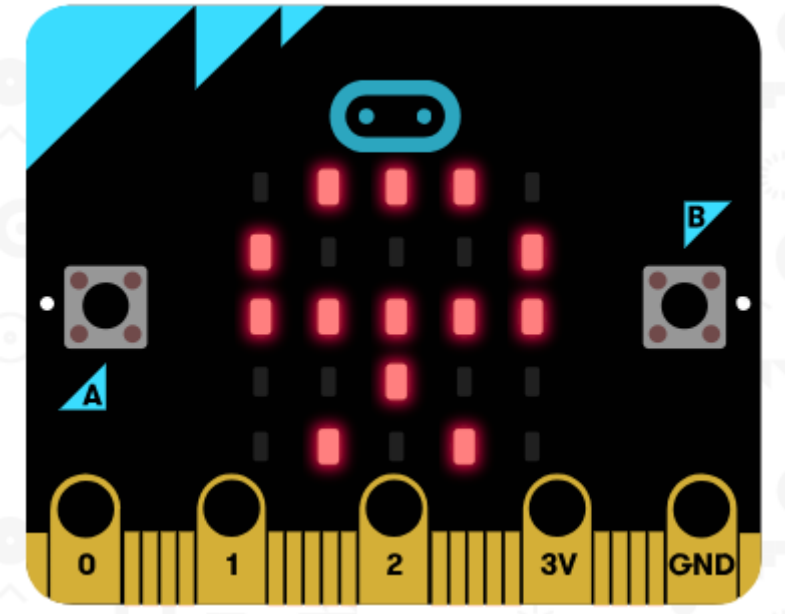
Situar en el interior el bloque anterior, la instrucción “mostrar LEDs”, este se localiza en la categoría Básico. Este bloque permite marcar de forma independiente los LEDs que se quieren encender.



Pulsar sobre los LEDs que se quieren iluminar hasta conseguir el icono deseado.



Deberá aparecer en el simulador el icono con la forma diseñada.



Propuesta.

Diseñar un código que muestre un logotipo diseñado por el programador.

Copyright 2026

Programando la micro:bit

6 Sensor de temperatura

El sensor de temperatura integrado en la placa detecta la temperatura ambiente en grados Celsius.

CelsiusKelvin

By Celsius_kelvin_estandar_1954.png: Homo logosderivative work: kismalac (Celsius_kelvin_estandar_1954.png) [CC BY-SA 3.0 or GFDL], via Wikimedia Commons

Reto Termómetro digital.

Un termómetro es un instrumento que sirve para medir la temperatura; el más habitual consiste en un tubo capilar de vidrio cerrado y terminado en un pequeño depósito que contiene una cierta cantidad de mercurio o alcohol, el cual se dilata al aumentar la temperatura o se contrae al disminuir y cuyas variaciones de volumen se leen en una escala graduada.

Los termómetros digitales son aquellos que, valiéndose de dispositivos transductores, utilizan circuitos electrónicos para convertir en números las pequeñas variaciones de tensión obtenidas, mostrando finalmente la temperatura en un visualizador.

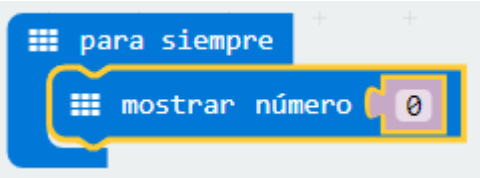
Objetivo.

Mostrar en el panel LED la temperatura detectada por el sensor de temperatura integrado en la micro:BIT.

https://www.youtube.com/embed/Avl3wo4Rkww

Descripción del código.

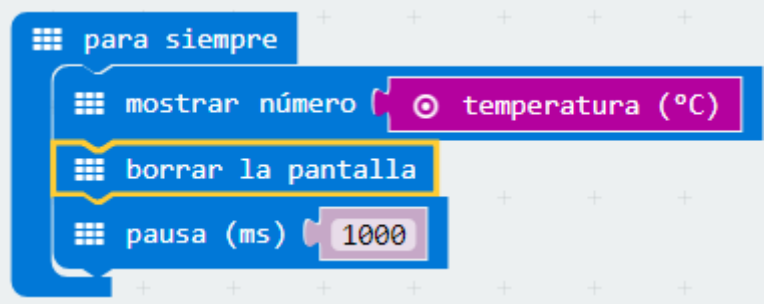
Se propone iniciar el programa usando el evento “para siempre”. Añadir el bloque, localizado en la categoría Básico, mostrar número. Este bloque muestra en la pantalla LED el número introducido, desplazándose si es mayor de 1 cifra.



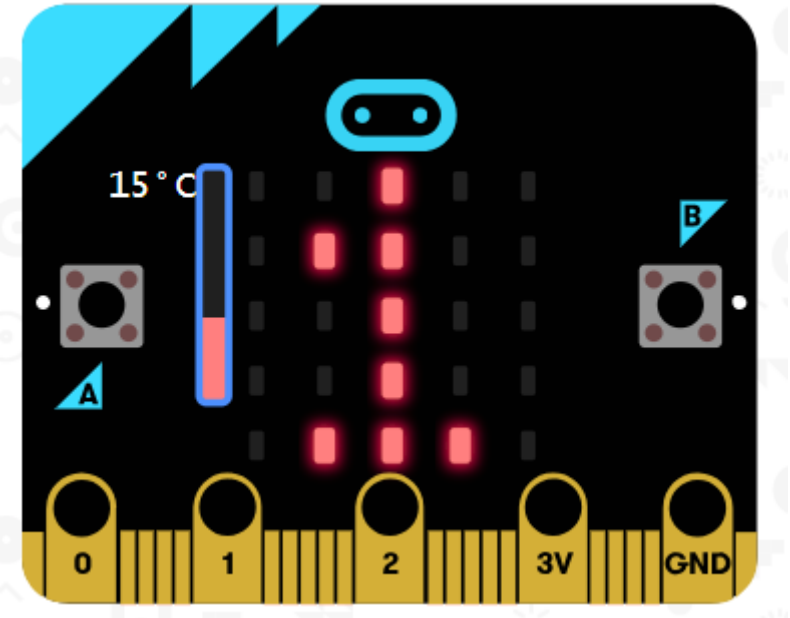
Sustituir el número “0” por el valor obtenido del sensor de temperatura. Este bloque se localiza en la categoría Entrada.



A continuación se borra la pantalla y se introduce una pausa de 1000 ms. Quedando el programa de la siguiente forma:



En el simulador aparecerá una barra vertical que permite modificar la temperatura, siendo la temperatura marcada la mostrada en el panel LED.



Propuesta.

Hacer un programa que al pulsar el botón A muestra la temperatura en grados celsius y al pulsar el botón B la temperatura mostrada sea en grados Fahrenheit.

Reto Aviso de placas de hielo.

La mayoría de coches incorporan en el salpicadero, junto a la pantalla que marca la temperatura exterior, un testigo con forma de copo de nieve, que avisa de la posibilidad de que haya placas de hielo en la carretera. Este testigo se suele iluminar cuando la temperatura baja de 3 grados centígrados.

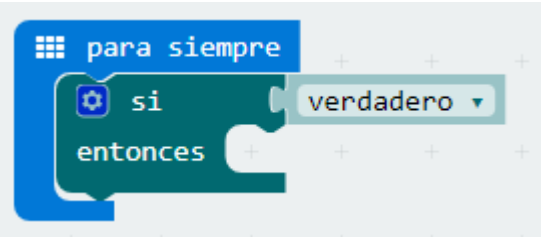
Objetivo.

Mostrar en el panel LED de la micor:BIT un icono con forma de copo de nieve cuando la temperatura esté por debajo de 3 grados.

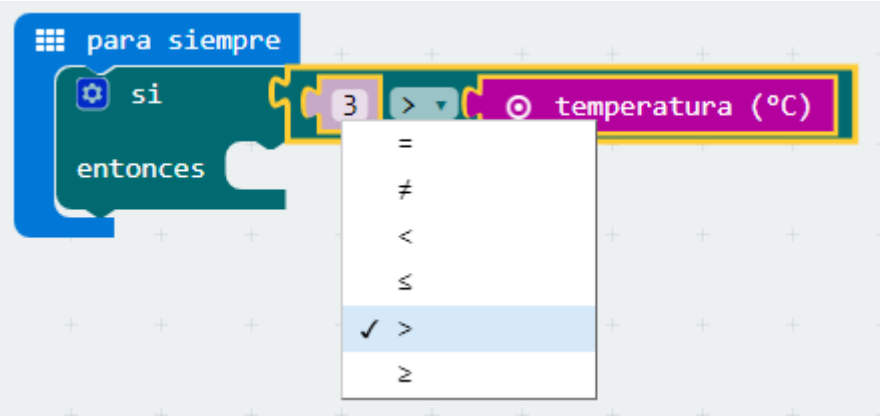
https://www.youtube.com/embed/T8Ely3YaHhg

Descripción del código.

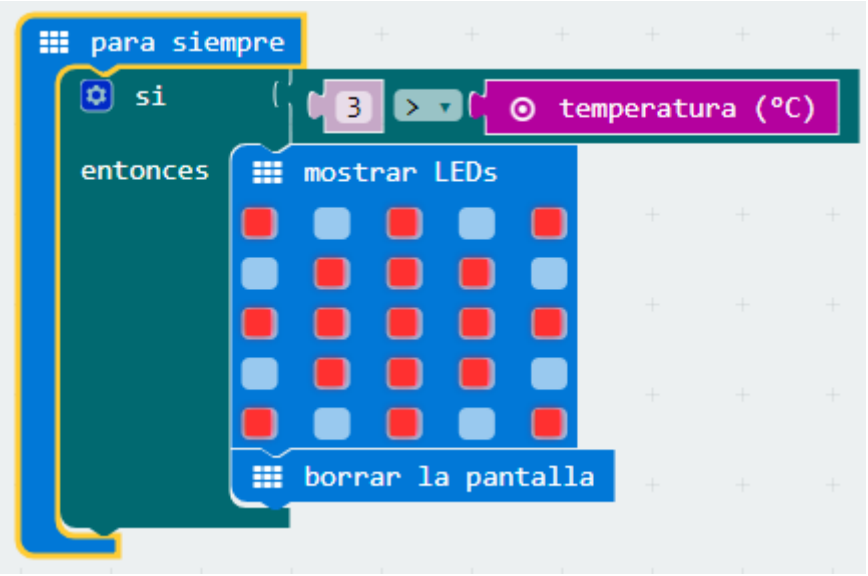
Para iniciar el programa, se propone usar el evento para siempre. Posteriormente se debe añadir el operador lógico si … entonces situado en la sección Lógica. Este bloque verifica si es verdad que se cumple una condición, si es así, ejecuta las instrucciones introducidas.



Para comprobar si la temperatura está por debajo de los 3º centígrados se usará el operador que devuelve verdadero siempre que 3 sea mayor que el valor reportado por el sensor de temperatura. El bloque se localiza en la categoría Lógica.

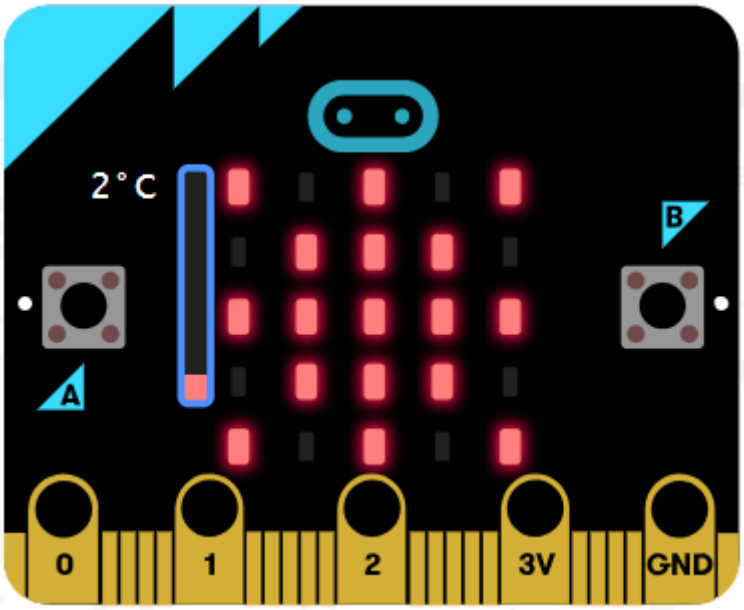


Para mostrar un icono con forma de copo de nieve se usará el bloque mostrar LED, también se debe añadir el bloque borrar la pantalla para que no se quede encendido el panel LED cuando suba la temperatura de 3º centígrados



Una vez terminado el código, aparecerá el símbolo de hielo, cuando la temperatura sea menor de 3º centígrados.

Copyright 2026



Propuesta.

Añadir un aviso sonoro cuando la temperatura sea inferior a 3 grados celsius.

Reto Temperatura óptima de una nevera

La temperatura óptima de un frigorífico es de 7ºC, mientras que la temperatura de un congelador debe estar entorno a los -18ºC. Algunas neveras incorporan un avisador acústico que se activa cuando la temperatura no es la óptima.

Objetivo.

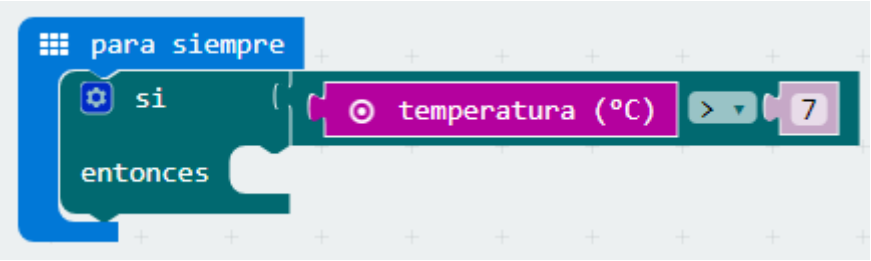
Mostrar en el panel LED de la micro:BIT la temperatura del frigorífico y activar una alarma cuando la temperatura esté por encima de 7ºC

https://www.youtube.com/embed/AJHt2rd805E

Descripción del código.

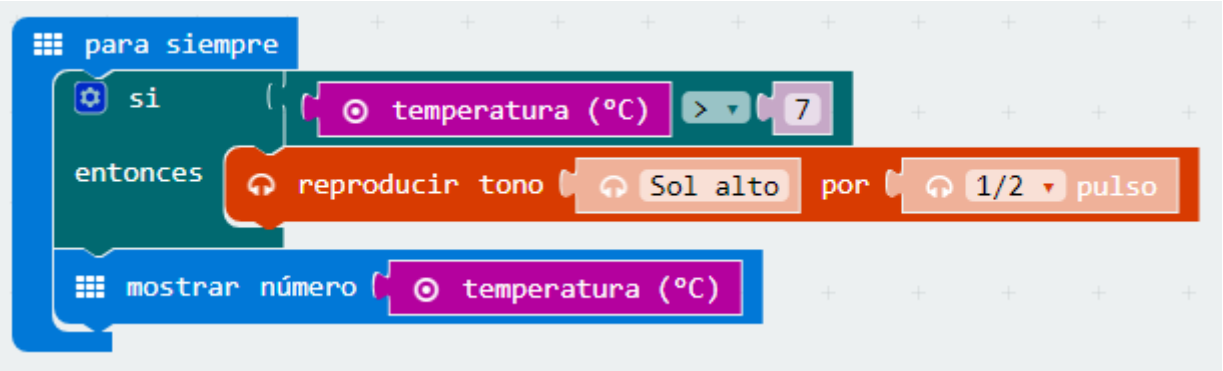
En este reto se incorpora un altavoz como actuador externo. Se conectará uno de los polos al GND y el otro al PIN 0.

Para iniciar el programa se usará el bloque “para siempre”. Dentro se situará el condición “si entonces”. La condición a verificar será: si la temperatura es mayor de 7 grados.



Si el valor es verdadero entonces se reproducirá un tono de aviso.

También se muestra la temperatura medida.



En el simulador se puede variar la temperatura para comprobar el correcto funcionamiento del programa.

Propuesta.

Modificar el código para monitorizar el congelador.

Programando la micro:bit

7 Sensor de luminosidad

Los LEDs de la placa micro:BIT también pueden actuar como entrada haciendo que detecten la luz ambiente.

Artist's rendition of the new light-bending metamaterial

By Photo Credit: Keith Drake [Public domain], via Wikimedia Commons

Reto Interruptor crepuscular.

Cuando la intensidad de la luz cae por debajo de un nivel de iluminación predeterminado y ajustado con anterioridad, el interruptor crepuscular enciende la iluminación. Por el contrario, si la intensidad de la luz es mayor que el nivel prefijado, los interruptores apagan la iluminación.

Objetivo.

Crear un programa que mida el nivel luminoso existente, para ello usará el sensor de luz de la micro:BIT. Posteriormente se comprobará si este valor es inferior a 50, si es así, se encenderán todos leds de la placa a su máxima intensidad.

https://www.youtube.com/embed/xZu3Cxfzf0s

Descripción del código.

Se usará el evento para siempre, para iniciar el programa. Se introducirá la condición si … entonces, si no, para comprobar si se está por encima o por debajo del valor teórico tomado como límite en esta ocasión 50. Si el valor es inferior a 50, se fijará la intensidad de los LEDs al máximo usando el bloque ajustar brillo 255. Si el valor es superior a 50 se usará el bloque borrar la pantalla para, de esta forma, apagar todos los LEDs.

El código quedaría de la siguiente forma, se podrá comprobar su funcionamiento variando la intensidad en el simulador de la tarjeta.



Brillo adaptativo

La matriz de LED de micro:bit es también un sensor que puede captar tanto la luz visible como la invisible. En este último caso se encuentra la luz infrarroja emitida por los mandos a distancia.

Los valores de la variable **nivel de luz** del menú **Entrada** se encuentran comprendidos entre 0 y 1023. Como en el caso del sensor de sonido, estos valores **no tienen correspondencia con ninguna unidad física**, como el lux.

> Para poder obtener buenas medidas de luz, la matriz de LED debe estar descubierta. Las fundas para micro:bit, especialmente las de color oscuro, pueden alterar las medidas del sensor de luz.

De acuerdo con Martínez de Carvajal (2019), el sensor de luz arrojará aproximadamente las siguientes medidas:

- 0 en condiciones de oscuridad absoluta.
- 100 en una habitación con luz artificial.
- 400 en un día muy nublado.
- 1023 a pleno sol.

Podemos retomar el proyecto del sonómetro y acompañarlo de una pantalla con brillo adaptativo, como la de los teléfonos móviles. La pantalla debe iluminarse con más brillo cuanto mayor sea la luz ambiental.

Copyright 2026

El evento **al presionarse el botón A** medirá el nivel de sonido y lo mostrará gráficamente con la ya conocida barra.



Usaremos un evento de tiempo **cada ms** para medir la luz ambiental cada segundo y así poder ajustar el brillo de pantalla al nivel de luz medido. De esta forma, a mayor luz ambiental, mayor brillo de pantalla. Como el brillo máximo es de 255 y el nivel de luz máximo medible es de 1023, dividiremos la lectura del nivel de luz entre 4 usando un bloque aritmético **/** del menú **Matemática** .



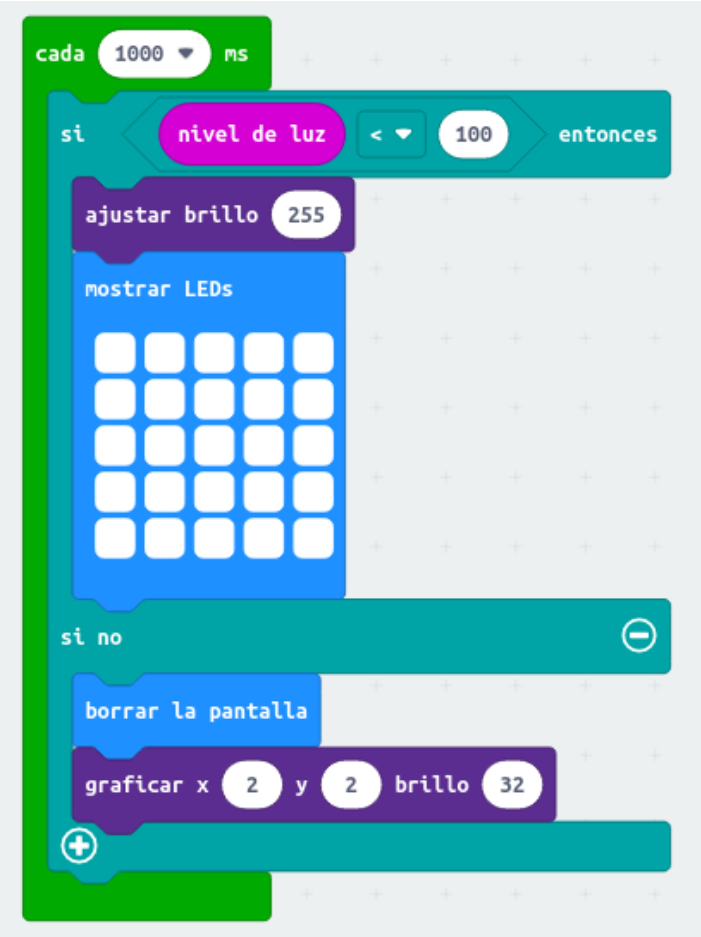
Luz crepuscular

Vamos a dar solución a un reto propuesto por Muñoz (2022) programando un automatismo que encienda la pantalla de **LED** cuando la luz ambiental caiga por debajo de un cierto valor, por ejemplo 100.

Todos los algoritmos programados hasta este momento se ejecutan linealmente desde el primer bloque colocado en la parte superior hasta el último situado abajo. Sin embargo, para solucionar el reto de la luz crepuscular necesitaremos que el flujo del programa siga caminos distintos según se cumpla o no una determinada condición.

Empezaremos evaluando la luz ambiental cada segundo mediante un evento de tiempo **cada ms**. Esto significa que el automatismo tendrá un tiempo de reacción de un segundo.

Dentro del evento de tiempo, usaremos la estructura **si entonces si no**, disponible en el menú **Lógica** , para encender con el máximo brillo la luz de la matriz de **LED** si la luz ambiental es inferior a 100 o para apagar la matriz en caso contrario. Tras el apagado de la pantalla dejaremos un pequeño punto en el centro iluminado a baja intensidad como piloto de funcionamiento.

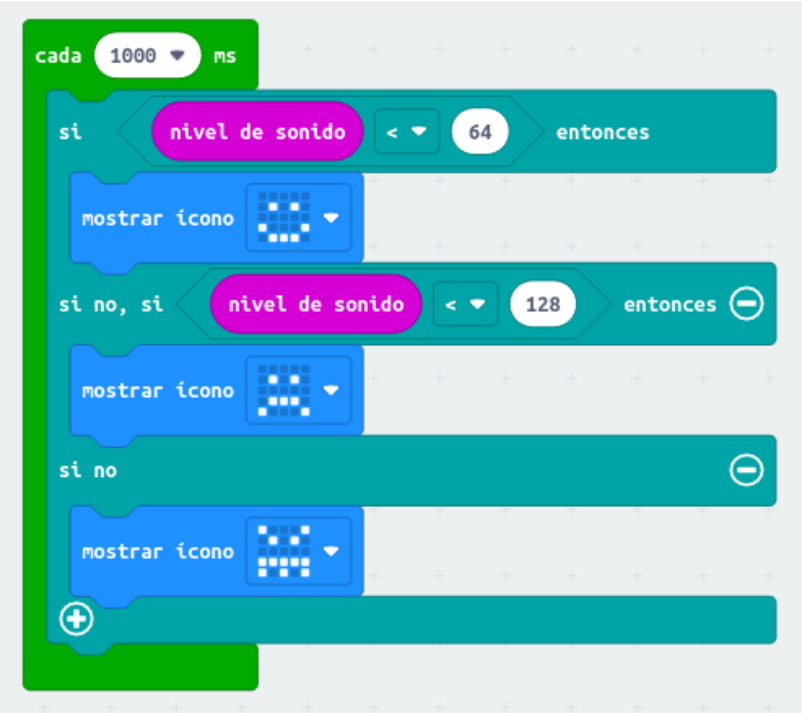


El bloque comparador **<** (menor que) usado dentro de la estructura **si entonces** también se encuentra dentro del menú **Lógica**.

Semáforo del ruido en el aula

Make Code permite construir sentencias condicionales más complejas. El código del semáforo del ruido emplea un bloque **si entonces si no, si entonces si no**. Para construir este bloque se ha utilizado la estructura **si entonces si no** del ejemplo anterior y se ha añadido una condición suplementaria pulsando sobre el pequeño círculo con el símbolo +. De igual manera, pulsando sobre los círculos con los signos - podremos eliminar condiciones.

Copyright 2026



Así pues, si el nivel de sonido es inferior a 64, la pantalla mostará un icono sonriente. En caso contrario, si el nivel de sonido es alto pero todavía inferior a 128, se mostrará una cara triste. FInalmente, si el sonido es igual a superior a 128, se mostrará una cara enfadada.

Copyright 2026

Programando la micro:bit

Reto 9 Medir la intensidad de la luz ambiente

Los sensores de luz se usan para detectar el nivel de luminosidad y producir una señal de salida representativa de la cantidad de luz detectada.

En algunas ocasiones puede ser interesante conocer el nivel lumínico que hay en diferentes situaciones y así poder calibrar un dispositivo que dependa del valor de la intensidad luminosa, como puede ser una cámara fotográfica.

MakeCode dispone del bloque nivel de luz en la categoría Entrada, que lee el valor de luz aplicado en la pantalla LED. Este valor se mueve en un rango de 0 (oscuro) a 255 (luminoso).

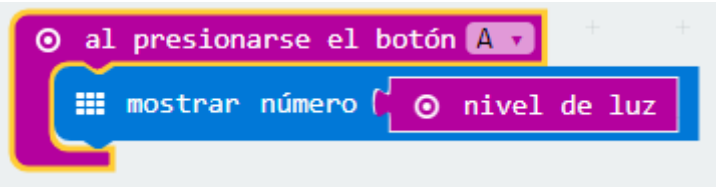
Objetivo.

Al presionar sobre el pulsador A se mostrará el valor numérico de la intensidad de la luz ambiente. Al pulsar sobre B, se mostrará gráficamente la intensidad lumínica. Al pulsar A y B de forma simultánea, se apagará la pantalla.

Descripción del código.

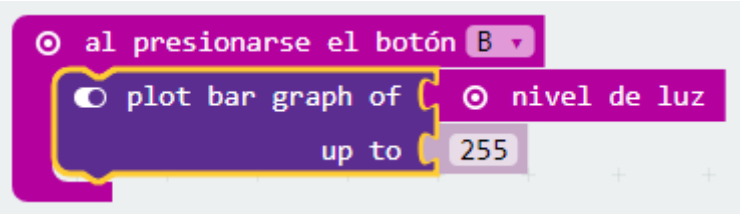
Se propone crear tres programas diferentes y que cada uno se inicie según el botón o combinación de botones presionados.

Al pulsar sobre el botón A se mostrará el valor numérico de la intensidad luminosa. Se usará el bloque mostrar número introduciendo el bloque nivel de luz como valor a presentar.

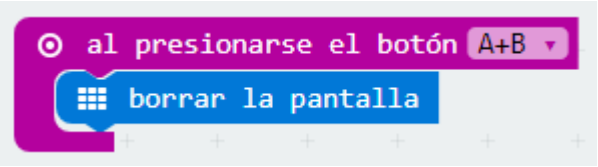


Al pulsar el botón B se encenderán los LEDs necesarios, para mostrar de forma gráfica el valor del nivel de luz. Se usará el bloque plot bar graph of ... up to ... localizado en la sección LED. En primer

lugar se introduce el bloque nivel de luz y en el segundo término se sustituye el valor por defecto por 255, valor máximo que detecta el sensor de luz.



Para terminar se usará la pulsación combinada de los botones A y B para borrar y apagar la pantalla LED.



Propuesta.

Crear un código que permita encender todos los LEDs del panel de la micro:BIT al pulsar el botón A y apagarlos al pulsar el botón B, pero no se encenderá si el valor del sensor de luz es superior a 175.

Reto Modificar el brillo de los LEDs

La mayoría de los teléfonos móviles incorporan la posibilidad de modificar la luminosidad de la pantalla, para adaptarse a la luz ambiente.

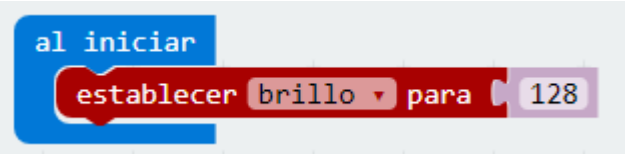
Objetivo.

Diseñar un programa que permita modificar la luminosidad de los LEDs, si se pulsa el botón A la luminosidad aumentará, si se acciona el botón B la luminosidad disminuirá. Si se pulsan A+B la luminosidad se situará a un valor intermedio (128).

Descripción del código.

Para iniciar el programa se definirá una variable que se llamará “brillo”. En la categoría Variables se podrá crear una nueva variable o renombrar la que sale por defecto.

Se establecerá un valor inicial de 128 para la variable creada.

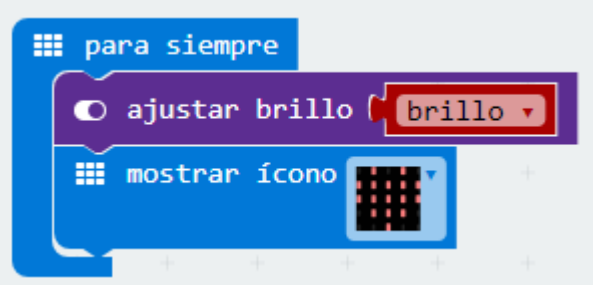


Se propone usar pasos por valor de 32 unidades para evidenciar el cambio de brillo en los LEDs, pero se puede usar cualquier otro valor.

Al accionar el pulsador A se aumentará la variable “brillo” en 32 pasos. Al pulsar el botón B se disminuye el valor de la variable “brillo” en 32 pasos. Por último si se pulsa a la vez las teclas A y B se establece el valor inicial de la variable “brillo”.



Ya para terminar faltaría mostrar en el panel LED un icono que muestre los cambios en el brillo de los LEDs. Se debe usar la variable “brillo” como valor que toma el bloque ajustar brillo



Propuesta.

Se propone como reto hacer un código que ajuste el brillo de forma automática, a mayor luminosidad medida por el sensor de luz, mayor valor del brillo de los LEDs.

Programando la micro:bit

9 Acelerómetro

El acelerómetro mide la aceleración de tu micro:bit. Se activa cuando tu placa se mueve y también puede detectar otras acciones como agitar, girar y hasta soltar tu micro:bit en caída libre!

Micro bit 3 axes de l accéléromètre

By Ugopedia [CC BY-SA 4.0], from Wikimedia Commons

Reto Dado electrónico.

Un dado es un objeto cúbico usado en juegos de azar. En las caras del cubo aparecen puntos que representan distintos números.

Se propone crear un dado electrónico que, usando el panel LED, muestre el resultado de obtener un número al azar entre 1 y 6.

Objetivo.

Tras agitar la micro:BIT, se deberá calcular un número aleatorio cuyo valor esté entre 1 y 6. Posteriormente, se mostrará en el panel LED el valor obtenido.

Pruébalo ¡¡lo puedes sacudir pulsando en *Shake*!! este simulador es genial !!

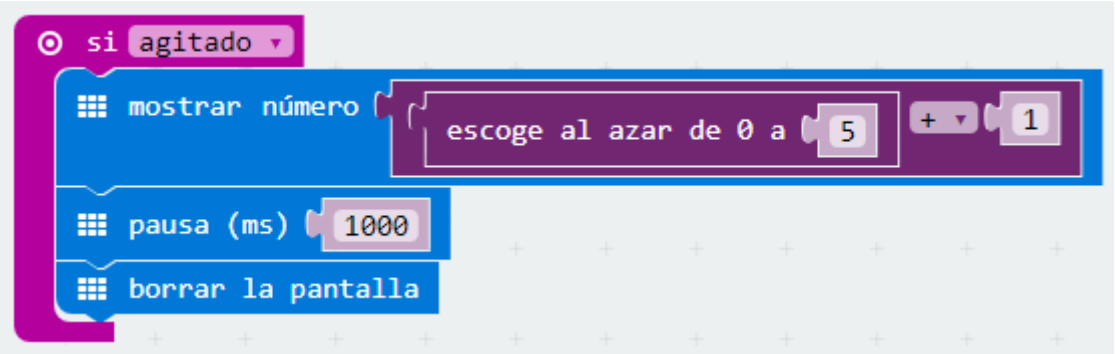
https://makecode.microbit.org/---run?id=_A1Tcyy1zLRgP

Descripción del código.

Como evento de inicio del programa se usará el bloque si agitado, se mostrará un número escogido al azar entre 0 y 5, al que se le sumará 1. Se suma 1 dado que el bloque escoge al azar, tiene como valor fijo de inicio el 0.

Tras una pausa de 1 segundo se borra la pantalla, quedando a la espera de que se vuelva agitar la micro:BIT.

Copyright 2026



Propuesta.

Diseñar un programa que permita jugar a la ruleta además se deberá indicar si es rojo o negro o si gana la banca.

Reto Cara o cruz.

El juego de cara o cruz es un juego de azar en el que se emplea una moneda. Gana quien acierte qué lado de la moneda (de los dos posibles) caerá cara arriba. Cada uno de los dos lados tiene un nombre distintivo y son mencionados como opciones para ser elegidas por los participantes. Por lo tanto, los participantes pueden ser dos equipos o dos personas y así cada bando tiene el 50% de probabilidades de acertar.

Este juego es frecuentemente empleado como un mecanismo para tomar decisiones o sortear, ya sea entre amigos o en forma informal, pero también es completamente aceptado en ciertas circunstancias, por ejemplo al inicio de un partido de fútbol profesional.

Objetivo.

La micro:BIT se situará con el panel LED hacia abajo. Al posicionarlo hacia arriba, se mostrará una cara o una cruz, ganando el que haya escogido la opción mostrada.

Descripción del código.

En primer lugar se creará una variable que se puede llamar “azar”.

Se usará como evento de inicio el bloque si pantalla hacia arriba. A continuación se asigna a la variable “azar”, un valor al azar entre 0 y 1.

Usando el operador lógico si entonces si no se elige como opción es la ganadora. Se asigna la carita feliz si la variable “azar” toma un valor de cero y en caso contrario ganará la cruz. Se mostrará la opción ganadora usando el bloque mostrar icono.

Tras una pausa de un segundo se borra la pantalla.

Para volver a repetir el proceso se deberá poner la micro:BIT con el panel hacia abajo y al volver a poner el panel hacia arriba se repetirá el código programado

.



Para ver el funcionamiento del programa se recomienda volcar el código a una micro:BIT. En el simulador, el proceso no es tan intuitivo.

Propuesta.

Hacer un programa que permita jugar a “Pares o Nones” contra la micro:BIT

Programando la micro:bit

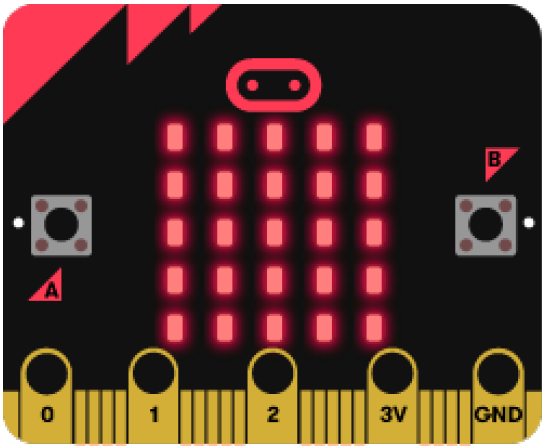
10 Brújula

La brújula detecta el campo magnético terrestre por lo que se puede saber en qué dirección está orientada la micro:bit. (Necesita ser calibrada para asegurar un resultado preciso.)

Calibrar la brújula

La primera vez que grabamos un programa sale este mensaje por la pantalla:

Tilt to Fill Screen *Inclinar para llenar la pantalla*



Luego lo que tenemos que hacer **después** de que aparezca el texto anterior, sale un punto que irá rellenando la pantalla a medida que lo vas inclinando.

Objetivo: que se llene toda la pantalla de leds.

Con un vídeo se ve mejor

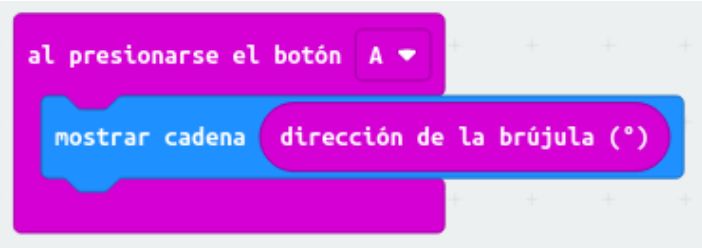
https://www.youtube.com/embed/jvK0ql_u5F0

Pagina oficial de calibración.

Brújula numérica

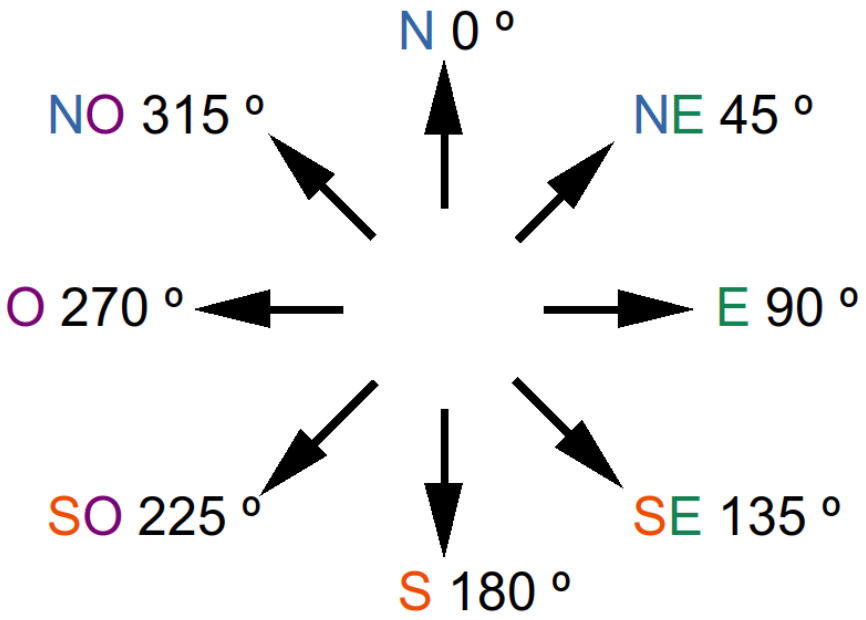
El sensor de campo magnético integrado en micro:bit también puede usarse para medir el **campo magnético terrestre**. De esta forma es posible determinar la dirección del norte magnético y programar así una brújula.

De acuerdo con la esencia de micro:bit bastan unas pocas líneas de código para realizar una brújula digital. El código mostrado a continuación indica en pantalla **la dirección a la que apunta el logotipo** cada vez que se presiona el botón A.



Como el resto de bloques de sensores, el bloque **dirección de la brújula (º)** se encuentra disponible en el menú **Entrada**.

La dirección será mostrada en **grados sexagesimales**. Conviene recordar que el valor de 0º corresponde al norte y el de 180º al sur. La siguiente rosa de los vientos puede servir para clarificar la correspondencia entre los puntos cardinales y los rumbos asociados:



Por ejemplo, y de acuerdo con la rosa de los vientos, una lectura de 340º indicará que el logotipo de micro:bit apunta en dirección casi norte, concretamente nornoroeste.

La **brújula funcionará mejor en exteriores**, ya que el sensor magnético puede ver alteradas sus lecturas por la presencia de masas metálicas en el interior de los edificios, como las de los electrodomésticos o las armaduras de pilares y forjados.

Es posible que al encender micro:bit, aparezca un mensaje en inglés solicitando agitar la tarjeta para **calibrar el sensor**. Esta cuestión se trata en el apartado siguiente.

Reto Conocer la orientación

El norte magnético es la dirección que señala la aguja imantada de una brújula, dirección que no coincide con la del Polo Norte geográfico. Se puede conocer la orientación si se conocen los grados de desviación respecto el Norte Magnético.

Objetivo.

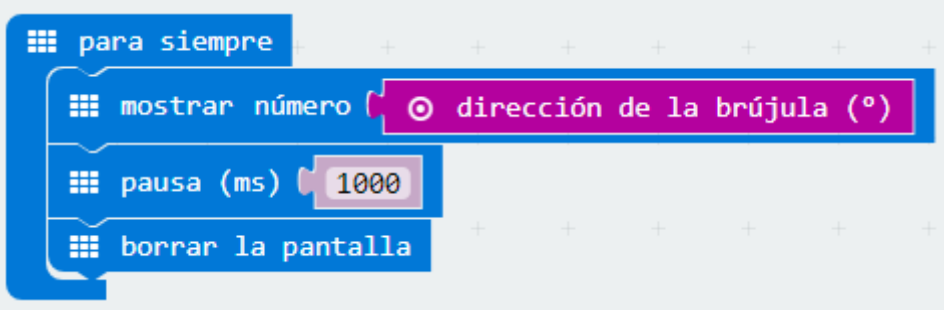
Tras calibrar la brújula, nos dirá la desviación en grados respecto al Norte Magnético de la micro:BIT.

Descripción del código.

Usar el evento para siempre, para iniciar el programa.

Mostrar el valor del sensor dirección de la brújula (º) que ofrece el ángulo de desviación respecto el Norte Magnético.

Esperar 1 segundo y borrar la pantalla.



Brújula analógica

Vamos a completar la funcionalidad de la brújula añadiendo un cursor que marque continuamente la dirección del norte. Usaremos un **evento de tiempo** para leer el sensor y actualizar el dibujo del cursor cada medio segundo.

Dentro del evento de tiempo, el código dibujará una flecha hacia arriba cuando la dirección de micro:bit sea mayor que 315º o menor que 45º, es decir, cuando el logotipo apunte aproximadamente al norte. Los bloques condicionales y los comparadores usados en el código se

encuentran en el menú **Lógica**.

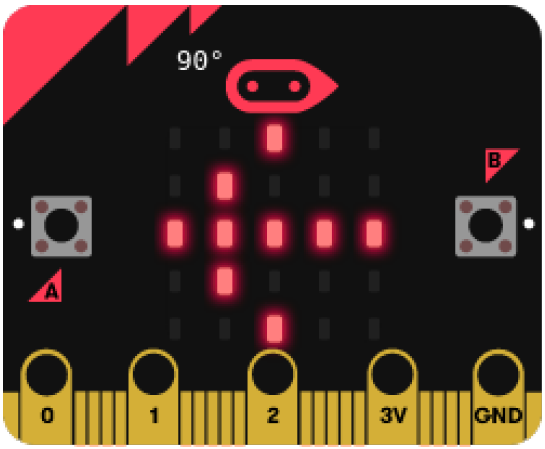


Sin embargo, este programa sólo mostrará la flecha cuando el logo apunte al norte, borrando la pantalla en caso contrario. Para que haya una flecha en pantalla apuntando permanentemente al norte, habrá que pulsar sobre el signo **+** para añadir líneas al bloque condicional.

Copyright 2026



Nótese que cuando el logo de micro:bit apunta al este, dirección de la brújula (º) < 135, hay que dibujar la flecha apuntando a la izquierda (flecha oeste), y que cuando el logo apunta al oeste, hay que mostrar una flecha apuntando a la derecha (flecha este). Sólo de esta forma la flecha apuntará siempre al norte.



Reto Aviso sonoro de orientación Norte.

Cuando se necesita orientarse con un mapa, lo que primero que hay que hacer, es situar el mapa con orientación norte. Para ellos se debe usar una brújula.

Objetivo.

Crear un programa que mediante un aviso acústico se advierta de que la micro:BIT está orientada al Norte.

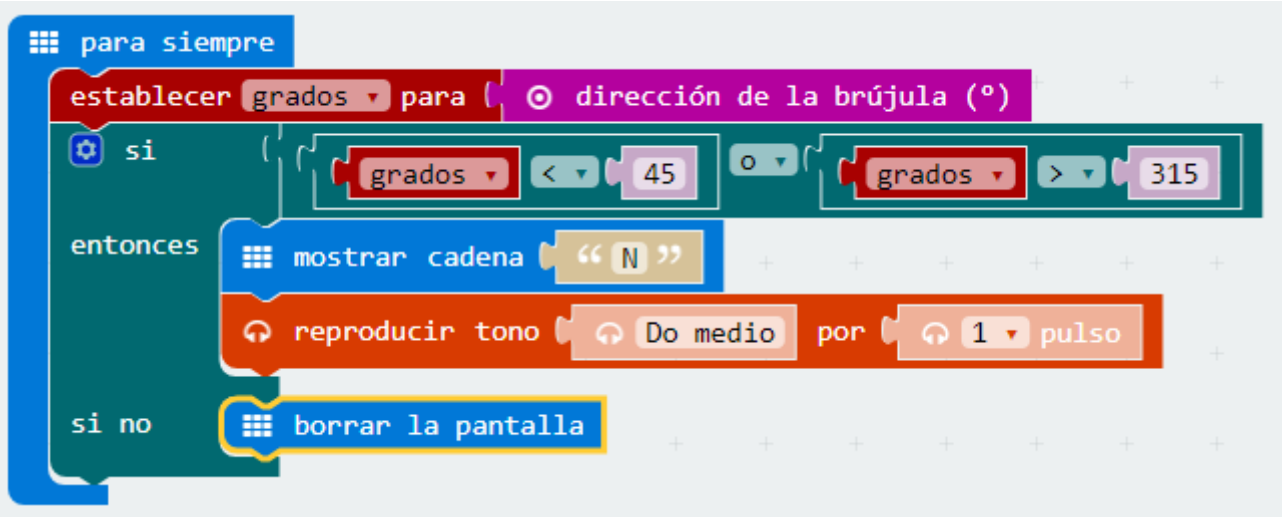
Descripción del código.

Usar el evento para siempre, para iniciar el programa.

Crear una variable donde se almacenará el valor del sensor obtenido por el bloque dirección de la brújula (º).

Se introduce el condicional si entonces si no para comprobar en qué intervalo se encuentra el valor obtenido por el sensor.

Si el valor se encuentra entre 315º y 45º la micro:BIT se encuentra orientada al Norte, por lo que se mostrará una N y se reproducirá una nota. Si no se cumple la hipótesis, se borra la pantalla.



Programando la micro:bit

11 Radio

La radio te permite comunicar tu micro:bit con otras micro:bit. Por ejemplo, puedes conectar todas las tarjetas dentro de un aula a una misma emisora, usarla para enviar mensajes entre ellas y mucho más!.

Radio receiver 06

By Iron Bishop [GFDL, CC-BY-SA-3.0 or CC BY-SA 2.5], from Wikimedia Commons

Micro bits (26186158207)

By Libraries Taskforce (Micro:bits) [CC BY 2.0], via Wikimedia Commons

Reto Sensor inalámbrico.

En algunas situaciones se puede dar el caso de que no se pueda tener conectado el sensor con la unidad receptora de datos mediante cable. Cuando esta situación se presenta una opción es usar sensores inalámbricos.

Usar sensores inalámbricos aporta numerosas ventajas. Entre ellas destacan la seguridad, la centralización de la monitorización y el ahorro de costes por no tener que cablear la instalación.

Un sensor inalámbrico está compuesto por el sistema recolector de datos, un sistema de emisión de datos y una fuente de alimentación. Lo más habitual es que la transmisión de datos sea mediante radiofrecuencia. Por otro lado, se encuentra un sistema receptor de datos conectado al sistema de almacenamiento y procesado.

Objetivo.

Diseñar un sistema inalámbrico de recolección de datos. La micro:BIT emisora se situará en el interior de una nevera y emitirá la temperatura. La unidad exterior mostrará el valor de la temperatura.

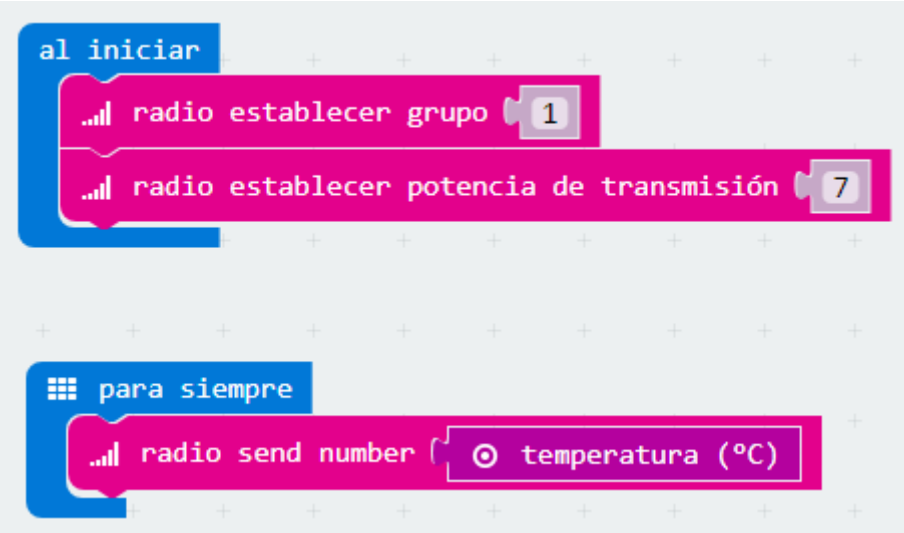
Descripción del código.

Se deberán crear dos códigos, uno para la micro:BIT emisora y otro para la receptora.

Para que las tarjetas se puedan comunicar entre sí, ambas deben estar en el mismo grupo de comunicación. También se debe emitir con la máxima potencia para evitar la atenuación que sufre la tarjeta emisora al encontrarse dentro de la nevera. Por este motivo, se deberá usar el bloque radio establecer potencia de transmisión situado en la categoría Radio y usar el máximo valor que es 7. Se recomienda usar el evento al iniciar para establecer el grupo de comunicación y asignar la potencia de transmisión.

micro:BIT emisora:

Tras establecer el grupo de emisión en 1 y poner la potencia de emisión al máximo, se enviará el valor de la temperatura mediante el bloque radio send number.



micro:BIT receptora:

La cadena enviada se guarda en la micro:BIT receptora en una variable que por defecto se llama “receivedNumber”. Se usará el bloque mostrar número, para visualizar la temperatura medida en la nevera. Tras un segundo se apagará el panel LED. No olvidar establecer al inicio el grupo de comunicación.

El código de la tarjeta receptora quedaría de la siguiente forma:



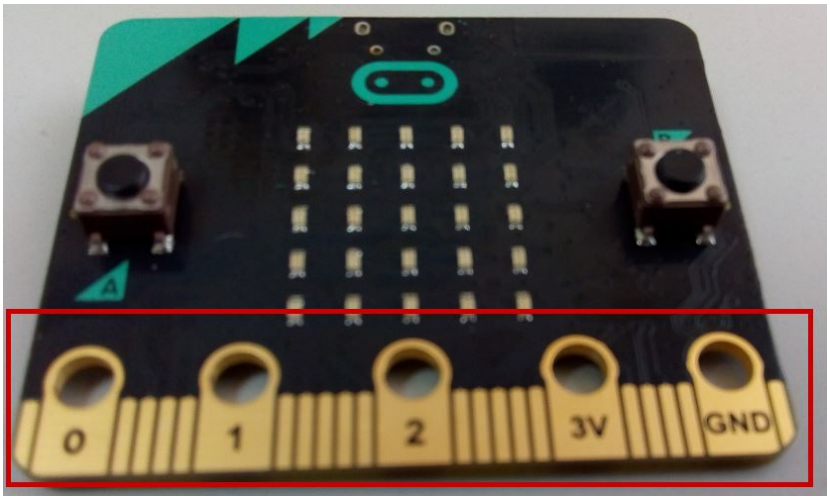
Propuesta.

Se propone mostrar una animación en el panel de la micro:BIT emisora, que confirme que el programa está en marcha.

Programando la micro:bit

12 Pines de entrada y salida

La micro:BIT dispone de 25 conectores situados en el borde inferior . A través de ellos se podrán programar motores, LEDs o cualquier otro componente o sensor externo.



~ Biografía del autor ~

Reto ¿Hay continuidad?

En algunas ocasiones interesa conocer si un cable tiene continuidad en toda su longitud. Para comprobar la continuidad se suele usar un multímetro, midiendo la resistencia del conductor. Una lectura de resistencia infinita indicaría una interrupción en alguna parte del interior del cable.

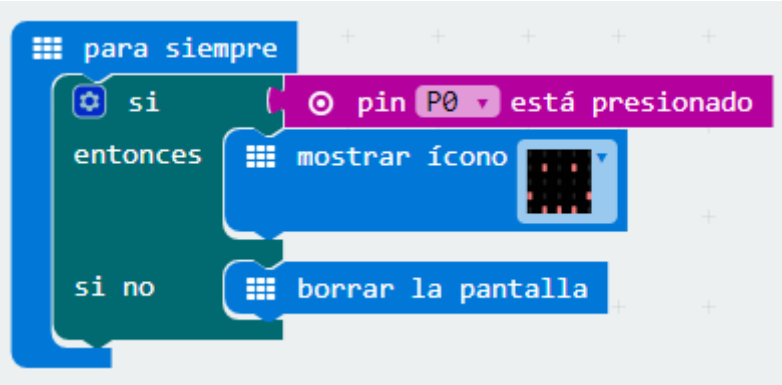
Objetivo.

Diseñar un programa que permita comprobar si un cable tiene continuidad. Si no está cortado, mostrar una cara feliz.

Descripción del código.

Se usará el bloque pin (P0) está presionado de la categoría Entrada para comprobar la continuidad. Se propone emplear el evento para siempre y usar un condicional que verifique si el P0 está presionado. Si se cumple la condición se muestra una carita feliz, en caso contrario, el panel

permanece apagado. El código podría quedar de la siguiente forma:



Para comprobar el funcionamiento se puede usar un cable con bananas conectados al GND, si se conecta con el P0 y el cable tiene continuidad, deberá aparecer una carita feliz en el panel LED.

Propuesta.

Modificar el código para que cuando el cable a comprobar tenga continuidad haga sonar la melodía “jump up” y si no tiene continuidad que se emita la melodía “jump down”.

Reto Música

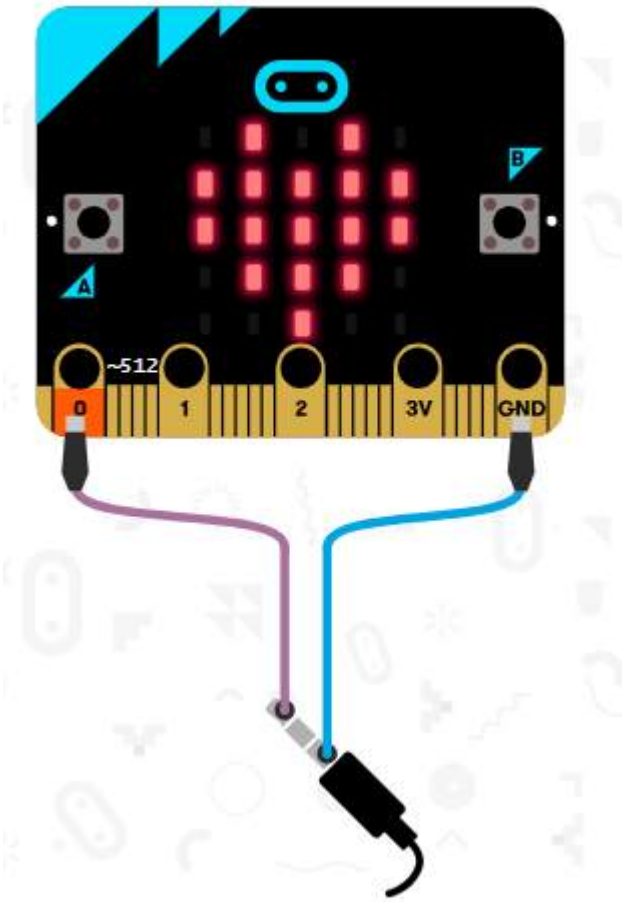
Se puede añadir música utilizando unos cascos o altavoces de PC.. conectandolo al puerto P0 (pon sonido a tu ordenador y lo oirás):

https://makecode.microbit.org/---run?id=_PmV0UdLJrWRs

Conexión con el "minijack"

Para la conexión se puede utilizar unas pinzas de cocodrilo (ver conexión)

Copyright 2026



https://www.youtube.com/embed/Di6b-SMuC-g

Si no se tienen pinzas de cocodrilo, haciendo un poco DYC (lease *mañoso*) con papel de aluminio y celo (ver cómo se hace)

Descripción del código

El código es muy fácil:



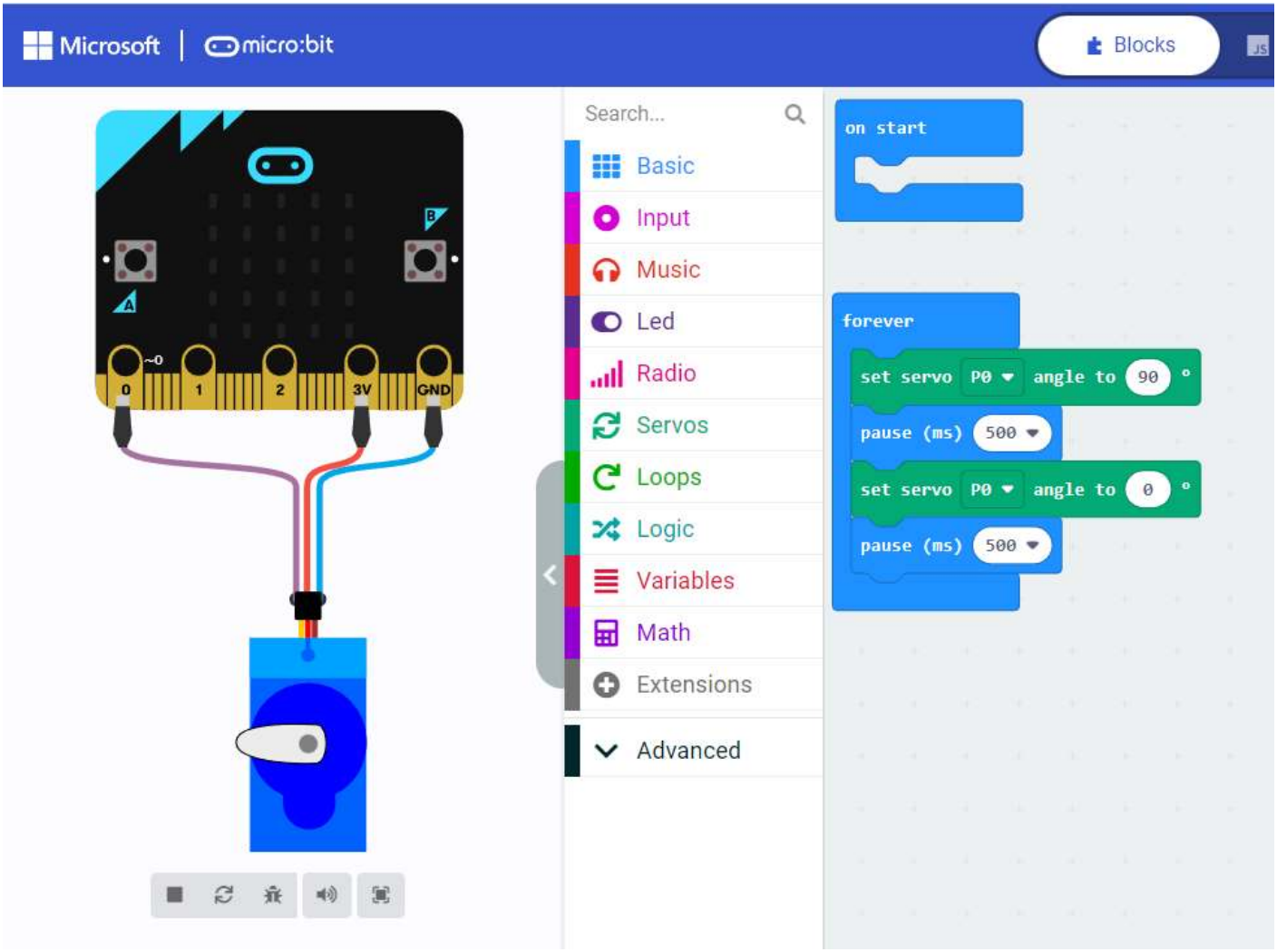
Hay que tener cuidado con lo de *repitiendo una vez* prueba no ponerlo y verás que se van mezclando melodías cada vez que ser repite el bucle y al final no se sabe qué es lo que está sonando

https://makecode.microbit.org/#pub:_PsreyC1tVbfg

No intentes relés directamente

A pesar de que microbit parece que se puede poner directamente, no tiene suficiente energía para mover un motor, aunque sea un servo sencillo:

Copyright 2026



La única manera es:

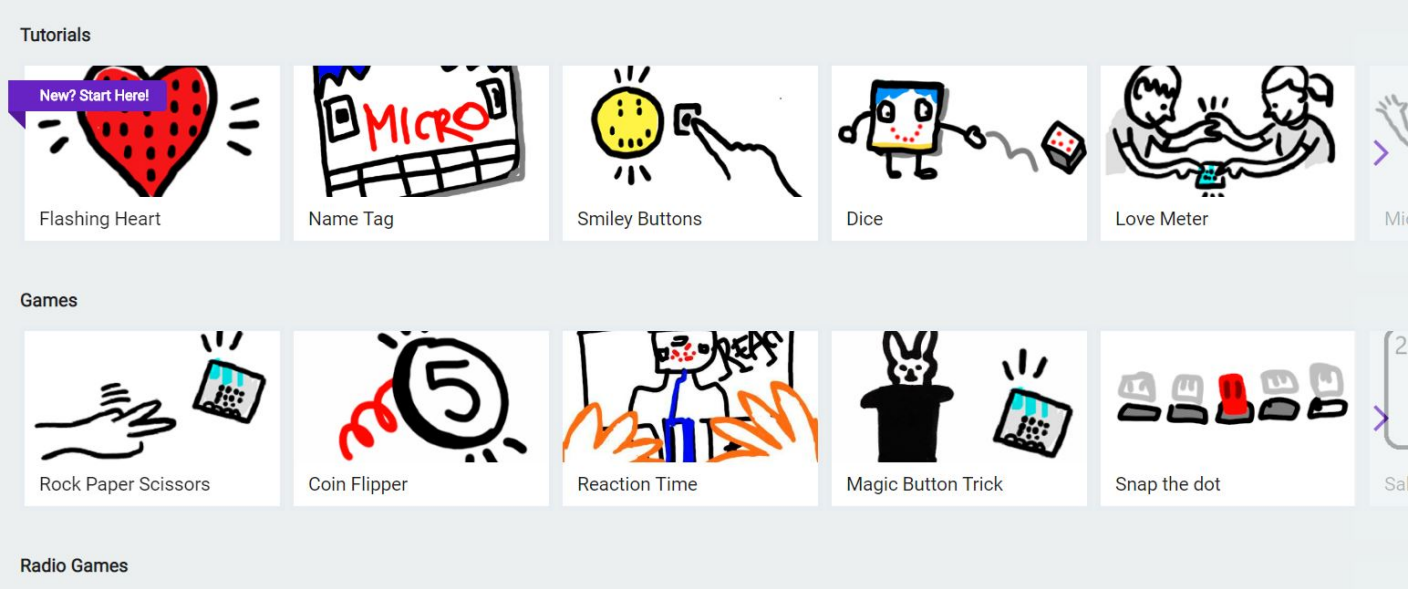
- O utilizar una placa de expansión, hay muchas en el marcado, un ejemplo es Microbit y Smartcar
- O sin placa, hacerlo manualmente, dando energía con unas pilas, lo tienes explicado aquí y este es el resultado https://youtube.com/shorts/Hx3_5GM-umc

https://www.youtube.com/embed/Hx3_5GM-umc

También venden tarjetas para hacer intermediarias entre estos elementos y la microbit, el Smartcar es un ejemplo.

Reto libre

Ahora te retamos a lo siguiente: Entra en Makecode https://makecode.microbit.org y **ELIGE UN PROYECTO** EL QUE QUIERAS DE LOS QUE APARECEN **Y HAZLO**



Tendrás que hacerlo si quieres contestar a la pregunta test del curso correctamente !!!

Copyright 2026

SmartCar. Dentro del curso Microbit+SmartCar

Copyright 2026

SmartCar. Dentro del curso Microbit+SmartCar

SmartCar



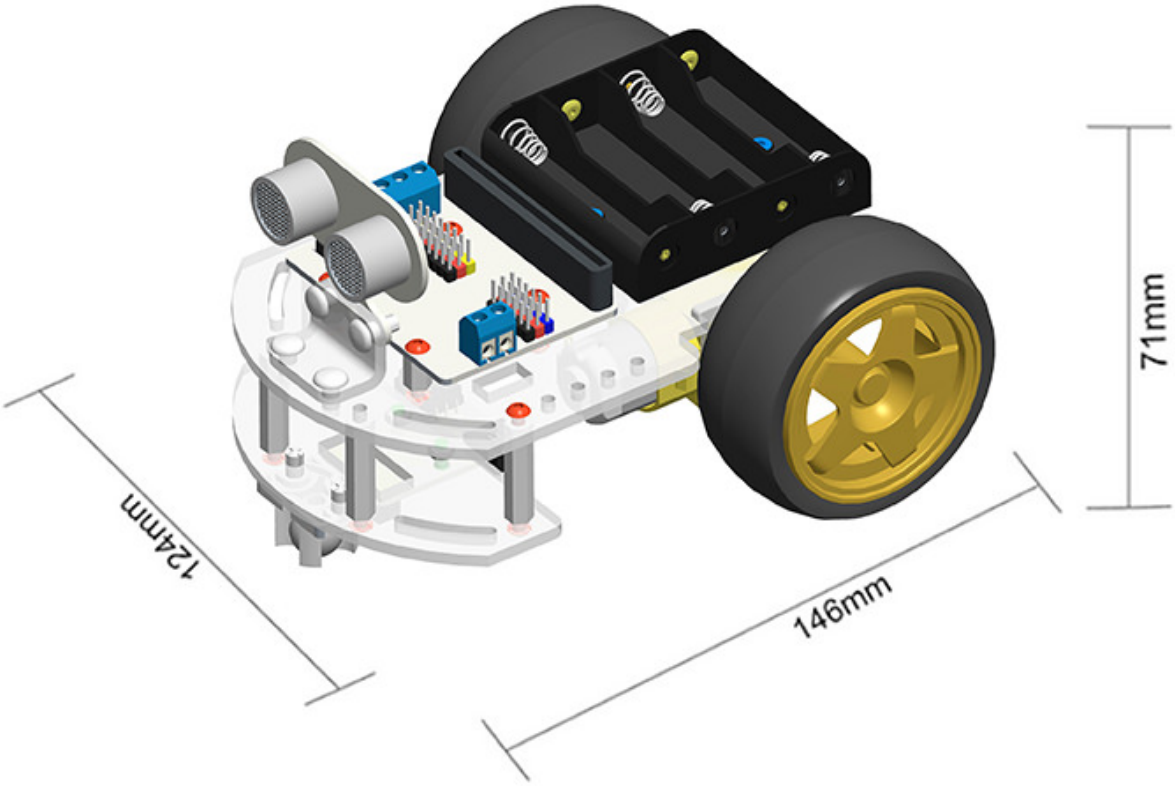
Micro:BIT se queda un poco triste si no se "mueve", pero hay que tener en cuenta que en la robótica el movimiento se paga:

- Adición de motores, chasis, ruedas...
- Necesitas una electrónica añadida de potencia y control de los motores.
- Necesitas unos sensores de lo contrario el movimiento se queda insípido sin reacción al mundo exterior.

La placa de micro:BIT no incorpora movimiento, por eso cuesta tan poco, pero si quieres este extra tienes que pagar (casi el doble de lo que cuesta micro:BIT). Aún así el conjunto sale más barato que mBot.

Este kit viene con sensor de ultrasonidos, sensor de sigue-lineas y la placa **motorbit** que se inserta la micro:BIT además de tener altavóz y salidas a puertos por si se quiere añadir servos por

ejemplo. Para más información de esta placa consultar esta web.

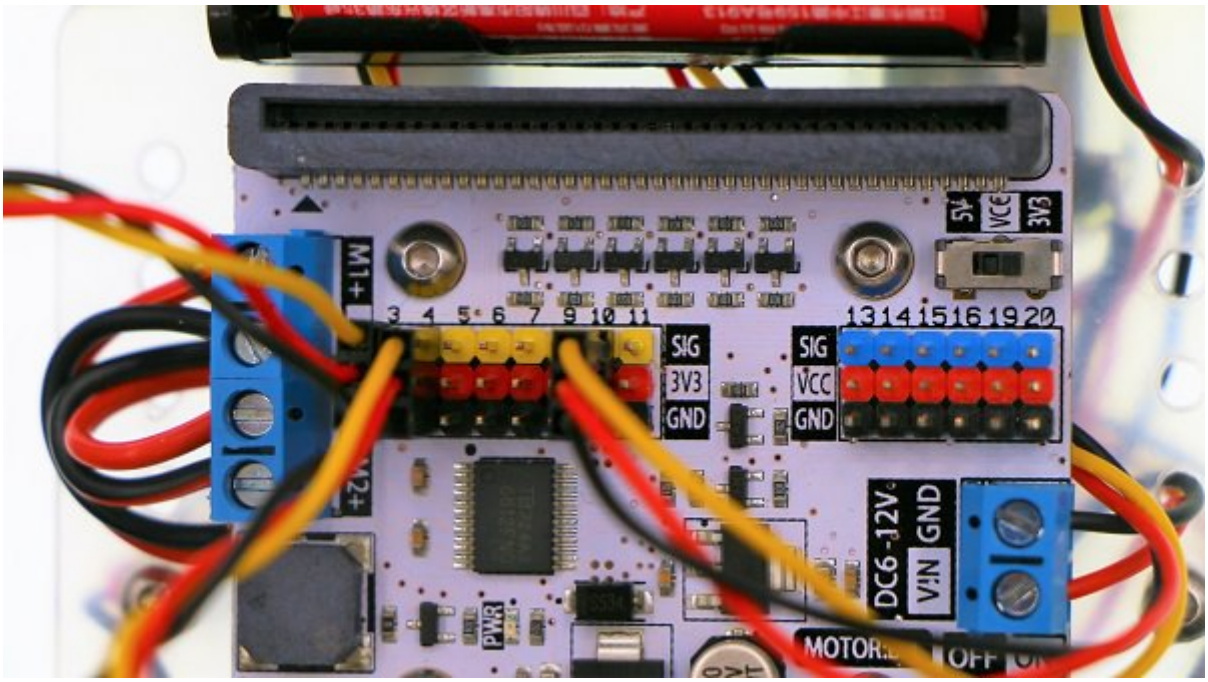


Montaje

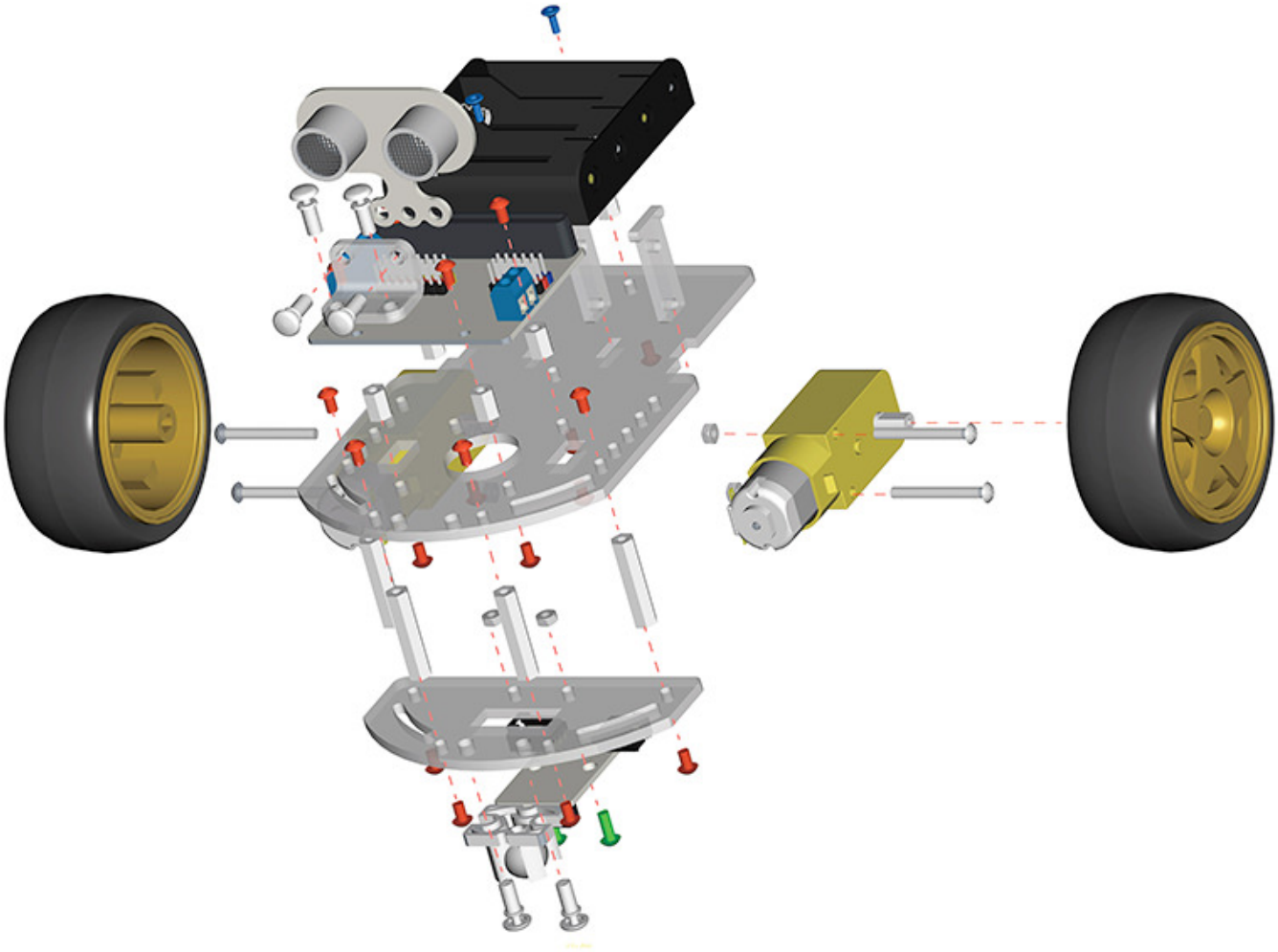
El paquete viene con unas piezas listas para montar que no requieren especial destreza DYC, lo único "*difícil*" es que hay que **soldar** los cables a los motores (4 pequeñas soldaduras).

El sensor sigue-lineas y el sensor ultrasonidos se puede poner en cualquier pin, pero en este curso lo fijaremos en:

- Sensor ultrasonidos en el PIN 10
- Sensor siguelíneas izquierda en el PIN3
- Sensor siguelíneas derecha en el PIN4



Copyright 2026

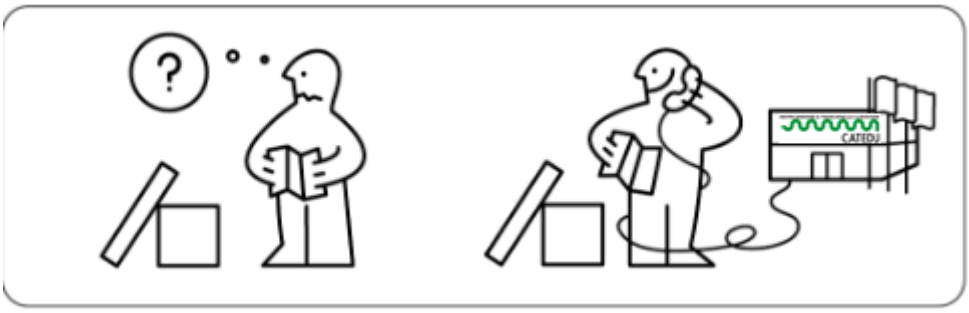


Con un poco de paciencia se hace muy bien

https://www.youtube.com/embed/_EXX9h0i1WU

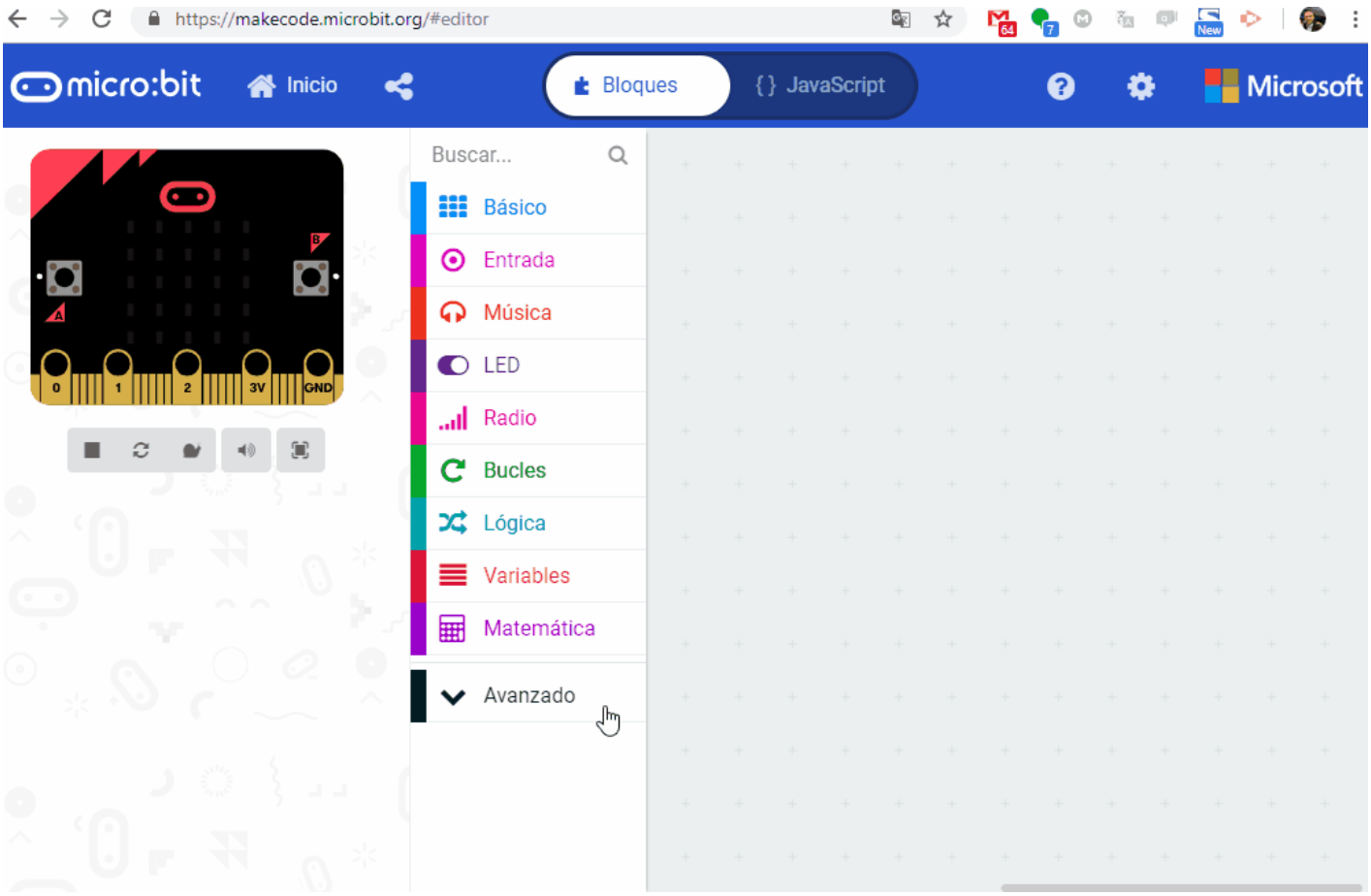
Nosotros no somos comerciales, luego ésto mejor que no :

Copyright 2026



Conexión con Makecode

Para tener las instrucciones específicas, hay que entrar en Makecode https://makecode.microbit.org/#editor entrar en **AVANZADO** en **EXTENSIONES** y buscar la extensión **MOTORBIT** instalar, y ya está tenemos instalado las funciones específicas de este coche:



SmartCar. Dentro del curso Microbit+SmartCar

Reto 1 Musica

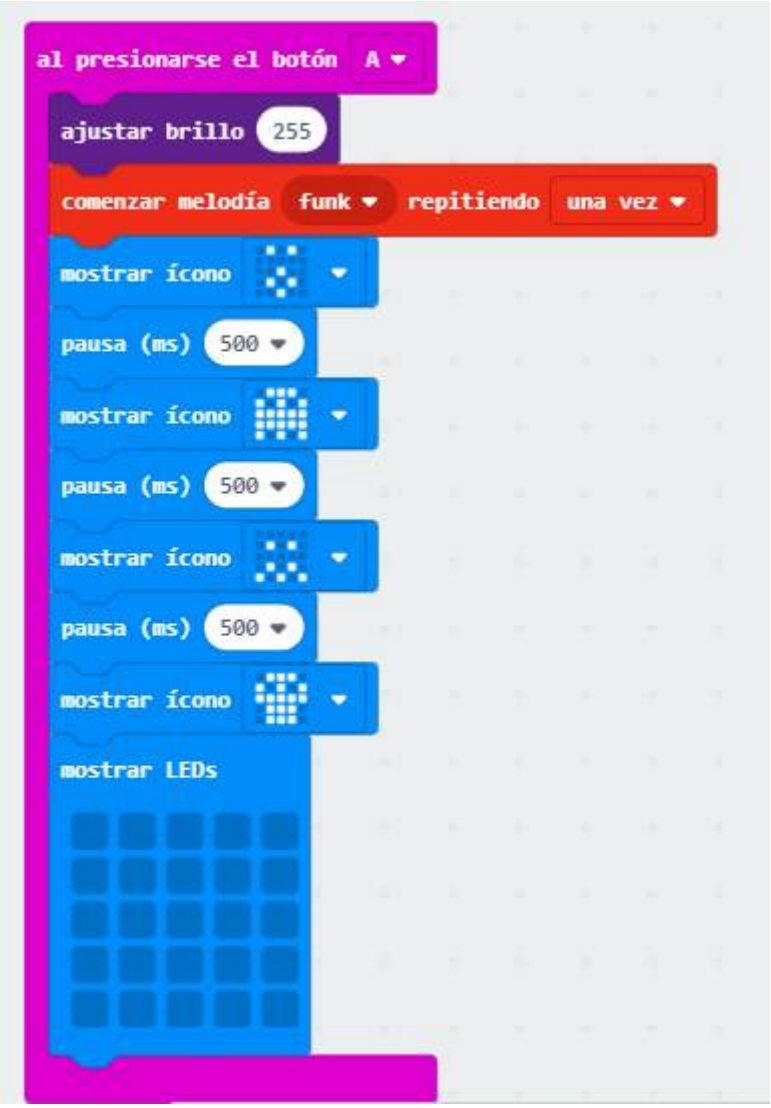
SmartCar tiene incorporado un altavoz ya en el puerto P0 por lo tanto no es necesario hacer cocodrilos o papeles de aluminio como hemos visto anteriormente.

Vamos a hacer un programa que al pulsar el botón A suene música etc...

https://www.youtube.com/embed/IEuQvhxcw7w

Descripción del programa

Es simplemente que si presiono el botón A pues que suene y salgan iconos por la pantalla:



Puedes probarlo en este simulador

https://makecode.microbit.org/---run?id=_hMmRRea9s76w

Y aquí lo tienes para descargar:

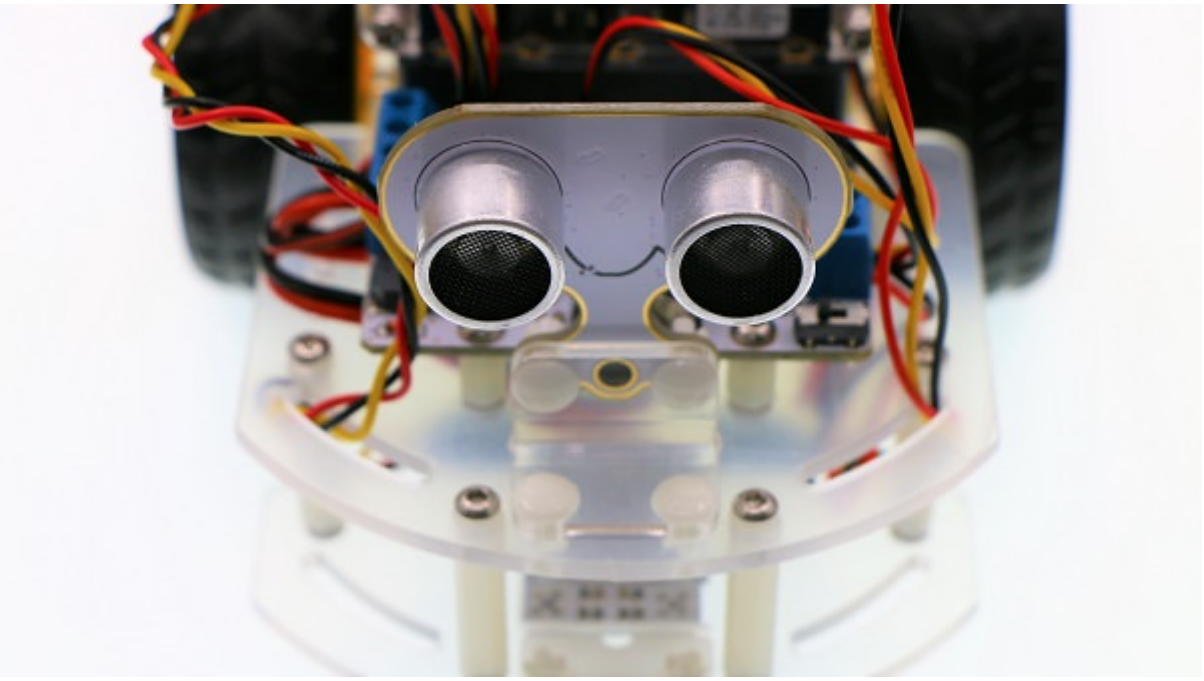
Copyright 2026

https://makecode.microbit.org/#pub:_hMmRRea9s76w

Copyright 2026

SmartCar. Dentro del curso Microbit+SmartCar

Sensor distancia



Al poner la extensión micromotor se añade la instrucción de sonar que está en otro apartado:



Este sensor, en nuestro kit lo conectaremos en **PIN 10** aunque puede estar conectado en cualquier otro. Además recomendamos crear una variable (por ejemplo "distancia") y poner las unidades de medida en cm que son más intuitivas. Esta instrucción se pone al principio del bucle y sólo hay que utilizar la variable **distancia**:



El sensor de distancia funciona igual que los sónares: Uno "ojo" es realmente un altavóz que emite un sonido con frecuencia alta (*ultrasonido, por eso no lo oímos*) y el otro "ojo" es un micrófono que recibe ese pulso. El circuito electrónico calcula la distancia con la diferencia de tiempo en emisión y la recepción del eco, igual que los rádares, sónares...

https://giphy.com/embed/4xGCaTMCO59le

via GIPHY

O incluso los murciélagos, delfines...

https://giphy.com/embed/3o7TKu5aIDY4tU3SXm

via GIPHY

Reto 2 Me quedo a 5cm

El siguiente reto es que el microCar se quede a 5cm

ATENCIÓN vamos a poner el sensor de ultasonidos en el PIN 10

https://www.youtube.com/embed/iCLnAHLLolc

Solución

La solución se encuentra en esta página https://www.elecfreaks.com/learn-en/motor_bit_smart_car_case_02/

El proyecto te lo puedes descargar aquí

https://makecode.microbit.org/#pub:_J0HabpedtUvY

Reto

Que empiece parado y que se aleje si ponemos la mano, o sea, intenta cogerme

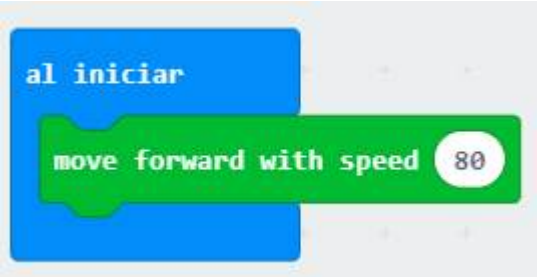
Reto 3 Evitar obstáculos

¿Cómo no? lo pide a gritos ! vamos a hacer un roomba

https://www.youtube.com/embed/nVwOLRhooOc

Descripción del proyecto

Empezamos el programa que al iniciar vaya recto:



Luego hacemos el bucle "para siempre": * El sensor no funciona muy fino, hay veces que da *0 falsos* por lo tanto el bucle **si** ponemos una condición **si es mayor de 5 y es menor de 10** para quitarnos estos falsos positivos de obstáculos. * si encuentra obstáculo, que pare un poco, que *recule* y que gire * cada instrucción anterior con una pequeña pausa, cuanto más grande sea la pausa más *reculará*, girará etc.. * si no encuentra obstáculo que siga recto



Aquí lo tienes en editor

https://makecode.microbit.org/#pub:_5RPYL4TfKXae

Reto

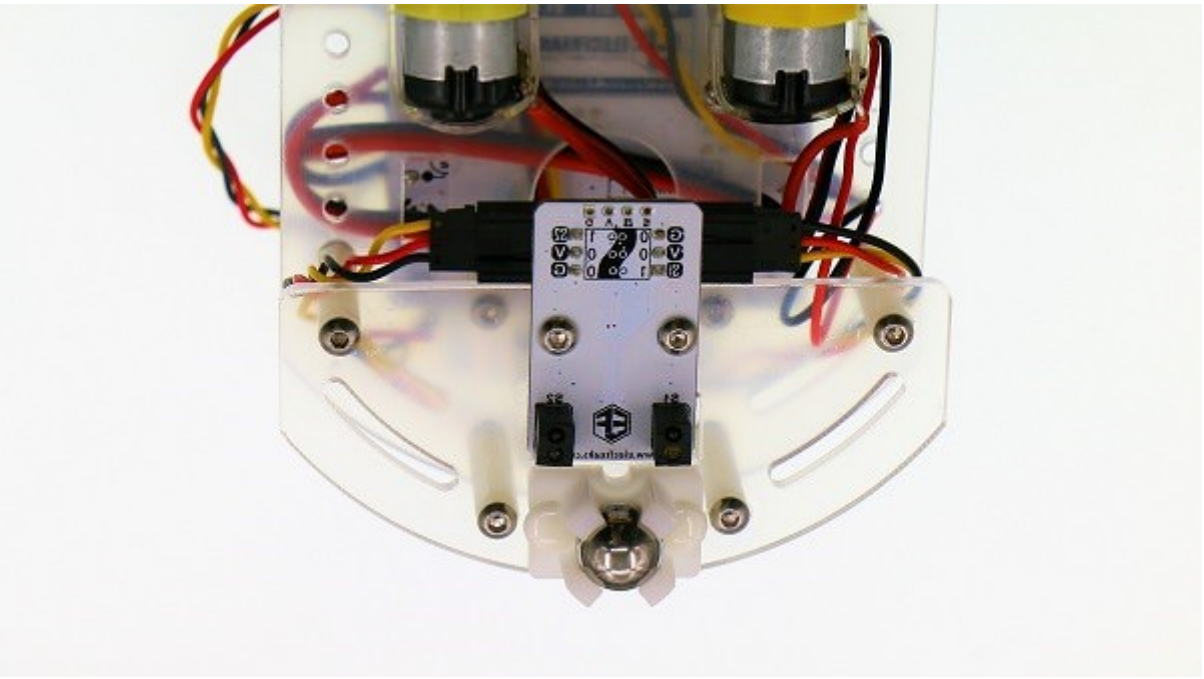
Si te fijas sólo gira a la derecha. Modifica el anterior programa para que gira a la derecha o a la izquierda de forma aleatoria.

Copyright 2026

SmartCar. Dentro del curso Microbit+SmartCar

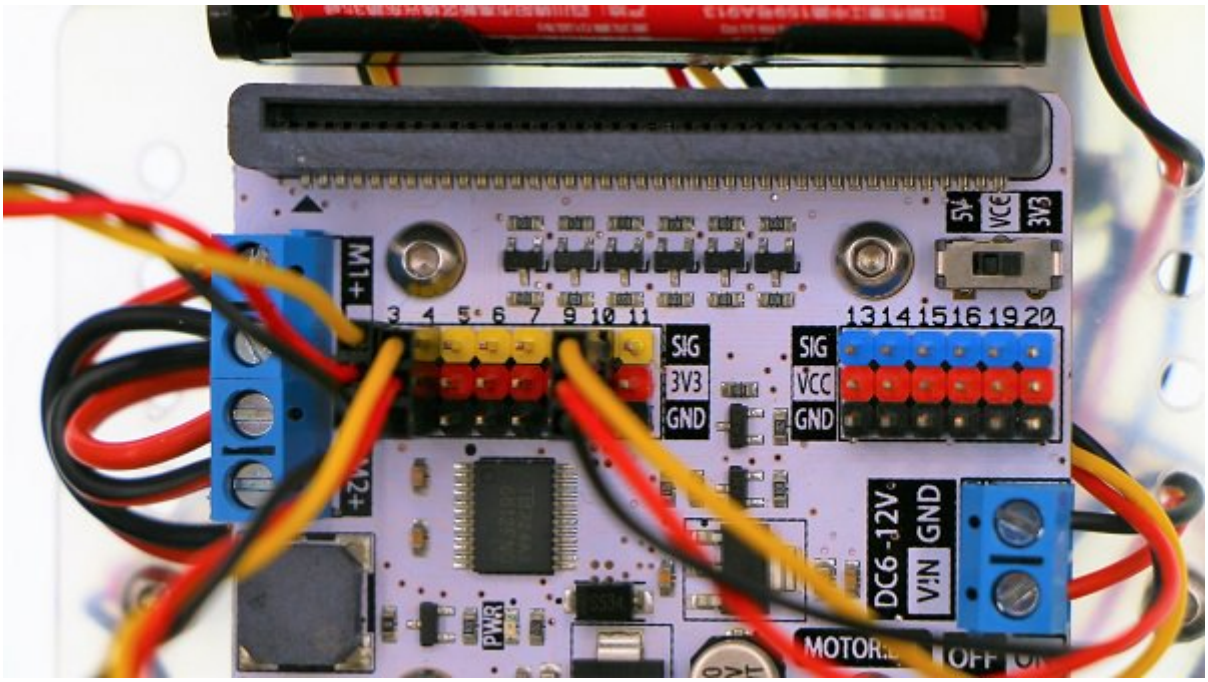
Sigue-lineas

El sigue-lineas son dos sensores que están colocados debajo del robot

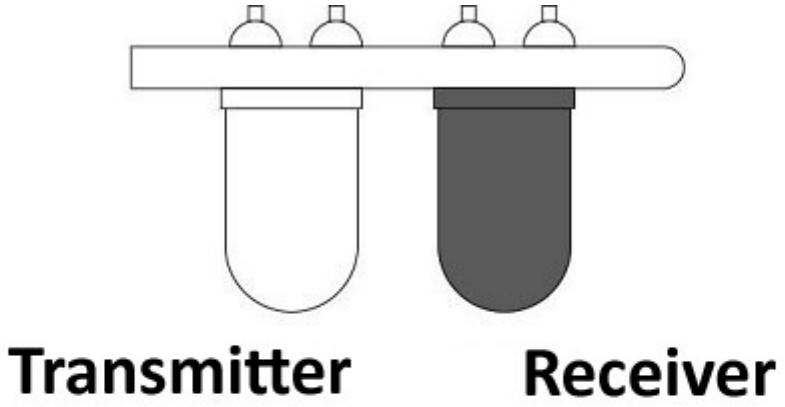


Por unificar criterios, los conectaremos en P3 y P4

Copyright 2026



Cada sensor tiene dos leds, uno emisor y otro receptor. El receptor recoge la luz reflejada, si hay debajo algo que no refleja la luz (por ejemlo una línea negra) entonces manda OFF en caso contrario ON





OJO VA AL REVÉS es decir * cuando hay linea negra es OFF * cuando no hay linea es ON

por lo tanto queremos:

- Que cuando sea OFF sea un 1 lógico (línea)
- Que cuando sea ON sea un 0 lógico (no hay línea)

Esto se llama **CONFIGURACIÓN PULL-UP** (pincha aquí para saber más) luego lo primero que tenemos que hacer es configurar estos sensores como PULL-UP con estas instrucciones (han traducido **UP** como *subir*):



Están un poco escondidas:



Y luego crear unas variables por ejemplo **izquierda** y **derecha** dentro del bucle que lean esos sensores. El resto del código sólo utilizaremos estas variables:



Reto 4 Seguir la línea

Otro reto que pide a gritos este sensor:

https://www.youtube.com/embed/cY6nqgHxXio

> Consejo: en clase, antes de atacar con este reto, aconsejamos otro más sencillo como que el robot se mueva y si encuentra línea que se pare, este reto lo puedes ver aquí

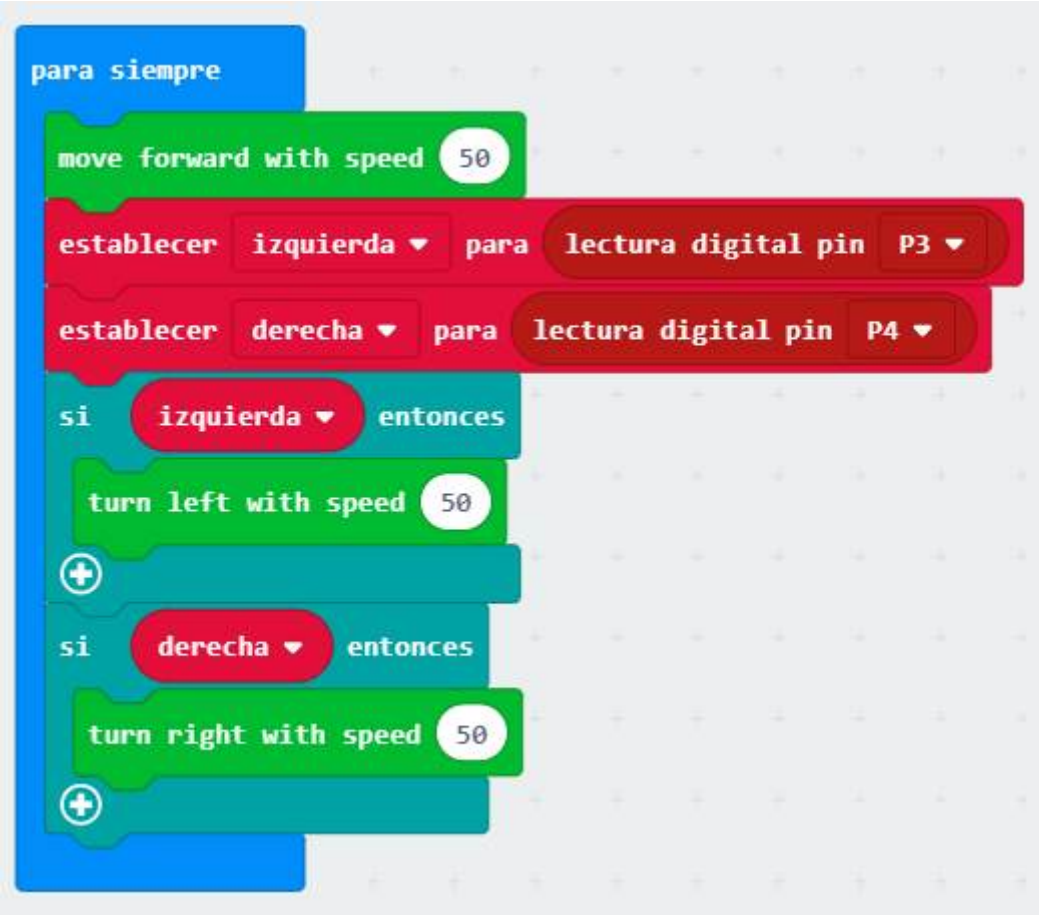
Descripción del programa

Al empezar configuraremos los sensores en PULL-UP tal y como hemos explicado:



Luego entramos en el bucle donde simplemente dice:

- Sigue hacia delante
- Si te desvías hacia la derecha gira a la izquierda
- Si te desvías hacia la izquierda gira a la derecha



El programa tiene la pega que si aumentamos la velocidad, pierde la línea.

Evidentemente hay muchas versiones y mejoras, por ejemplo aquí pero ésta https://makecode.microbit.org/_U3VP8JhVTXaJ es desde luego la versión más sencilla.

https://makecode.microbit.org/#pub:_U3VP8JhVTXaJ

Copyright 2026

SmartCar. Dentro del curso Microbit+SmartCar

Reto 5 Mando a distancia

Esta vez **vamos a utilizar dos micro:BITs**

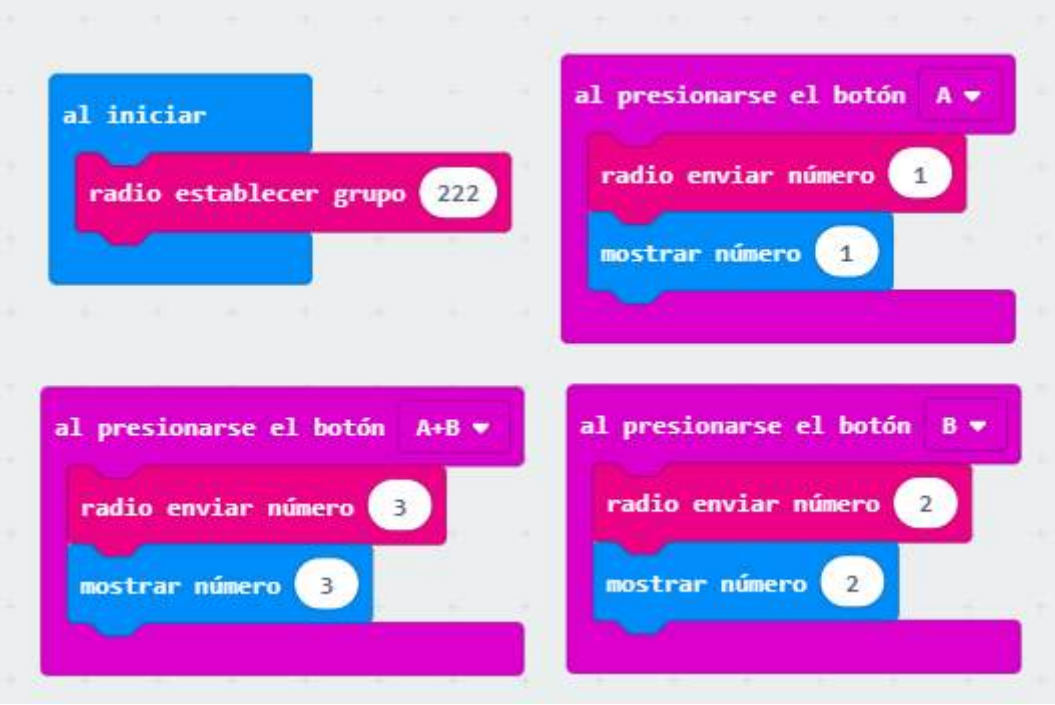
https://www.youtube.com/embed/oxumk9GYoVU

Descripción del programa

Microbit que hace de mando

Este microBIT hay que alimentarlo con pilas o utilizando una batería típica de móvil.

El mando se inicia en un grupo (en este caso el 222) y simplemente realiza lo siguiente: * Si se pulsa A manda un 1 y lo visualizo * Si se pulsa B manda un 2 y lo visualizo * Si se pulsa A+B manda un 3 y lo visualizo



El programa te lo puedes descargar aquí:

https://makecode.microbit.org/#pub:_cWz6s9aeTPXr

Microbit que está en SmartCar

Al iniciar el programa asigna este microbit al mismo grupo de radio que el mando y además asigna una nueva variable con valor 0



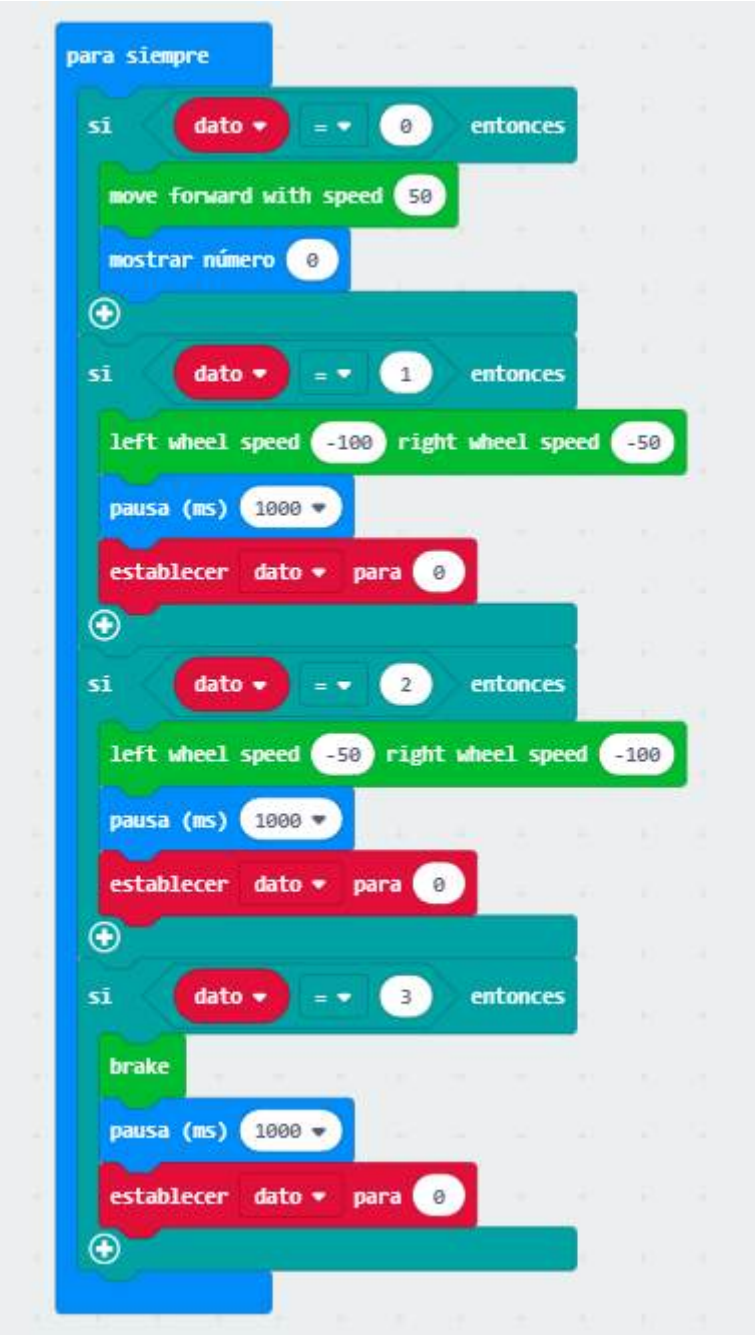
Al recibir un número lo asigna a esa variable y además lo muestra:



Y establecemos un bucle por siempre que :

- Si no ha recibido nada, es por lo tanto dato=0 luego que siga hacia delante
- Si recibe 1 que gire hacia la derecha y hacia atrás
- Si recibe 2 igualmente pero al otro lado
- Si recibe 3 que pare
- En los tres casos anteriores damos un tiempo para que ejecute la instrucción con una pausa y luego reseteamos dato para que siga el robot su camino

Copyright 2026



El proyecto te lo puedes descargar aquí :

https://makecode.microbit.org/#pub:_ftuFv8AReFYq

Copyright 2026

Basic kit. Fuera del curso \(de momento...\)

Copyright 2026

Basic kit. Fuera del curso \(de momento...\)

Basic kit

¿Qué es?

Su principal placa se atornilla en la microBit y desde esa placa se van añadiendo los diferentes elementos que proporciona la microbit

https://www.youtube.com/embed/ErNDclT4a7A

Copyright 2026

Basic kit. Fuera del curso \(de momento...\)

Reto 1 Movimiento de un servomotor

Reto 17 Moviendo un servomotor

Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

Objetivo.

Diseñar un programa que al pulsar el botón A el servo se posicione a 180º, si se acciona el pulsador B se debe situar a 0º y al pulsar conjuntamente A+B se ubicará a 90º.

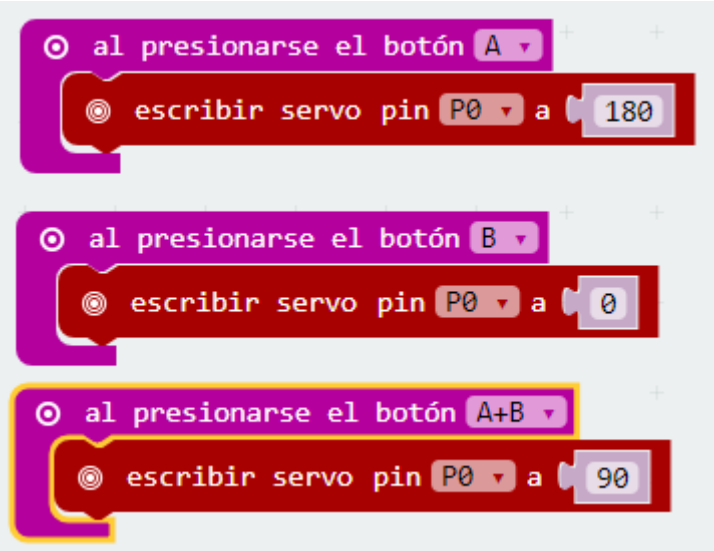
Descripción del código.

El bloque escribir servo pin (P0) a (180) que sirve para posicionar el servo, se encuentra ubicado en la categoría Pines, que aparece al pulsar en la sección Avanzado del menú principal.

Se puede elegir en qué pin se conecta el servo y con qué ángulo se posiciona el brazo del servo. Se usará el bloque al presionarse el botón, para iniciar los diferentes programas según el pulsador accionado.

El programa quedará de la siguiente forma:

Copyright 2026



En el simulador aparece la imagen de un servo y hará lo programado según el pulsador accionado.

Propuesta.

Crear un código que permita mover el servo en pasos de 15º al pulsar el botón A y devuelva el brazo a 0º al accionar el pulsador B.

Para finalizar...

Copyright 2026

Para finalizar...

Muro micro:BIT

MURO proyectos MICROBIT

Colabora en este muro https://padlet.com/CATEDU/microbit añadiendo proyectos que veas interesantes:

https://padlet.com/embed/euk7b9yvu3pe

Hecho con Padlet

Copyright 2026

Para finalizar...

Grupo Robotica Educativa

Además esta placa tiene su propio grupo en Facebook

Copyright 2026

Para finalizar...

Pensamiento computacional

¿Dónde se encaja este robot?¿se puede comparar este robot con otros robots de otros cursos que hacemos desde CATEDU?

Esta es la hoja de ruta, no se tiene que tomar al pie de la letra, pero intenta ayudar al profesorado que tenga una visión global de tanta oferta:

Como se puede ver MICRO:BIT es una alternativa muy buena y barata para que los centros puedan empezar con la robótica y a nuestro criterio es el robót con mejor relación coste con prestaciones.

Guía orientativa

https://docs.google.com/presentation/d/e/2PACX-1vQHiZvv1cGHet7eXVy-QcECY4Lj0k0l7ntDi8MevRWHQX-9myA0bfR5IofMeuGZkWD0Hw-Ob-MGoco_/embed?start=trueloop=true&delayms=3000

> Tenemos un **grupo Telegram Robótica Educativa en Aragón**, https://t.me/roboticaeducativaaragon



Copyright 2026

Para finalizar...

Enlaces de interés

Comunidad de España: Microes.org

En Microes.org http://microes.org/ creamos guías didácticas como la que tienes en tus manos para ayudar a los profesores y usuarios a sacar el máximo partido a micro:bit.

¿Dudas? ¿Quieres colaborar con la comunidad? Contáctanos: e-mail: info@microes.org teléfono: 945 29 80 50 (de 8:00 a 14:00)

Comunidad mundial Microbit.org

Página de microbit oficial: https://microbit.org/es/

Comprar la Microbit

En España en : http://robotopia.es/ o complubot o en el extranjero aquí https://microbit.org/es/resellers/

Bluetooth

Quieres manejar micro:BIT con Bluetooth, mira Nosotros aconsejamos las aplicaciones bitty http://www.bittysoftware.com/apps.html

Twitter #microbit

Copyright 2026

Tweet #microbit

Copyright 2026

Para finalizar...

Sobre el autor

José Francisco Muñoz Fernández (Almería, 1968). Arquitecto superior, desde que cayó en sus manos un ordenador quedó fascinado por esta ciencia digital. La tecnología catalizó su innato espíritu investigador e innovador y le llevó a integrarla en todas las facetas de su vida formativa y laboral.

Tras años ejerciendo como arquitecto, decide dar más protagonismo a sus otras dos pasiones, la tecnología y la educación, comenzando a ejercer como profesor de robótica.

Esta experiencia docente le sirve para comprobar el gran potencial que tiene la robótica en las aulas y le empuja a querer ayudar a otros docentes en la puesta en práctica rápida y eficaz de la Robótica Educativa en sus centros.



José Francisco Muñoz (Almeribot.com)

Copyright 2026

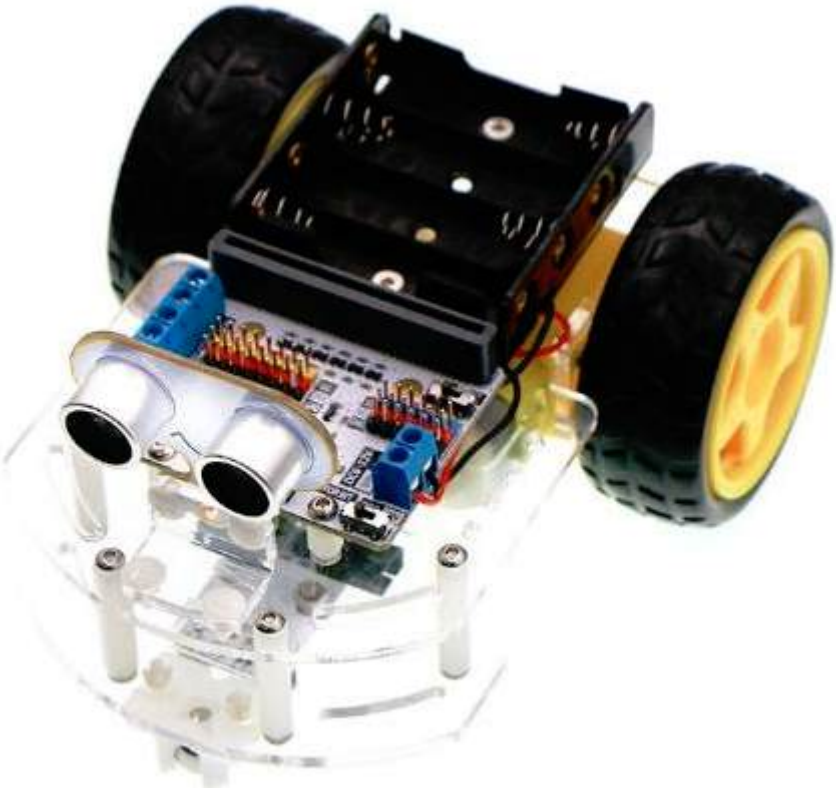
Para finalizar...

¿El curso continúa?

Micro:BIT tiene muchas extensiones que permiten extender este curso, y por lo tanto son cursos totalmente nuevos:

Curso Micro:BIT y MicroCar

Contenidos del curso : https://catedu.gitbooks.io/smartcar-micro-bit/content/



https://giphy.com/embed/Meg4PTH32tPpe

via GIPHY

Para finalizar...

Créditos

Autoría

- {{ book.authors[0] }}

Colaboradores:

{% for collaborator in book.collaborators %} * {{collaborator.name}} en {{collaborator.edited}} {% endfor %}

Cualquier observación o detección de error en soporte.catedu.es

Los contenidos se distribuyen bajo licencia **Creative Commons** tipo **BY-NC-SA** excepto en los párrafos que se indique lo contrario.