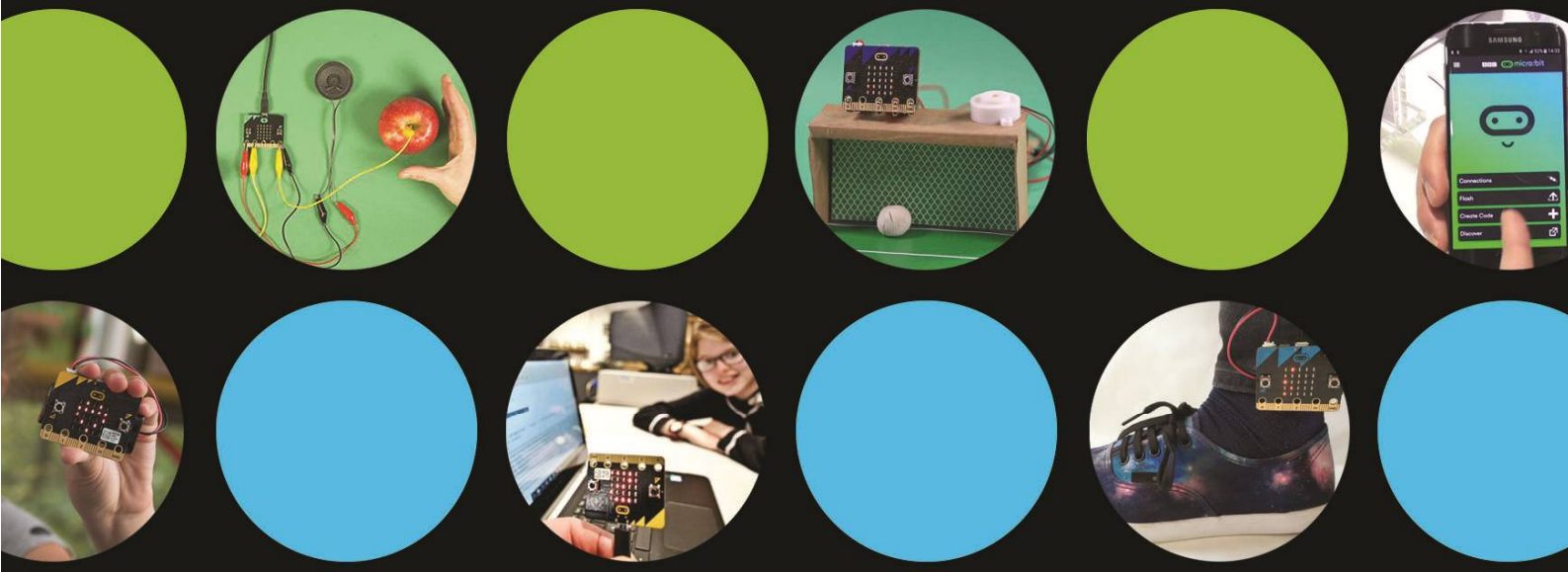
Guía de manejo • Resumen online gratuito

micro:es

Manual de Programación micro:bit



MakeCode

SCRATCH

open source hardware

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

Manual de Programación micro:bit

Título: Manual de Programación micro:bit
Autor: José Francisco Muñoz (https://almeribot.com/)
Editor: Microes.org

1ª Edición: Julio, 2018

Este manual es un resumen online de la Guía de manejo impresa (en breve disponible).
Puedes descargarlo de forma gratuita en:
http://microes.org/formacion.php

micro:es

Comunidad micro: bit en España

¿Dudas? ¿Quieres colaborar con la comunidad?. Contáctanos:

e-mail: info@microes.org
teléfono: 945 29 80 50 (de 8:00 a 14:00)

Índice de contenidos

~ Introducción a la micro:BIT ~

1.- ¿Que es la micro:BIT?

La micro:BIT es una pequeña tarjeta programable, con un costo asequible a cualquier bolsillo. 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.

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 este enlace: http://kittenbot.cc/bbs/topic/3/microbit-makecode-offline-version

Además la micro:Bit se puede programar con JavaScript, Pyton, Scratch (añadiendo una extensión) y Tickle (aplicación para iPad).

3.- Características.

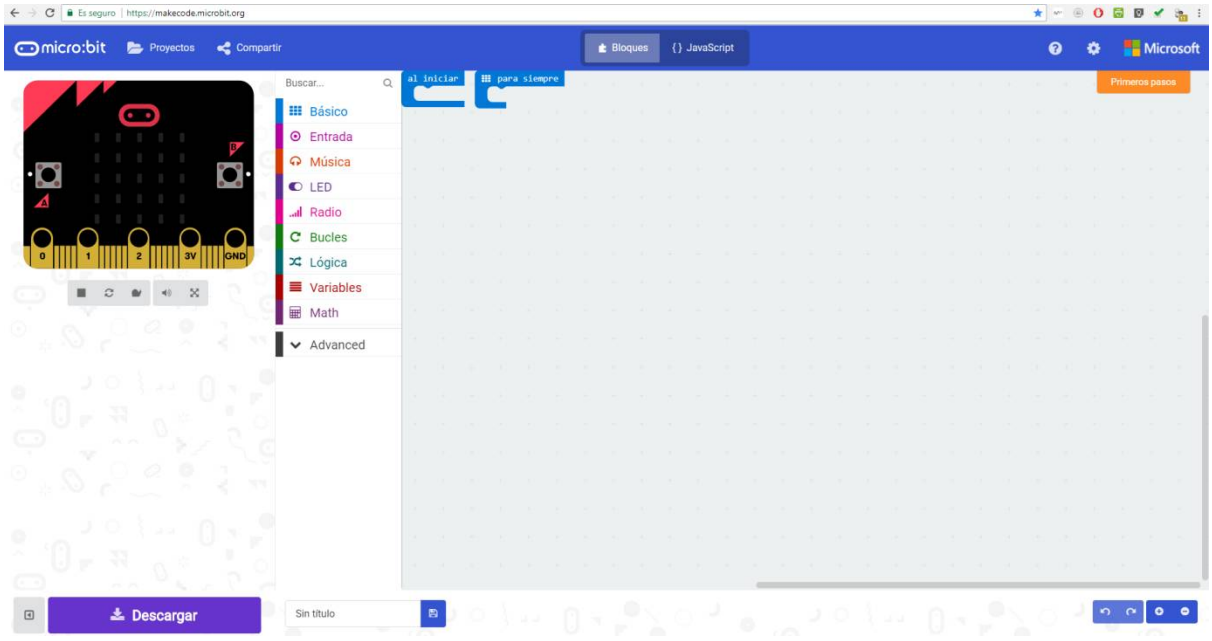
La micro:BIT incorpora:

- 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.

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:

4.1 Al pulsar sobre micro:bit lleva a la página https://microbit.org/code/ donde encontrar ejemplos de codificación en diferentes lenguajes de programación.
4.2 La sección Proyectos permite elegir entre:

a) 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.
b) Proyectos. En esta sección se encuentran diferentes tutoriales, paso a paso, donde aprender a codificar la micro:BIT usando MakeCode.
c) Ejemplos. Se ofrecen varios códigos de ejemplo.

4.3 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.

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

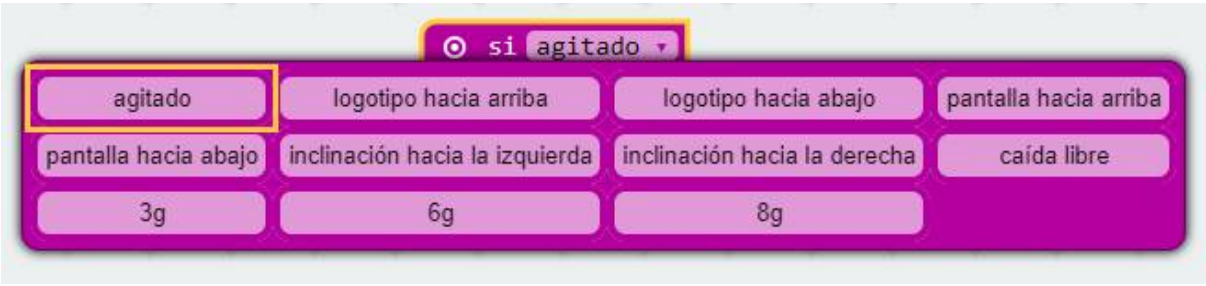
4.5 En el extremo derecho de la Barra de Herramientas se encuentran las secciones de Ayuda, Configuración, acceso a la página principal de MakeCode y en naranja la sección primeros pasos, que ofrece un tutorial de uso.

En la barra inferior aparecen los siguientes atajos:
4.6 El primer icono de la izquierda, permite ocultar el simulador, lo que permite tener más espacio disponible en el Área de Programación.
4.7 Descargar. Descarga el código a la micro:BIT. Se usará un nombre por defecto si no se asignó uno previamente.
4.8 Pulsando sobre el icono del disquete, permite asignar un nombre y descargar el programa al disco duro.

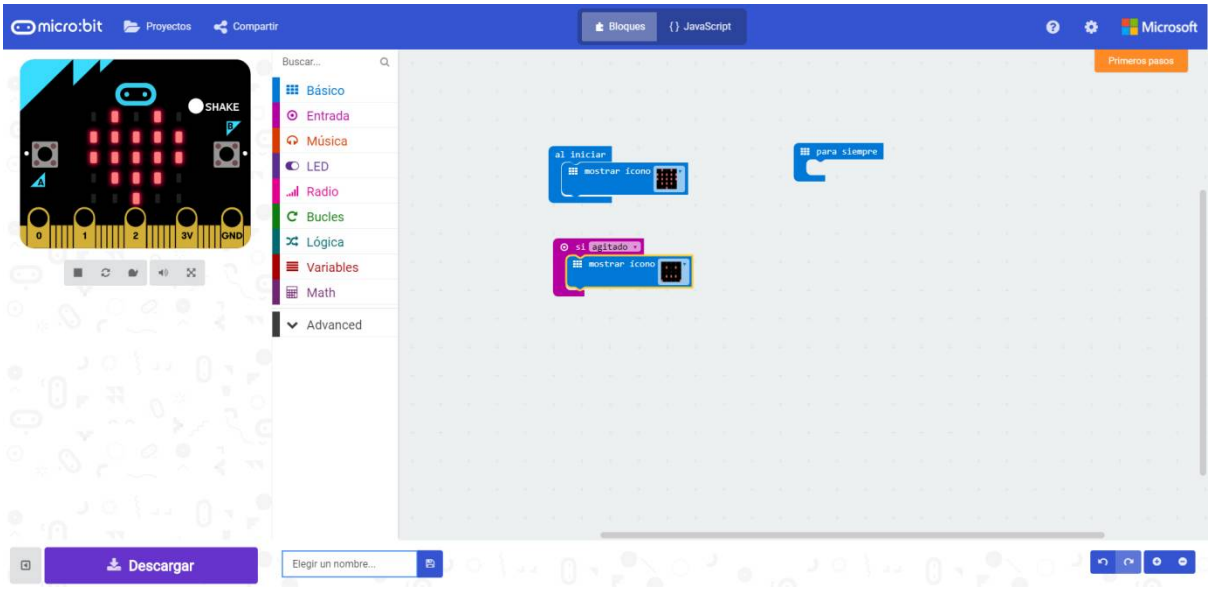
4.9 Las flechas permiten deshacer y rehacer los pasos de codificación.
4.10 En el extremo derecho inferior aparece el icono “+” que aumenta el zoom del área de programación y el icono “-” que disminuye el zoom.

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.

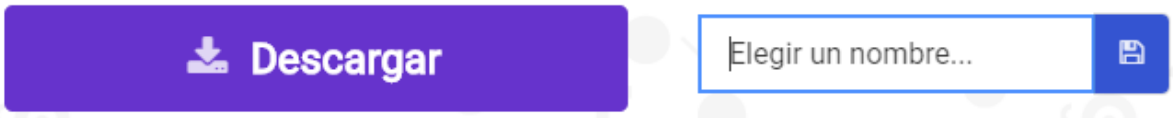


Para crear el programa, se deberá ir arrastrando los diferentes bloques al área de programación.
Se podrá comprobar el funcionamiento del programa en el simulador situado a la izquierda. Además según los bloques introducidos, aparecerán diferentes opciones que permiten interactuar con el simulador.



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.

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.

~ 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.

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.

Reto 1. Hola Mundo.

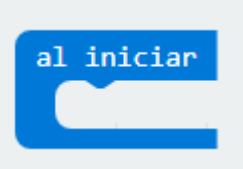
Cuando se aprende a programar, se llama “Hola Mundo” al primer programa que se crea. Este código imprime el texto «¡Hola Mundo!» en un dispositivo de visualización.

Objetivo.

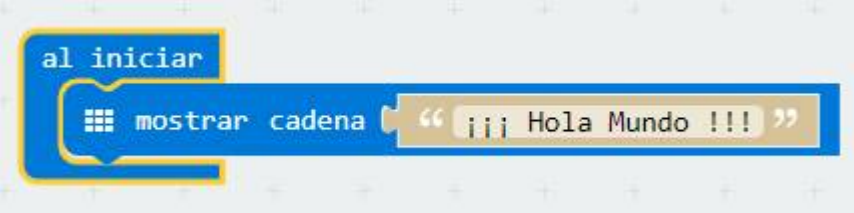
Al conectar la micro:BIT, se debe mostrar el texto «¡¡¡Hola Mundo!!!» en el panel LED de la tarjeta.

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.

Propuesta.

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

Reto 2. Animando los iconos.

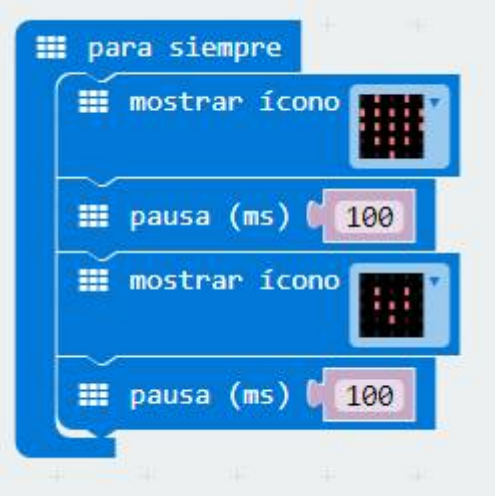
La animación es una técnica que logra crear sensación de movimiento a imágenes estáticas. Se consigue mediante una secuencia de imágenes ordenadas, que al ser mostradas consecutivamente, consiguen generar la ilusión visual de movimiento.

Objetivo.

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

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.

Propuesta.

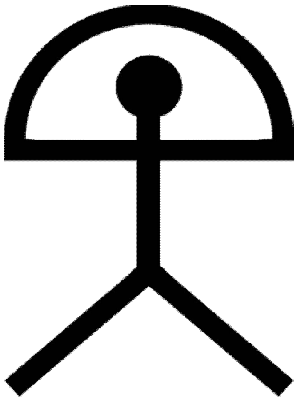
Hacer un código que muestre la animación de una tijera cortando.

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.

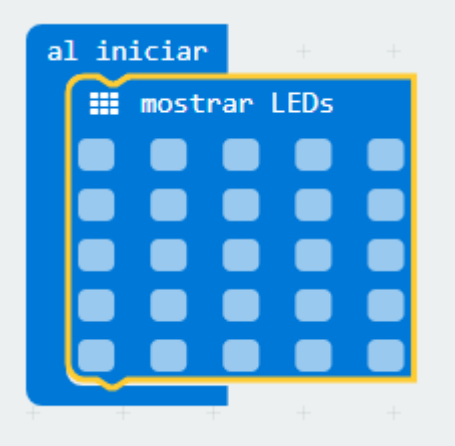
Objetivo.

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.

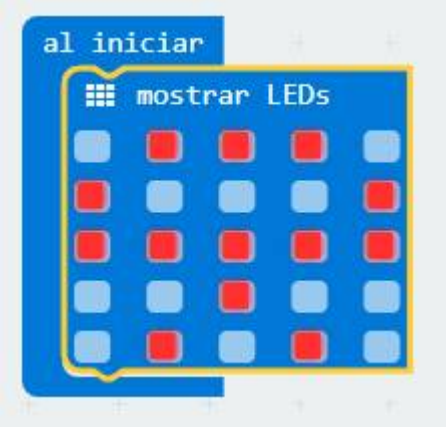


Descripción del código.

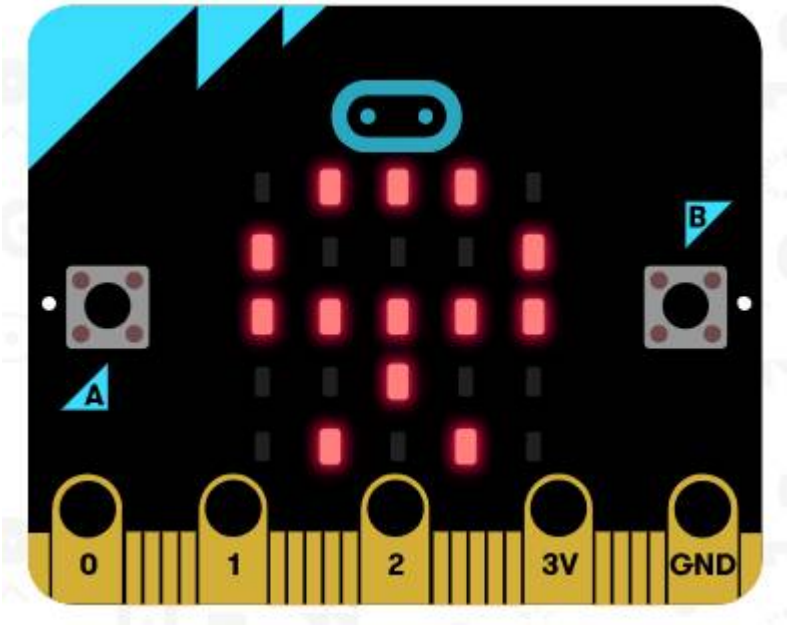
Se usará el evento al iniciar, situado en la categoría Básico .
Situar en el interior el bloque anterior, la instrucción “mostrar LEDs”, este se localiza en la categoría Básico. Este bloque permite marcar de forma independiente los LEDs que se quieren encender.



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.

Sensor de temperatura

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

Reto 4. Termómetro digital.

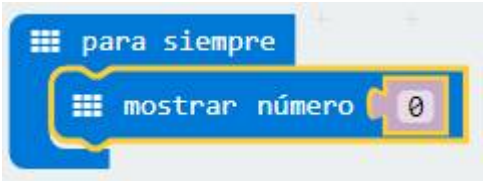
Un termómetro es un instrumento que sirve para medir la temperatura; el más habitual consiste en un tubo capilar de vidrio cerrado y terminado en un pequeño depósito que contiene una cierta cantidad de mercurio o alcohol, el cual se dilata al aumentar la temperatura o se contrae al disminuir y cuyas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

Descripción del código.

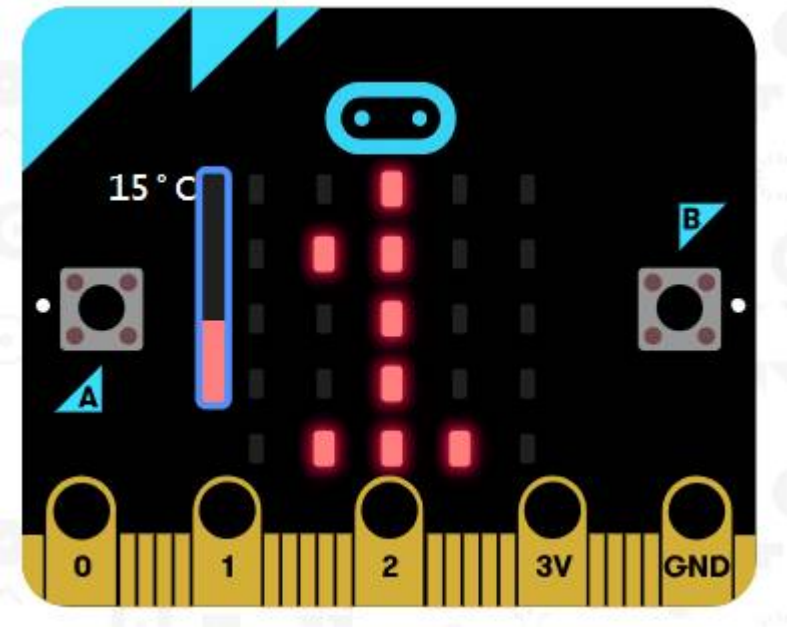
Se propone iniciar el programa usando el evento “para siempre”. Añadir el bloque, localizado en la categoría Básico, mostrar número. Este bloque muestra en la pantalla LED el número introducido, desplazándose si es mayor de 1 cifra.



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 5. Aviso de placas de hielo.

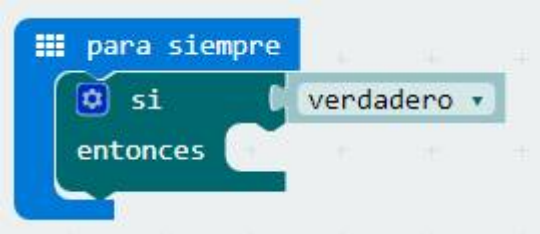
La mayoría de coches incorporan en el salpicadero, junto a la pantalla que marca la temperatura exterior, un testigo con forma de copo de nieve, que avisa de la posibilidad de que haya placas de hielo en la carretera. Este testigo se suele iluminar cuando la temperatura baja de 3 grados centígrados.

Objetivo.

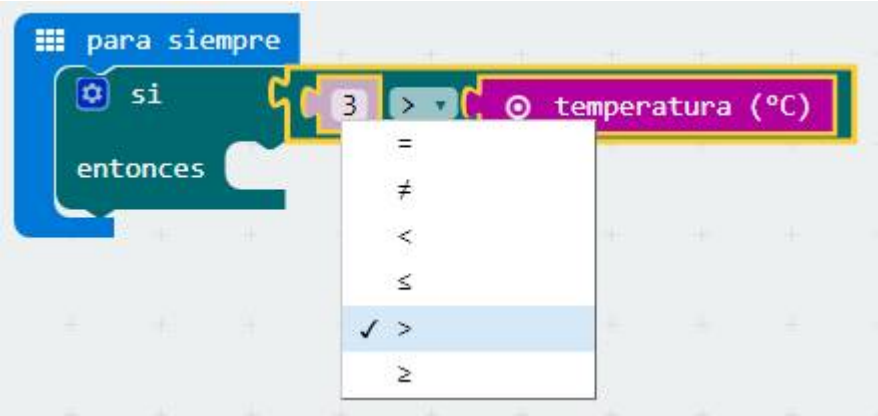
Mostrar en el panel LED de la micor:BIT un icono con forma de copo de nieve cuando la temperatura esté por debajo de 3 grados.

Descripción del código.

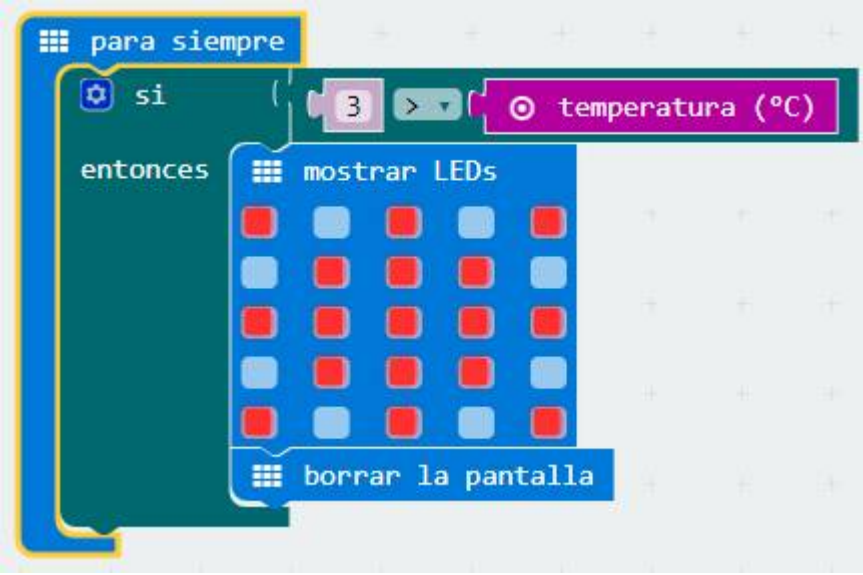
Para iniciar el programa, se propone usar el evento para siempre. Posteriormente se debe añadir el operador lógico si … entonces situado en la sección Lógica. Este bloque verifica si es verdad que se cumple una condición, si es así, ejecuta las instrucciones introducidas.



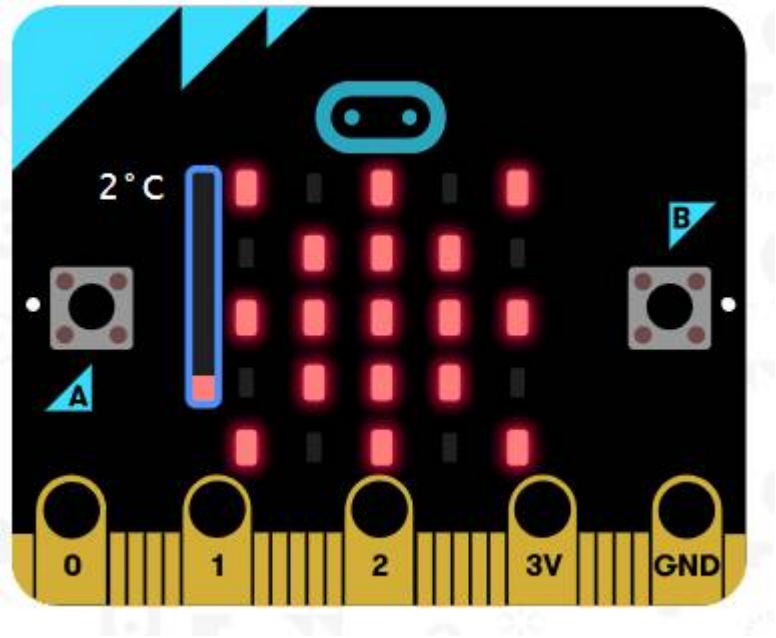
Para comprobar si la temperatura está por debajo de los 3º centígrados se usará el operador que devuelve verdadero siempre que 3 sea mayor que el valor reportado por el sensor de temperatura. El bloque se localiza en la categoría Lógica.



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.



Propuesta.

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

Reto 6. Temperatura óptima de una nevera

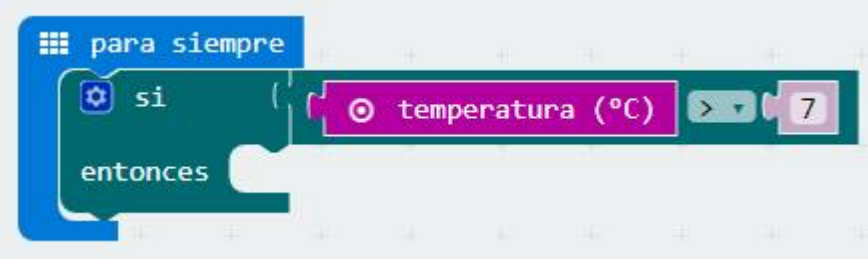
La temperatura óptima de un frigorífico es de 7ºC, mientras que la temperatura de un congelador debe estar entorno a los -18ºC. Algunas neveras incorporan un avisador acústico que se activa cuando la temperatura no es la óptima.

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

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.

Sensor de luminosidad

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

Reto 7. 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.

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.

Propuesta.

Una variante de este reto sería hacer que el brillo de los LED se adapte a la intensidad de la luz media por el sensor. A mayor intensidad recibida por el sensor, menor intensidad de iluminación y a menor intensidad medida, mayor luminosidad de los LED.

Pulsadores

Hay dos botones en la cara frontal de micro:bit (etiquetados como A y B). Puedes detectar cuando son pulsados de forma independiente o a la vez y ejecutar una acción en cada caso.

Reto 8. ¿Qué botón has pulsado?

A veces puede interesar usar un pulsador para iniciar, pausar o detener un código o subrutina. Algunos robots incorporan botoneras para esta finalidad.

Objetivo.

Crear un programa que muestre en pantalla la letra del pulsador que se ha accionado.

Descripción del código.

Se usará el bloque al iniciar para que al principio se muestre el texto “Pulsa un botón”. La micro:BIT no muestra caracteres acentuados, por lo que dejará un hueco si se pone la tilde en la palabra botón.
Posteriormente se usará el evento al presionar el botón A y se introducirá el bloque mostrar cadena donde se sustituirá el texto por defecto por la letra “A”.
Esto mismo se repetirá cuando se pulsa el botón B y la pulsación combinada de A+B.
El código quedaría de la siguiente forma:

Propuesta.

Crear una caja de música. Al pulsar el botón A haga hará sonar una melodía. El bloque necesario para hacer sonar una melodía se encuentra en la categoría Música.

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).

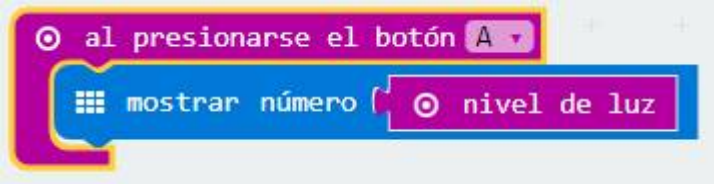
Objetivo.

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.

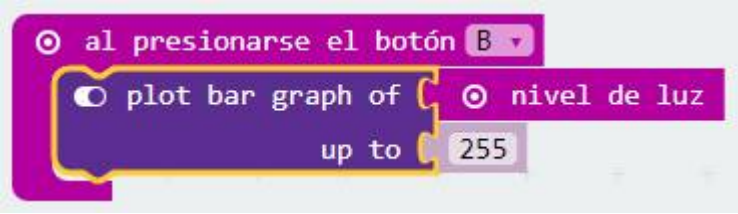
Descripción del código.

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

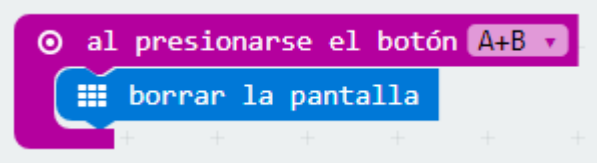
Al pulsar sobre el botón A se mostrará el valor numérico de la intensidad luminosa. Se usará el bloque mostrar número introduciendo el bloque nivel de luz como valor a presentar.



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 10. Modificar el brillo de los LEDs

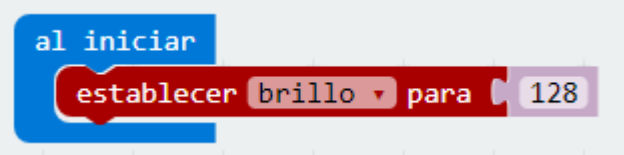
La mayoría de los teléfonos móviles incorporan la posibilidad de modificar la luminosidad de la pantalla, para adaptarse a la luz ambiente.

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.

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!

Reto 11. 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.

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.

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 12. 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

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.)

Reto 13. 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 (°) que ofrece el ángulo de desviación respecto el Norte Magnético.
Esperar 1 segundo y borrar la pantalla.

Propuesta.

Cuando la brújula apunte al norte magnético deberá aparecer una flecha apuntando hacia arriba. En caso contrario solo aparecerá el valor numérico.

Reto 14. 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.

Propuesta.

Hacer un código que ubique de forma más precisa el Norte Magnético.

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!.

Reto 15. 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.

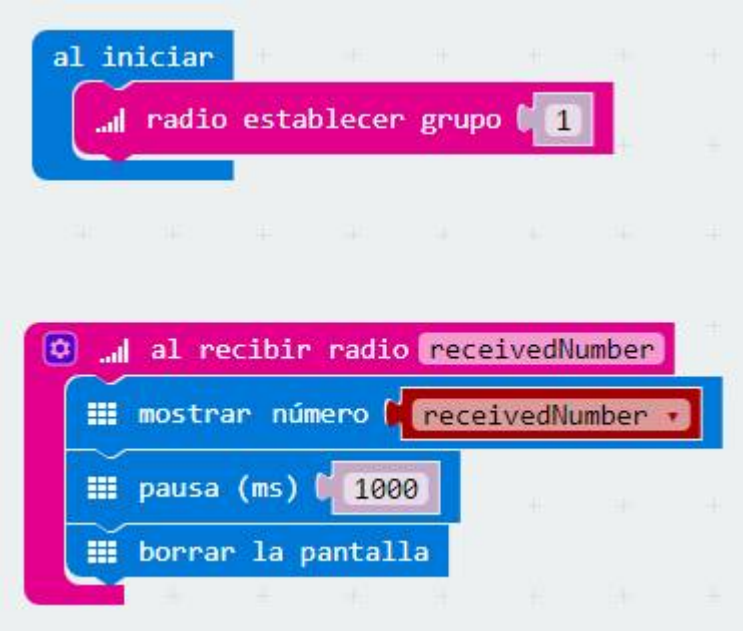
Diseñar un sistema inalámbrico de recolección de datos. La micro:BIT emisora se situará en el interior de una nevera y emitirá la temperatura. La unidad exterior mostrará el valor de la temperatura.

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.

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.

Reto 16. ¿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 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:

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.

~ Biografía del 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.

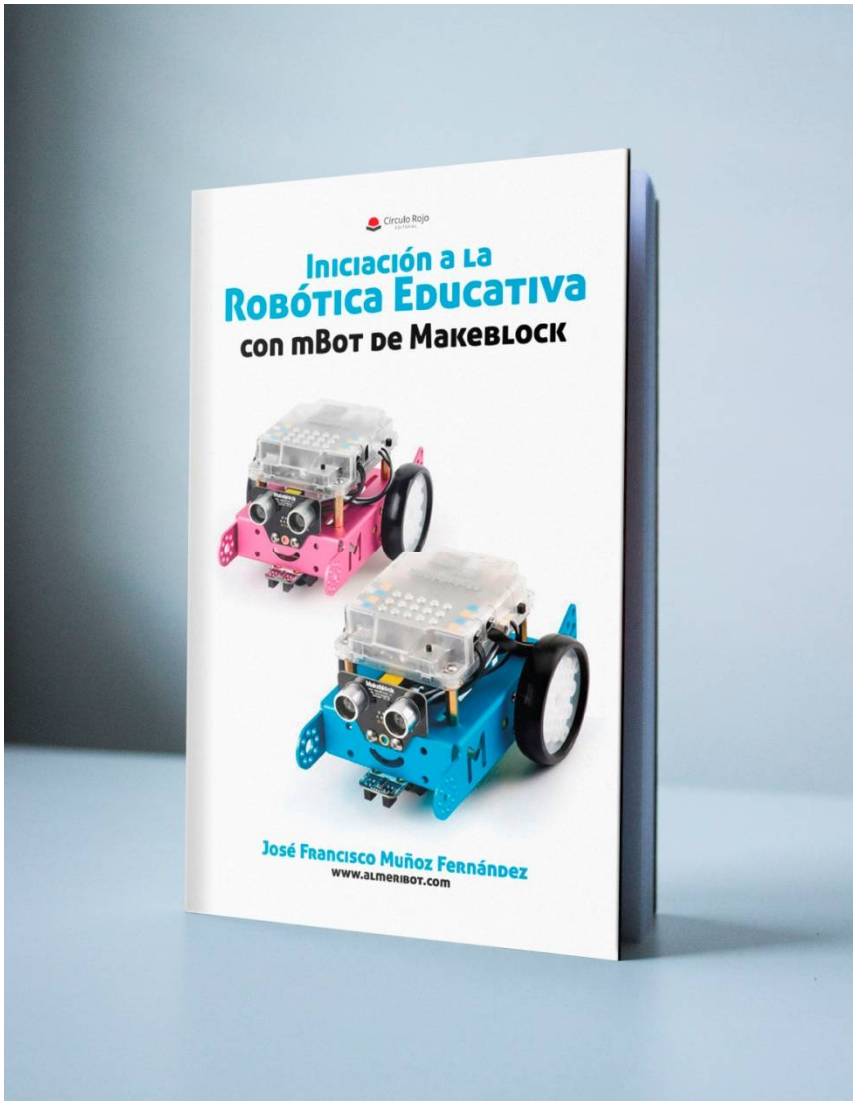
Tras años ejerciendo como arquitecto, decide dar más protagonismo a sus otras dos pasiones, la tecnología y la educación, comenzando a ejercer como profesor de robótica.

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.

~ Otros libros del autor ~

“Iniciación a la Robótica Educativa con mBot de Makeblock”

Disponible para su compra en:
https://www.makeblock.es/productos/libro-iniciacion-robotica-educativa-mbot/



~ También te puede interesar ~

"50 Proyectos con micro:bit"
Ernesto Martínez de Carvajal Hedrich • Primera Edición (2018) • 229 páginas a color

Disponible para su compra en:
http://microes.org/