

Sensores y automatismos simples

- [Interacción con micro:bit por medio de los pulsadores](#)
- [Bloques de entradas: sonómetro, magnetómetro y acelerómetro con una sola línea de código](#)
- [Bucles y eventos de tiempo: un termómetro](#)
- [Bloques condicionales: luz crepuscular](#)
- [Brújula digital](#)
- [Nivel de burbuja usando el giroscopio](#)

Interacción con micro:bit por medio de los pulsadores

Pulsadores mecánicos

El menú **Entradas** contiene los bloques que permiten captar datos de los sensores integrados en micro:bit. Para que la placa lleve a cabo acciones al pulsar los botones **A**, **B** ó **A y B** a la vez, disponemos del bloque de eventos denominado **al presionarse el botón**:

[Eventos de pulsación.jpg](#)

Al arrastrar el bloque anterior desde la caja de herramientas hasta el área de programación y seleccionar la opción de pulsación **A+B**, la imagen de la tarjeta cambia para mostrar la posibilidad de pulsar con el ratón sobre un nuevo botón virtual que simula la pulsación simultánea de los botones A y B.

[Pulsadores.png](#)

Para que micro:bit realice alguna acción específica al pulsar un botón, bastará con introducir los bloques de código necesarios dentro de un bloque de evento de pulsación. El siguiente programa mostrará una flecha apuntando hacia la izquierda al pulsar el botón **A**, una flecha apuntando hacia la derecha al pulsar el botón **B** y dos flechas al pulsar los botones **A+B**. Transcurrido 1 s, se borrará la pantalla.

[Programa de pulsadores2.jpg](#)

Los eventos **al iniciar** y **para siempre** han quedado vacíos simplemente porque no hemos previsto que se lleve a cabo ninguna acción inicial ni permanente.

Pulsador táctil

Desde la versión 2 de la tarjeta el **logo** situado en el anverso es un **sensor táctil**. El menú **Entradas** dispone de un bloque específico que puede ajustarse mediante un menú desplegable para capturar el evento deseado.

[Sensor Logo.jpg](#)

Como puede verse, los eventos asociados al logo son **pulsar**, **tocar**, **soltar** y **mantener pulsado**. **Cada vez que se toque el logo se producirán tres eventos:**

- Evento **al tocar el logotipo** en el instante en que se toca el logo.
- Evento **al soltar el logotipo** justo en el momento en el que se levanta el dedo del logo.
- Evento **al pulsar el logotipo** justo en el momento en el que se levanta el dedo del logo siempre que la pulsación haya sido corta.
- Evento **al mantener pulsado el logotipo** justo en el momento en el que se levanta el dedo del logo siempre que la pulsación haya sido larga.

El siguiente programa simula una **luz temporizada**.

[Luz temporizada.jpg](#)

Mada más tocar el logo, el evento **al tocar el logotipo** hará brillar la pantalla con una luz tenue. Al levantar el dedo, el evento **al soltar el logotipo** iluminará la pantalla a plena potencia. Si la pulsación ha sido corta, el evento **al pulsar el logotipo** mantendrá la luz encendida durante 3 s. Si la pulsación ha sido larga, el evento **al mantener pulsado el logotipo** mantendrá la luz encendida durante 15 s.

Reto ¿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.

Pruébalo aquí:

https://makecode.microbit.org/---run?id=_ibiDd0EKA15r

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.

Bloques de entradas: sonómetro, magnetómetro y acelerómetro con una sola línea de código

Sensor de sonido

Para usar los sensores integrados de micro:bit no es necesario cargar ni inicializar bibliotecas de código. Las medidas de los sensores se encuentran disponibles en el menú **Entrada** en forma de **variables**. En el lenguaje de bloques las variables se representan mediante rectángulos de extremos redondeados.

La versión 2 de micro:bit dispone de un **micrófono** que además de grabar sonidos puede medir el nivel de ruido. La **variable nivel de sonido** nos dará lecturas entre 0 (nivel mínimo de sonido) y 255 (nivel máximo). Estos niveles no se corresponden con ninguna unidad física, como el dB por ejemplo, y deben usarse con fines comparativos.

La razón de que algunos sensores de micro:bit proporcionen medidas entre 0 y 255, es que con un byte (8 bits) sólo se pueden representar $2^8 = 256$ números distintos, es decir, el 0 y los 255 primeros números naturales.

El bloque **plot bar graph of** del menú **LED** permite construir un sencillo medidor de sonido ambiente. Como 255 es un valor muy alto de intensidad de sonido, ajustamos el rango de medida de la barra, **up to**, a la mitad, es decir, a 128. Así la barra reflejará mejor el sonido de una voz o el sonido ambiental normal.

[Programa de barra de sonido.jpg](#)

La variable **nivel de sonido** debe arrastrarse desde el menú **Entrada**. En el momento en el que la variable haya sido introducida en el programa, el simulador de micro:bit cambiará, mostrando una **barra ajustable** que simulará el nivel de sonido captado por el micrófono. El valor numérico del nivel de sonido simulado también será mostrado al lado del **LED** del micrófono.

[microbit-Sonido.png](#)

Tras descargar el programa en la placa real, la matriz de LED representará continuamente el sonido recogido por el micrófono en forma de barra vertical. El **LED del micrófono** iluminado indicará que micro:bit está captando sonido.

Barra de sonido.gif

Una tarjeta micro:bit ejecutando este programa puede agotar un par de pilas alcalinas IEC R03 (AAA) en unas 40 horas (Frost 2018). Para **ahorrar energía** y prolongar la autonomía del medidor podemos reducir tanto el **brillo de la pantalla** como el **número de medidas por segundo** que realiza el sensor. Para conseguir esto último introduciremos en el bucle **para siempre** un bloque **pausa (ms)**. Si el bloque se ajusta a 100 ms, el sensor sólo realizará 10 mediciones del nivel de sonido cada segundo.

Ahorro de energía.jpg

Magnetómetro y acelerómetro

Con una mínima modificación, el código anterior puede usarse para monitorizar aquellas **magnitudes que puedan variar rápidamente**. Por ejemplo, podemos usar el sensor integrado de campo magnético (magnetómetro) para medir el campo magnético de la Tierra, el de una imán o el de una masa de hierro.

Podemos acceder al sensor mediante la variable **fuerza magnética (μT)**, que proporciona la **inducción magnética** medida en **microtesla**. Al cargar el programa, micro:bit comenzará a medir el campo magnético terrestre que varía, según la localización, entre 25 y 65 μT . Nótese que el magnetómetro no limita sus medidas al valor de 255.

[Programa de inducción magnética.jpg](#)

Otra medida interesante es la de la **aceleración de la placa**. La variable de acceso al acelerómetro se llama **aceleración (mg)** y proporciona las aceleraciones medidas en milésimas de g. Cuando la placa esté en reposo medirá la **aceleración de la gravedad terrestre**, que es de 1 g. Los movimientos bruscos de la placa en cualquier dirección deberían alterar el valor medido.

[Programa de aceleración.jpg](#)

Bucles y eventos de tiempo: un termómetro

Programación del termómetro

El siguiente programa proporcionará el valor numérico de la **temperatura ambiente en grados Celsius** cada vez que se pulse el **botón A**. La variable **temperatura (°C)** se encuentra disponible en el menú **Entrada**.

Termómetro de pulsador.png

El evento **al iniciar** comienza encendiendo un punto de la matriz de LED a modo de **piloto de funcionamiento**. El brillo de la pantalla se ajusta a un valor bajo para conseguir un **bajo consumo de energía**.

Por otro lado, **cada vez que se pulse el botón A**, ocurrirá un evento del tipo **al presionarse el botón A** que subirá el brillo de la pantalla al máximo (255) y mostrará la temperatura mediante una cadena de texto deslizante, para volver más tarde a dejar encendido el piloto de funcionamiento a bajo brillo.

La temperatura indicada será algo superior a la ambiental. Esto ocurre porque el sensor de temperatura se encuentra en el microprocesador y éste se calienta ligeramente cuando la placa está en funcionamiento. Martínez de Carvajal (2019) establece el error medio en 3°C, por lo que habrá que restar 3 al valor mostrado en pantalla para obtener la temperatura real. La **manipulación de la placa con los dedos** también contribuye al calentamiento y al error en la medida de la temperatura.

Podemos añadir un **evento de tiempo** para que el termómetro muestre la temperatura cada cierto tiempo. Para ello debemos usar el evento **cada ms**, dentro del menú **Bucles**.

Termómetro de evento.png

Cada 30000 ms, o cada **30 segundos**, el programa mostrará la temperatura aunque no haya sido pulsado el botón A. Dentro del bucle de tiempo no se sube el brillo, así que los dígitos se mostrarán con bajo brillo para ahorrar batería.

Es posible mejorar un poco más el programa haciendo que micro:bit muestre tres veces la temperatura cada 30 segundos. Podríamos repetir sin más la sentencia **mostrar cadena temperatura (°C)** tres veces dentro del bucle de tiempo, pero en su lugar vamos a usar un bucle del tipo **repetir veces** , que también se encuentra en el menú **Bucles**.

[Evento de tiempo y bucle.png](#)

Nótese que el programa ejecuta un bucle cada 30 segundos, y que dentro de ese bucle se ejecuta otro bucle que muestra la temperatura tres veces seguidas. Al hecho de introducir un bucle dentro de otro se le llama **anidar bucles**.

Mejorando la lectura del sensor de temperatura

Para corregir el error de 3 grados Celsius en la lectura del sensor, bastará con restar 3 al valor de la variable **temperatura (°C)**. El menú **Matemática** contiene bloques para realizar operaciones aritméticas. Si se usa el bloque de resta - dentro de los bloques **mostrar cadena** resulta sencillo realizar la corrección necesaria. El código del termómetro completo quedará:

[Termómetro corregido.png](#)

Los dos bloques **mostrar LED** han sido sustituidos por dos bloques **graficar x y** para conseguir que el código sea algo más compacto.

Bloques condicionales: luz crepuscular

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.

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

Sonómetro de pulsador.png

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**

Brillo adaptativo.png 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.

Luz crepuscular.png

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.

Caras de sonido.png

Así pues, si el nivel de sonido es inferior a 64, la pantalla mostrará 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.

Brújula digital

Una 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.

[Brújula digital.png](#)

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:

[Rosa de los vientos.png](#)

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.

Calibrando el compás

Al igual que la brújula de un teléfono móvil, el compás de micro:bit necesita ser calibrado para proporcionar medidas fiables. Los puntos suspensivos **... más** bajo el menú **Entrada** despliegan otro menú que contiene el bloque **calibrar brújula**.

[Calibrar brújula.png](#)



Añadiendo el **evento** mostrado, nuestra brújula digital permitirá ser calibrada cada vez que se pulse el botón B.

El proceso de calibración es muy simple, ya que sólo requiere mover la placa en todas las direcciones hasta que todos los **LED** de la pantalla estén iluminados.

[LED iluminados.png](#)

Una 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 315° o menor que 45° , 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**.

[Brújula norte.png](#)

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.

[Brújula 4 direcciones.png](#)

Nótese que cuando el logo de micro:bit apunta al este, dirección de la brújula ($^\circ$) < 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.

[este-oeste.png](#)

Nivel de burbuja usando el giroscopio

Mediante el **sensor de fuerza**, micro:bit puede determinar para cada uno de los tres ejes coordenados las proyecciones de la aceleración de la gravedad y, a partir de ellas, el **giro de la placa** con respecto al plano horizontal.

Vamos a usar la medida del giro de la placa para programar un sencillo nivel de burbuja. La burbuja será un punto luminoso en la pantalla **LED** de micro:bit. Cuando el punto se encuentre en el centro de la pantalla, cuyas coordenadas son (2,2), micro:bit estará nivelado. Si micro:bit está desnivelado hacia la izquierda o hacia la derecha, el punto se dibujará desplazado en esas direcciones.

La estructura del código, compuesto por múltiples sentencias condicionales, es muy similar al de la brújula analógica presentada en el apartado anterior. [Nivel.png](#)

El código consta de **un único evento temporal** dentro del cual se evalúa el giro de la placa cada 250 ms y que, en función del ángulo de inclinación, enciende el punto correspondiente. Por ejemplo, si la placa se inclina hacia la izquierda con una rotación inferior a -5° , se encenderá el punto situado más a la izquierda, cuyas coordenadas son (0,2). En caso contrario, si la placa está inclinada hacia la izquierda menos de -2° , se encenderá el siguiente punto, de coordenadas (1,2), y así sucesivamente.

[Tarjeta nivel.png](#)