



Universidad  
de Alcalá

DEPARTAMENTO DE ELECTRÓNICA

# Sistema de desarrollo micro:bit

**D. Julio Pastor Mendoza** ([julio.pastor@uah.es](mailto:julio.pastor@uah.es))

Profesor del Departamento de Electrónica (UAH)

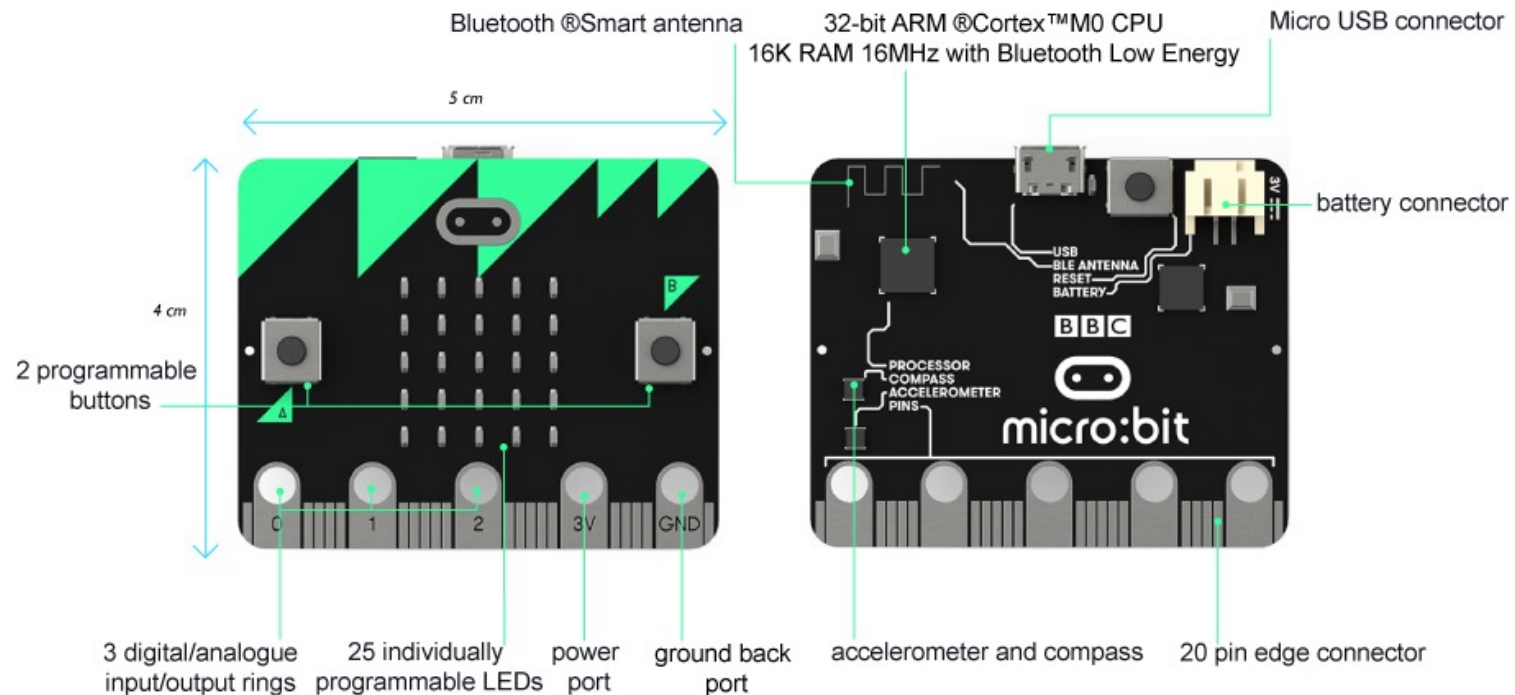


# Tarjeta micro:bit

Tarjeta micro:bit <http://microbit.org/es/>



- Plataforma educativa desarrollada por la BBC para colegios británicos
- Gran comunidad de usuarios y materiales didácticos (en inglés)
- Se programa en Bloques, Javascript, Python, ...

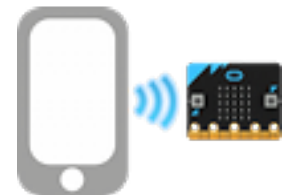
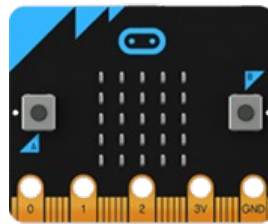
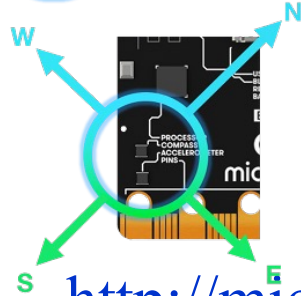
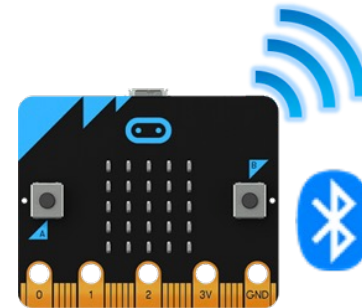
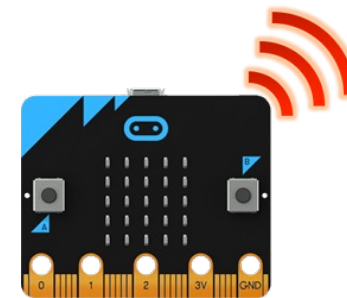
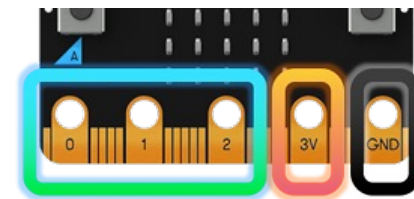
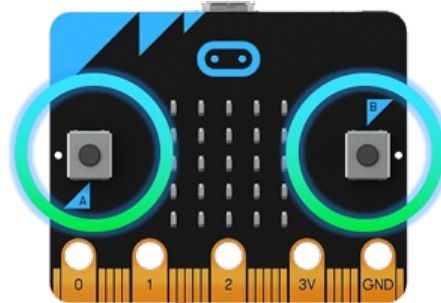
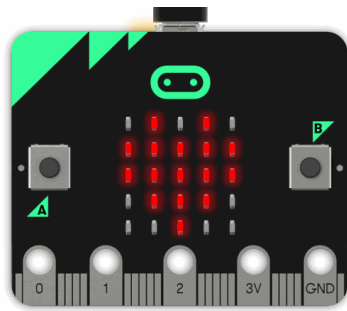




# Tarjeta micro:bit

S  
i  
s  
t  
e  
m  
a  
  
m  
i  
c  
r  
o  
  
b  
i  
t

## Sensores, entradas y salidas



<http://microbit.org/es/>



# Tarjeta micro:bit



S  
i  
s  
t  
e  
m  
a  
  
m  
i  
c  
r  
o  
  
b  
i  
t

**Micro USB**  
MSC, UART, CMSIS-DAP  
Drag-and-drop programming

**5x5 LED Matrix**

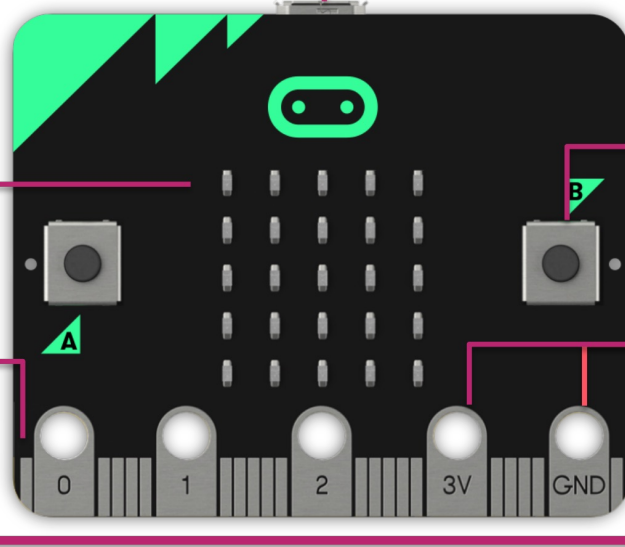
**Digital/analog IO**  
Muxable to SPI, UART, I2C

Pads for crocodile clips  
Holes for banana plugs

**User buttons**

**External supply**  
Regulated 3.3V in or battery out

**Edge Connector**



**2.4GHZ Antenna**  
Bluetooth low energy  
Gazell

**Nordic nRF51822**

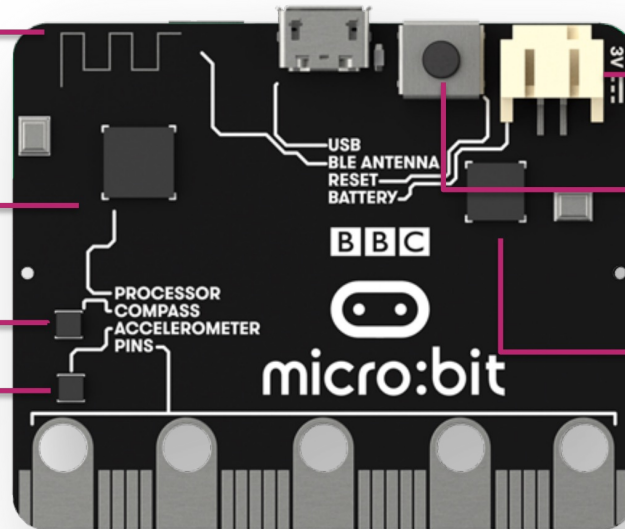
**Magnetometer**  
Freescale MMA8652

**Accelerometer**  
Freescale MAG3110

**Battery connector**  
JST connection for 3V

**Reset Button**

**Freescale KL26Z**  
USB Interface chip



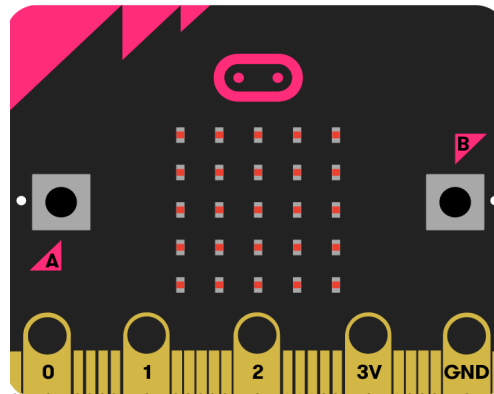


# Tarjeta micro:bit



## Conexiones externas de la tarjeta

- La tarjeta tiene **25 conexiones externas** que se denominan pines
  - 5 pines grandes
  - 20 pines pequeños.
- Están en un lateral de la tarjeta y la parte dorada son las conexiones.
- Hay 5 pines grandes que tienen un orificio para facilitar la conexión.
- Hay otros 20 pines más pequeños que se pueden usar cuando se conectan en una tarjeta de expansión.



<https://makecode.microbit.org/device/pins>



# Tarjeta micro:bit



## Pines grandes

- Hay 5 pines grandes que tienen un orificio para facilitar la conexión.
  - Marcados como 0, 1, 2, 3V y GND
  - Se pueden conectar fácilmente pinzas de cocodrilo o bananas de 4mm
  - Los pines 0, 1 y 2
    - Son pines multifunción de entrada y salida (GPIO – General Purpose Input Output)
      - » Entradas y salidas digitales
      - » Entradas analógicas
  - 3V
    - Es la salida de 3V si es alimentada con USB
    - Puede ser la entrada de 3V si no se alimenta por USB
  - GND
    - Es la masa o referencia de tensión.
- Si se toca el pin de GND con una mano, y un pin con la otra, se detecta la pulsación.





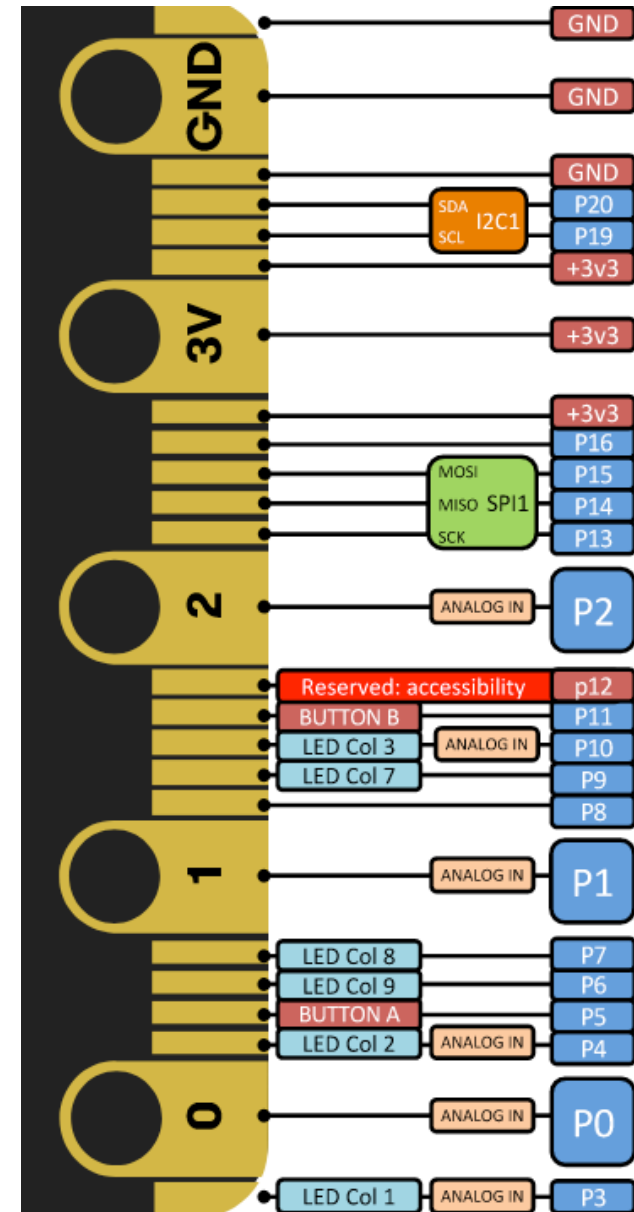
# Tarjeta micro:bit



## Pines pequeños

- Hay otros 20 pines más pequeños que se pueden usar cuando se conectan en una tarjeta de expansión.
- No están numerados en la tarjeta pero sí en las tarjetas de expansión.
- Algunos de los pines están conectados también a dispositivos internos de la tarjeta.
  - Pin 3, 4, 6, 7, 9, 10 están conectados a los LEDs. Pueden utilizarse como I/O digitales o ADC si LEDs apagados.

<https://makecode.microbit.org/device/pins>



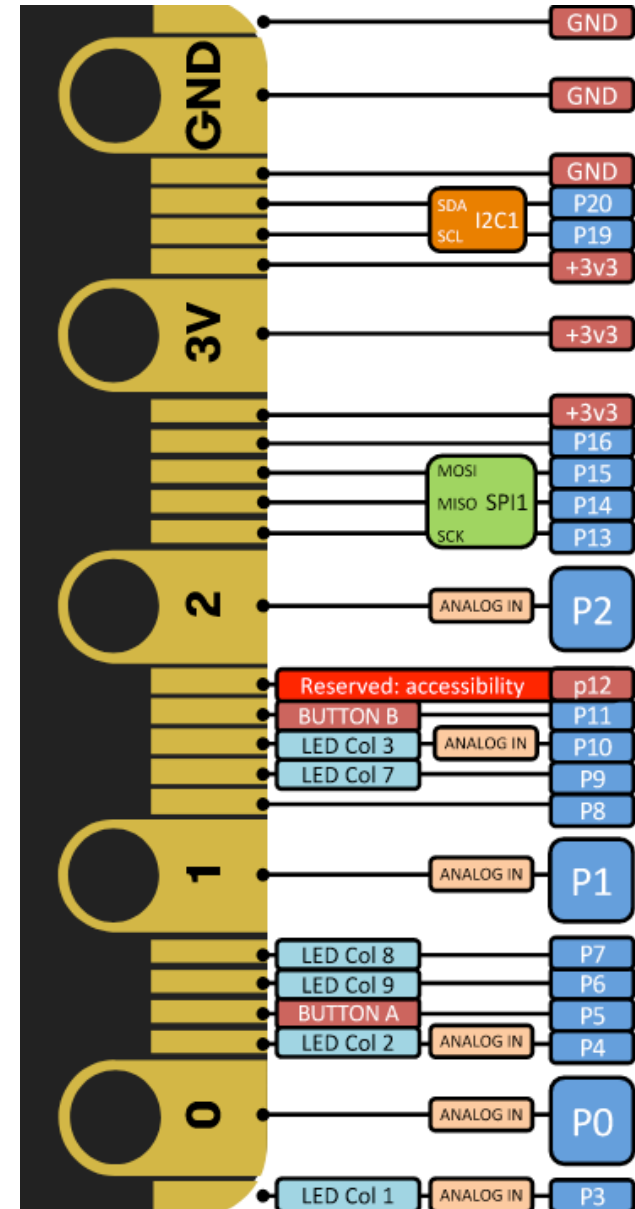
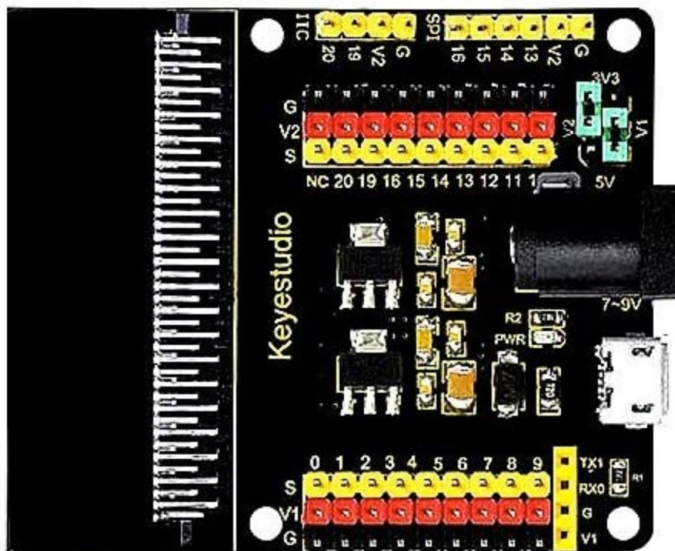




# Tarjeta micro:bit



## Pines pequeños: tarjeta de expansión



<https://makecode.microbit.org/device/pins>



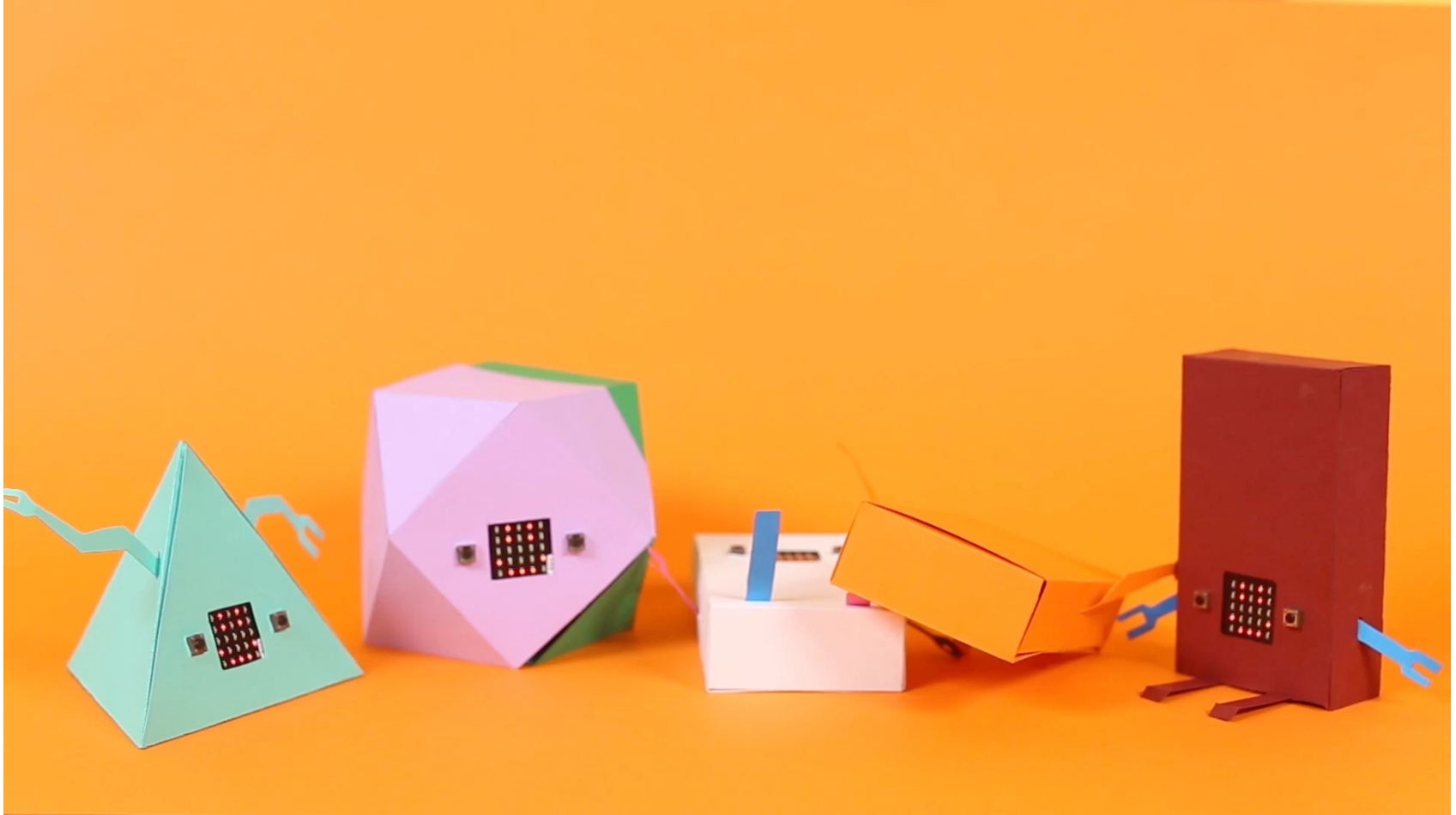


# Ejemplos de proyectos



S  
i  
s  
t  
e  
m  
a  
  
m  
i  
c  
r  
o  
  
b  
i  
t

## Micro Pet



<https://make.techwillsaveus.com/bbc-microbit/activities/micro-pet-advanced>

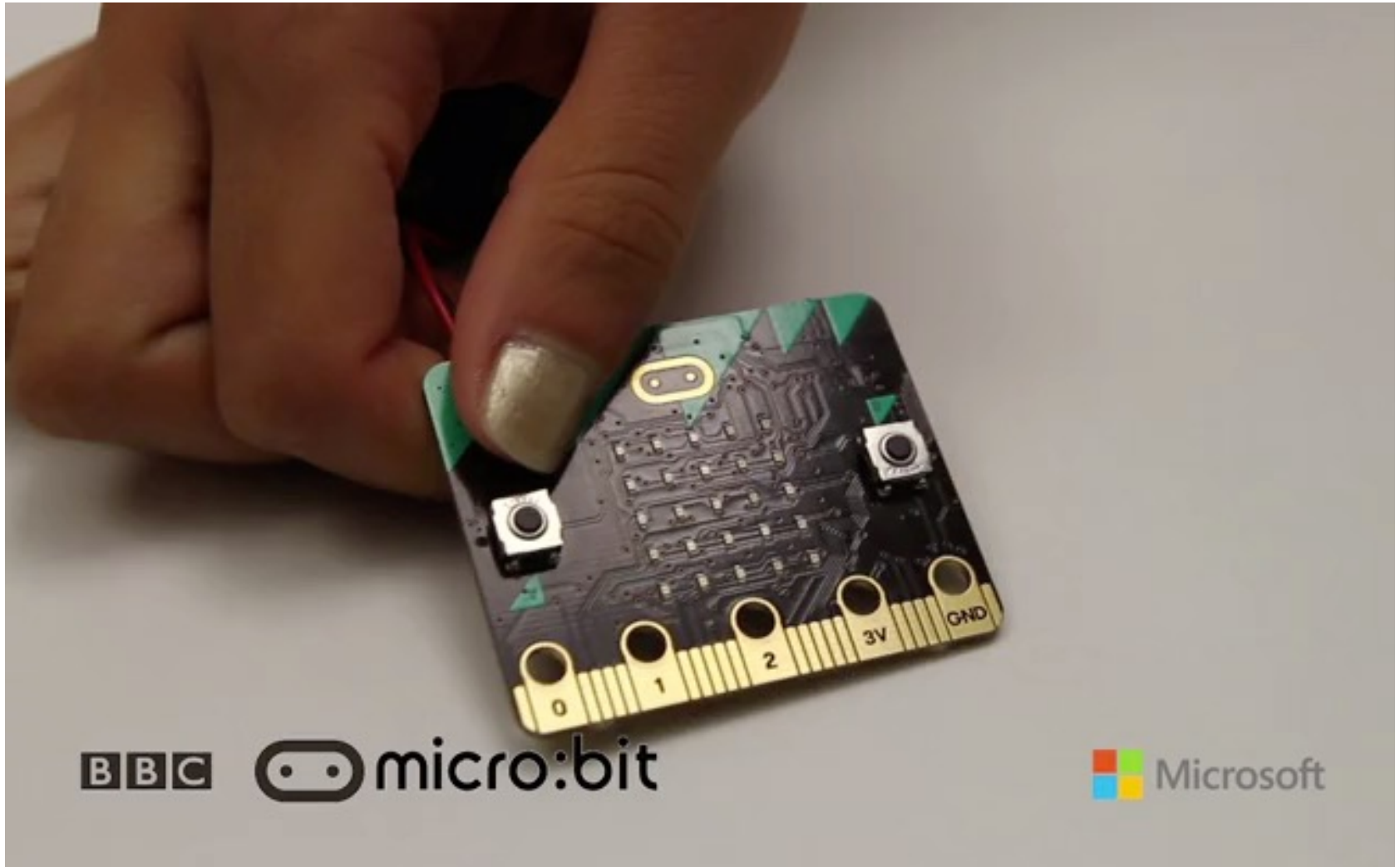


# Ejemplos de proyectos



S  
i  
s  
t  
e  
m  
a  
  
m  
i  
c  
r  
o  
  
b  
i  
t

## Die Roll



<https://www.microbit.co.uk/blocks/lessons/die-roll/activity>

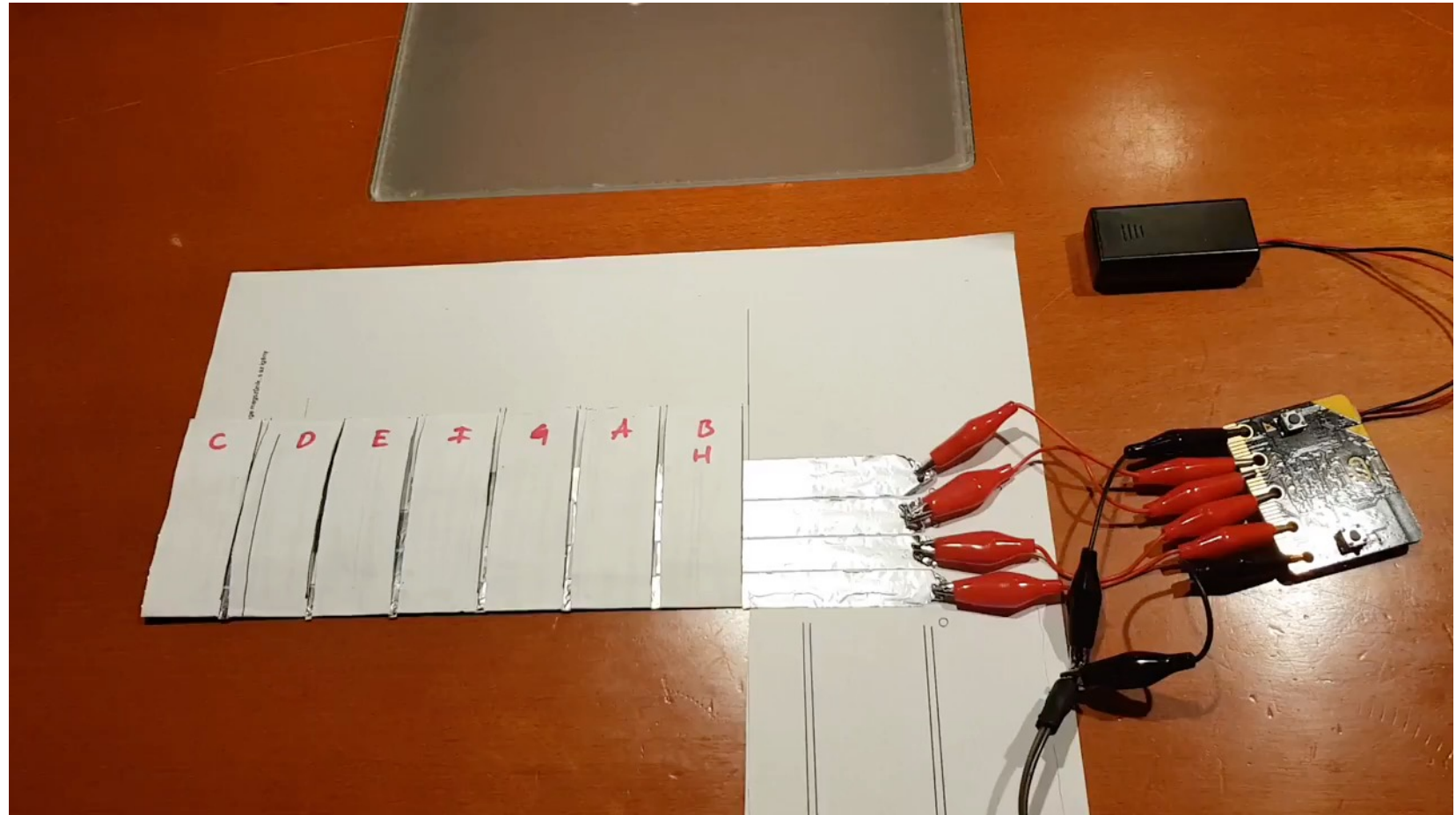


# Ejemplos de proyectos



S  
i  
s  
t  
e  
m  
a  
  
m  
i  
c  
r  
o  
  
b  
i  
t

## Paper Piano



<https://drive.google.com/file/d/1dgUUIEMWrP4NXYAxSSu5Rrd2ygb8OVcj/view>



# Ejemplos de proyectos



S  
i  
s  
t  
e  
m  
a  
  
m  
i  
c  
r  
o  
  
b  
i  
t

## Tetris



<http://www.101computing.net/bbc-microbit-tetris-game/>



# Entorno de programación



Sistema  
micro  
bit

Microsoft | micro:bit

Bloques JavaScript

Buscar...

- Básico
- Entrada
- Música
- LED
- Radio
- Bucles
- Lógica
- Variables
- Matemática
- Extensiones
- Avanzado
- Funciones
- Arreglos
- Texto
- Juego
- Imágenes
- Pines
- Serial
- Control

al iniciar para siempre

Descargar

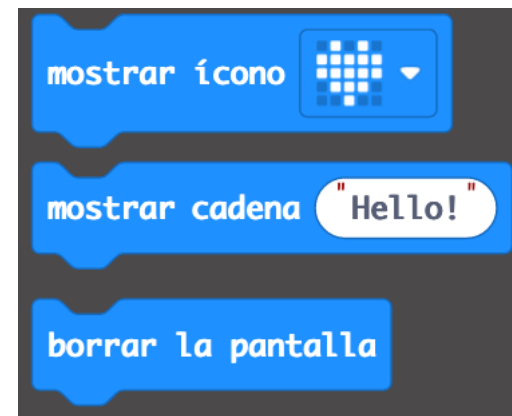
peuwba

Captura de pantalla



## Elementos básicos de programación

- **Escritura en la pantalla**
  - **Representa números o valores de variables**
  - **Permite dibujar lo que se quiere que aparezca en la pantalla**
  - **Visualiza iconos predefinidos**
  - **Representa cadenas de texto o el contenido de variables con texto**
  - **Borra la pantalla**





## Elementos básicos de programación

- Bucles principales y retardos







## Elementos básicos de programación

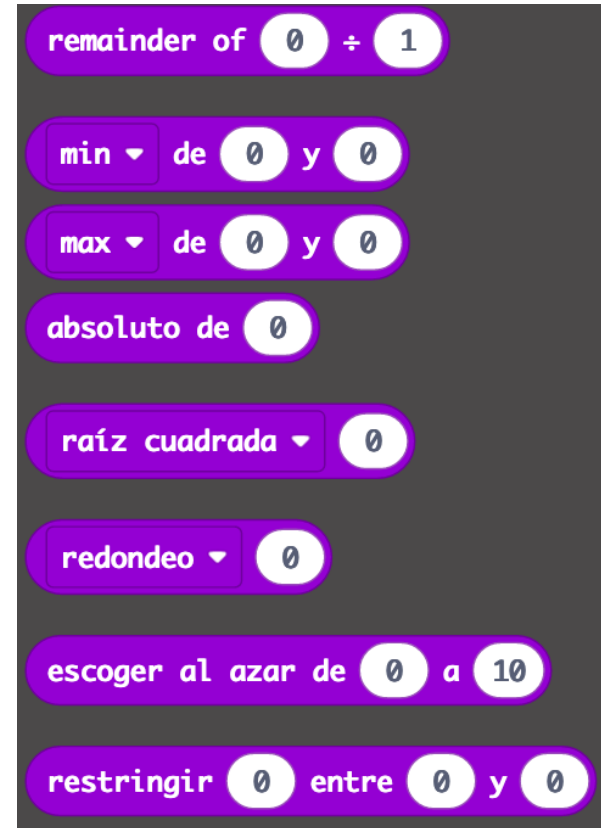
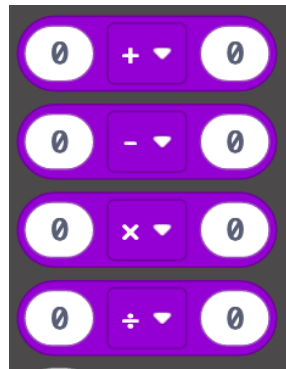
### – Matemáticas

- Operadores aritméticos

- +, -, \*, /, ...

- Otros operadores

- Número aleatorio, redondear, resto, ...



ajustar intervalo 0 de 0 hasta 1023 a intervalo de 0 hasta 4

escoge al azar verdadero o falso



## Elementos básicos de programación

### – Operadores Lógicos

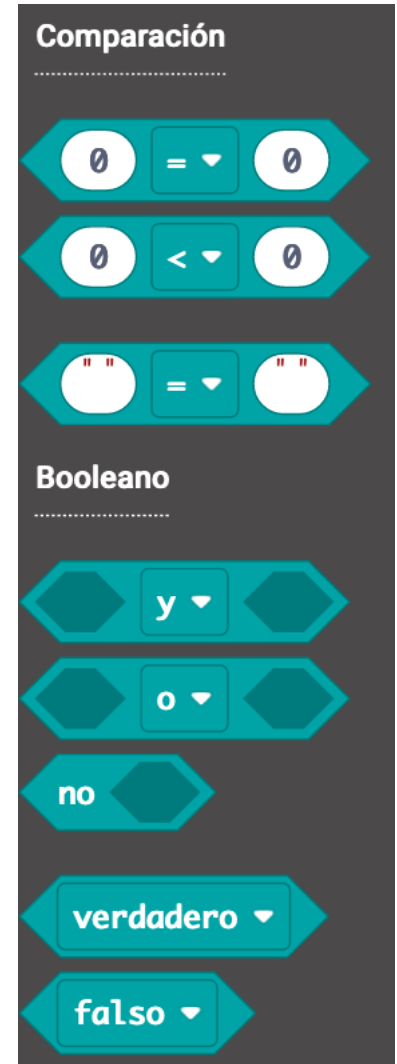
- Comparación

  - <, >, =

- Booleanos

  - O, Y, NO

### – Ejecución condicional





## Elementos básicos de programación

### – Bucles

- Repetir N veces
- Repetir mientras una condición sea verdadera
- Repetir N veces con información del número de iteración
- Repetir para todos los elementos de una lista con información de cada elemento
- Realizar algo cada cierto tiempo
- Comando para salir de un bucle o saltar a la siguiente iteración

repetir 4 veces  
ejecutar

mientras falso  
ejecutar

para index de 0 a 4  
ejecutar

para el elemento value de list  
ejecutar

cada 500 ms

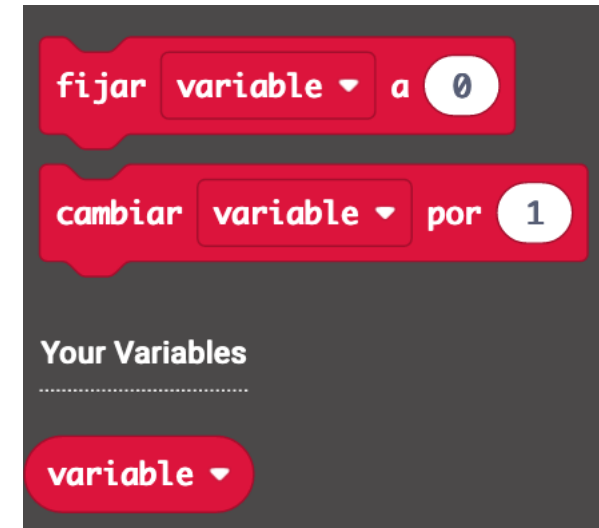
salir

continuar



## Elementos básicos de programación

- Variables
  - Guardar un valor en una variable
  - Incrementar o decrementar el valor





## Elementos básicos de programación

### – Operadores de texto

- **Número de caracteres de un texto**
- **Concatena textos**
- **Compara textos**
- **Extrae un sub texto de un texto**
- **Comprueba si una variable de texto está vacía**
- **Ver en qué posición está una letra**

longitud de "Hola"

unir "Hola" "Mundo" - +

compara "this" con ""

subcadena de "this" desde 0 de longitud 10

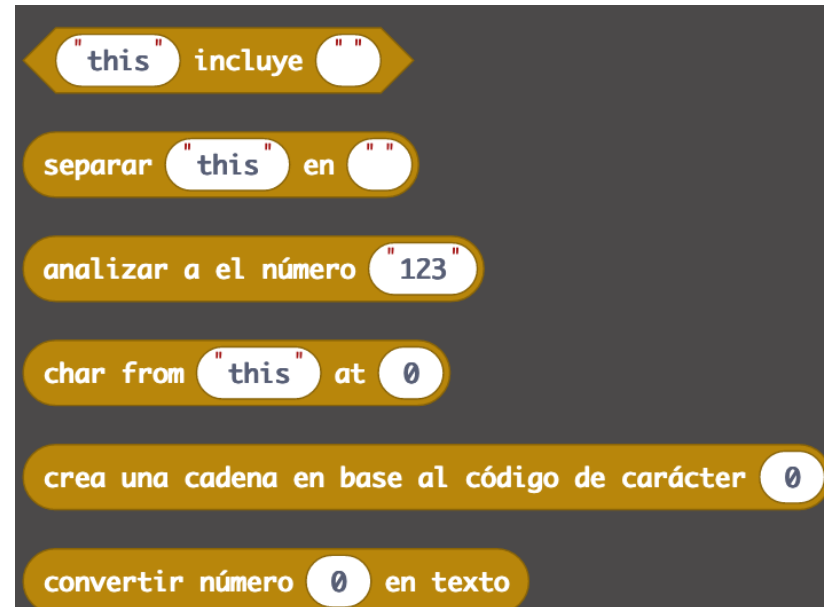
"this" está vacío

"this" hallar índice de ""



## Elementos básicos de programación

- Operadores de texto
  - Ver si un texto incluye un carácter
  - Separa el texto en varios trozos separados por un carácter
  - Transforma un texto con números en números.
  - Extrae un carácter de un texto
  - Obtiene el carácter que corresponde a un código ASCII
  - Convierte un número en texto





## Elementos básicos de programación

### – Listas

- Introducir datos en una lista de números
- Introducir datos en una lista de texto
- Leer el número de elementos de una lista
- Leer el valor de un elemento determinado de una lista

longitud del arreglo lista ▾

lista ▾ obtener el valor en 0

fijar lista ▾ a matriz de 0 1 - +

fijar lista de texto ▾ a matriz de "a" "b" "c" - +





## Elementos básicos de programación

### – Listas

- **Borrar un elemento determinado de una lista**
- **Extraer el último elemento de una lista**
- **Extraer el primer elemento de una lista**
- **Leer el valor de un elemento de la lista elegido de forma aleatoria**

list ▼ eliminar valor en 0

obtener y eliminar el último valor de list ▼

obtener y eliminar el primer valor de list ▼

obtener valor aleatorio de list ▼



## Elementos básicos de programación

### – Listas

- **Modificar el valor de un elemento de una lista**
- **Añadir al final un elemento de una lista**
- **Eliminar el último elemento de una lista**
- **Eliminar el primer elemento de una lista**

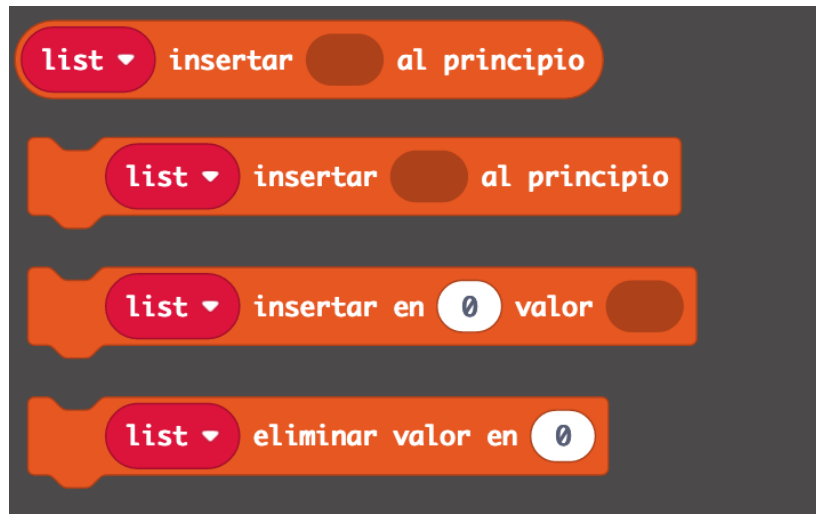




## Elementos básicos de programación

### – Listas

- **Añadir un elemento al inicio de una lista**
- **Insertar un elemento en una posición determinada de una lista**
- **Eliminar el elemento de una posición determinada de la lista**
- **Obtener el número de posición de un dato determinado en una lista**
- **Invertir la lista**





## Elementos básicos de programación

### – Funciones

#### Editar función

Añadir un parámetro

🔗 Texto

⚡ Booleano

📅 Número

☰ Array

📍 LedSprite

🖼️ Imagen





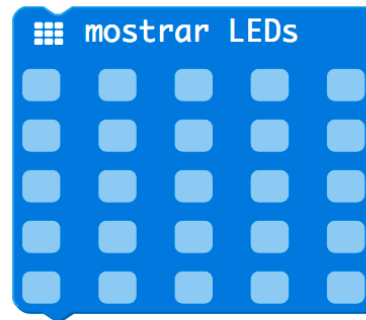
# Tarjeta micro:bit



## Utilizando la matriz de LEDs

### – Mostrar LEDs

- Visualiza un patrón determinado en los LEDs
  - Después de mostrar los LEDs espera 400ms

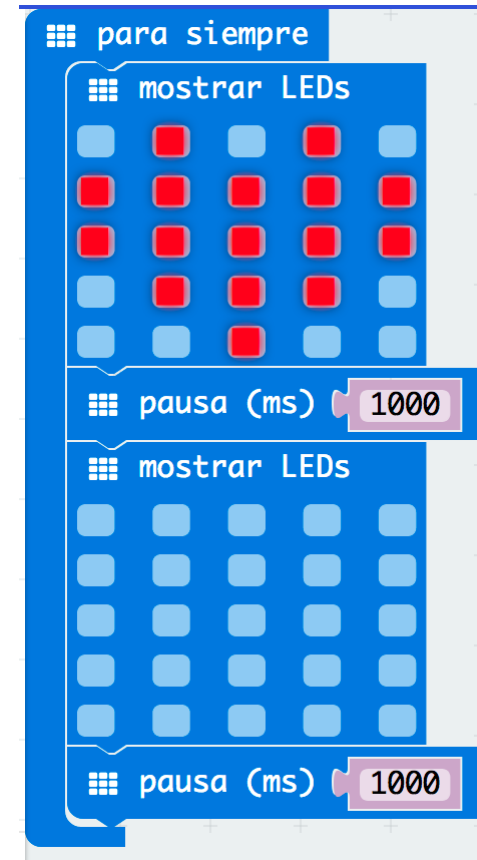


```
function showLeds(leds: string, interval?: number): void;
```

### – Ejemplo

#### • Blinky (modificado)

- Este ejemplo pinta un corazón y lo borra de forma repetitiva.
- El corazón permanece encendido 1400ms y apagado otros 400ms.





## Utilizando la matriz de LEDs

### – Mostrar numero

- **Visualiza un número o el valor de una variable**

- Si el número es mayor de 10 se desplaza.
- Al final del desplazamiento apaga el display un rato.
- Permanece en la función hasta que termina de visualizarse el número.

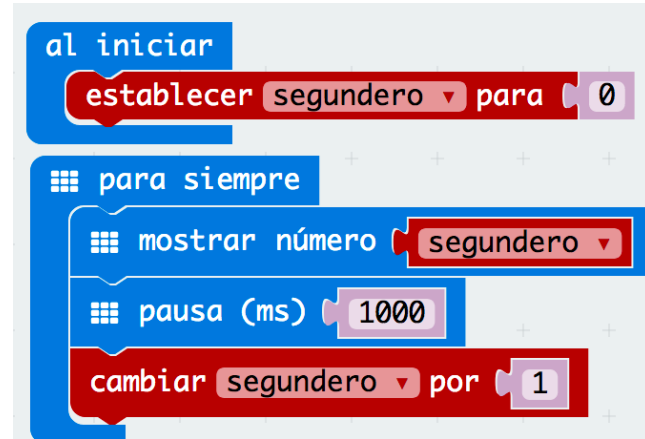


```
function showNumber(value: number, interval?: number): void;
```

### – Ejemplo

- **Cronometro (modificado)**

- El va aumentando el valor del número que se representa en los LEDs.
- Con un cronómetro puede comprobarse que la función “mostrar numero” tarda mucho en ejecutarse.



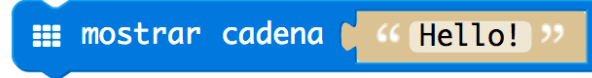


# Tarjeta micro:bit



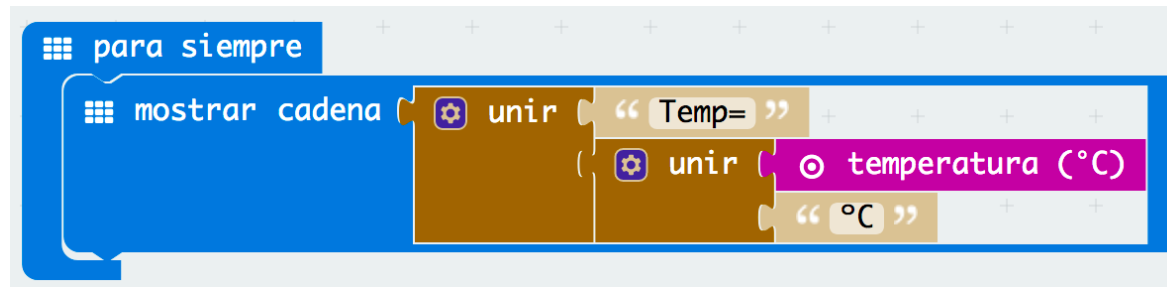
## Utilizando la matriz de LEDs

- **Mostrar cadena de texto**
  - Visualiza una cadena de texto o el contenido de una “variable de texto” o “String”



```
function showString(text: string, interval?: number): void;
```

- **Ejemplo**
  - **Termómetro**
    - Ejemplo que visualiza una cadena de caracteres que incluyen número





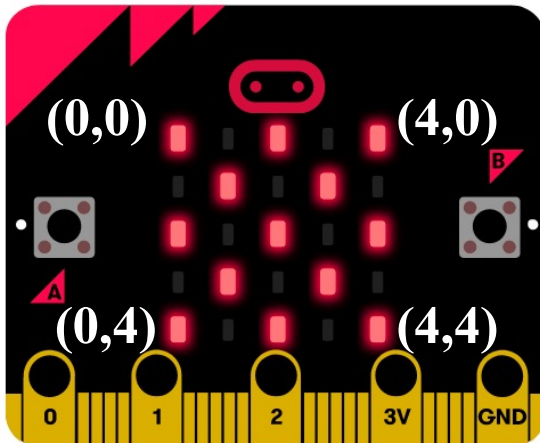


# Tarjeta micro:bit



## Activando y desactivando LEDs independientes

- Encender, Apagar o Cambiar LEDs independientes.



```
function plot(x: number, y: number): void;
```



```
function unplot(x: number, y: number): void;
```

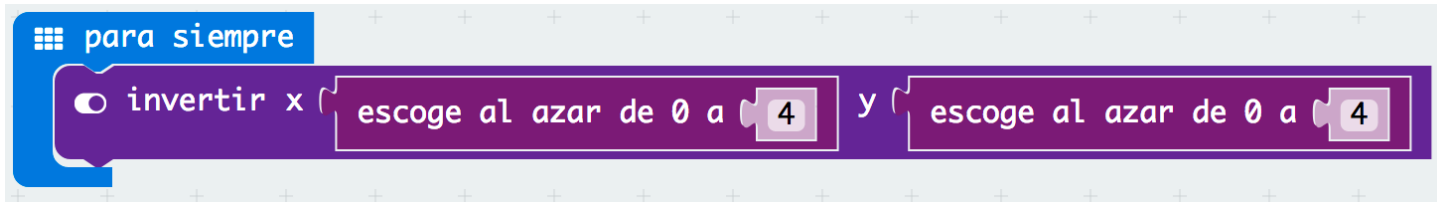


```
function toggle(x: number, y: number): void;
```

- Ejemplo

- Rando

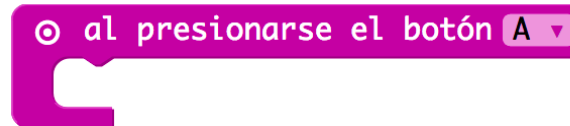
- Ejemplo que va cambiando el estado de los leds aleatoriamente
- Ejemplo de uso de "escoge al azar"





## Probando los pulsadores A y B

- “Al presionar el botón”
  - **Evento que se atiende al soltar el pulsador.**
    - El evento se activa al dejar de pulsar el pulsador cuando éste se pulsa durante un máximo de 1 segundo.
    - El evento A+B se activa al dejar de pulsar un pulsador cuando el otro pulsador estaba pulsado y la doble pulsación no ha durado más de 1,5 segundos.
  - **Identifica la pulsación de A, de B o de los dos a la vez (A+B)**



```
function onButtonPressed(button: Button, body: () => void): void;
```



# Tarjeta micro:bit



## Probando los pulsadores A y B

### – Ejemplo

#### • Pulsadores

- Ejemplo que permite entender cuándo se llama el evento asociado a la pulsación de pulsadores.
- Es importante ver lo que sucede al mantener los pulsadores apretados durante un tiempo superior a 2 segundos
  - » Ignora la pulsación

```
al iniciar
  establecer cuenta para 0
  mostrar número 0

al presionarse el botón A
  cambiar cuenta por 1
  mostrar número cuenta
```

### – Ejemplo

#### • Pulsaciones 2

- Ejemplo que visualiza cuenta el número de pulsaciones del Pulsador A
- Es necesario crear la variable “cuenta”
- La función “cambiar” es equivalente a sumar

```
al iniciar
  mostrar LEDs

al presionarse el botón A
  mostrar cadena " A "

al presionarse el botón B
  mostrar cadena " B "

al presionarse el botón A+B
  mostrar cadena " X "
```



# Tarjeta micro:bit



## Probando los pulsadores A y B

### – Ejemplo

#### • Pulsaciones 3

- Ejemplo que aumenta la variable cuenta mientras se pulsa el Pulsador A
- Cuando no se pulsa se visualiza.
- Nótese que mientras se visualiza “cuenta” no se hace caso a la pulsación.

```
al iniciar
  establecer cuenta para 0
  mostrar número cuenta

para siempre
  si botón A presionado
  entonces
    cambiar cuenta por 1
  si no
    mostrar número cuenta

  si botón B presionado
  entonces
    establecer cuenta para 0
    mostrar número cuenta
```

```
al iniciar
  establecer cuenta para 0
  mostrar número cuenta

para siempre
  si botón A presionado
  entonces
    cambiar cuenta por 1
  si no
    mostrar número cuenta

al presionarse el botón B
  establecer cuenta para 0
  mostrar número cuenta
```

### – Ejemplo

#### • Pulsaciones 4

- Ejemplo que hace lo mismo que el ejemplo anterior.
- Nótese la diferencia con el ejemplo anterior al pulsar el “Pulsador B”



# Tarjeta micro:bit

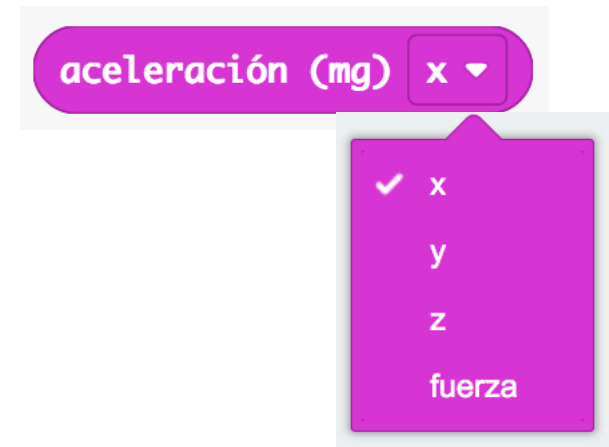


## Lectura de acelerómetros

- **Aceleración (mg)**
  - **Mide la fuerza detectada en los ejes X, Y y Z en mg (mili-g)**
    - Un mg es la milésima parte de la fuerza de la gravedad (g)
    - Realmente mide aproximadamente el tanto por mil del rango (ver “ajustar rango de acelerómetros”).
  - **También mide la fuerza conjunta detectada en las tres dimensiones**
  - **El valor medido va de -1023 a 1023**

- **Ejemplo:** Medidor de terremotos

```
para siempre
  mostrar número aceleración (mg) fuerza
  pausa (ms) 5000
```





## Lectura de acelerómetros

### – Ajustar rango de acelerómetros

- Permite ajustar el fondo de escala de medidas de aceleraciones que por defecto está a 1g
- No funciona en el simulador

ajustar rango de acelerómetro 1g ▾

- ✓ 1g
- 2g
- 4g
- 8g

### – Ejemplo

```
para siempre
  mostrar número aceleración (mg) x ▾
al iniciar
  ajustar rango de acelerómetro 4g ▾
```

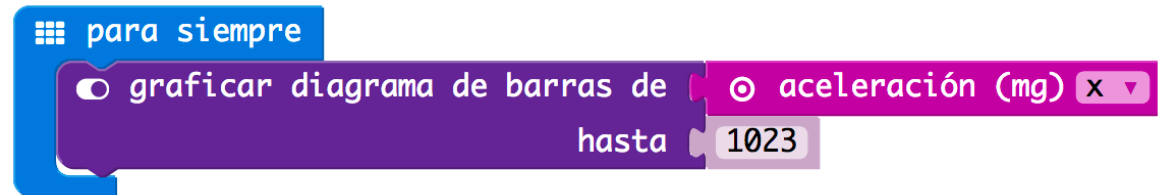


## Lectura de acelerómetros

### – Ejemplo

#### • Representación de la medida en barras

- Este ejemplo utiliza la función “graficar diagrama de barras” que representa en la matriz de LEDs una especie de diagrama de barras proporcional al valor de la variable introducida.
- El segundo parámetro es el valor máximo considerado para la representación.
- En el simulador se puede representar el valor de la variable.



### – Ejemplo

#### • Visualización de fuerza

- Este ejemplo representa cada 5 seg el valor de mg totales detectados teniendo en cuenta todos los eies.







# Tarjeta micro:bit



## – Ejemplo

- egg & spoon race

- El ejemplo visualiza un punto en la matriz de LEDs que se va moviendo al inclinar la tarjeta.
- Se puede modificar fácilmente para que vaya más rápido o sea más sensible.

```
al iniciar  
  establecer x para 2  
  establecer y para 2
```

```
para siempre  
  graficar x y  
  establecer accX para aceleración (mg) x  
  establecer accY para aceleración (mg) y  
  si  
    { accX < -150 y x > 0  
  entonces  
    cambiar x por -1  
  si no, si  
    { accX > 150 y x < 4  
  entonces  
    cambiar x por 1  
  si  
    { accY < -150 y y > 0  
  entonces  
    cambiar y por -1  
  si no, si  
    { accY > 150 y y < 4  
  entonces  
    cambiar y por 1  
  pausa (ms) 500  
  borrar la pantalla
```



# Tarjeta micro:bit

## Lectura de acelerómetros

### – Rotación (grados)

- Indica los grados de rotación de la tarjeta (-90 a 90)



```
function rotation(kind: Rotation): number;
```

```
para siempre
  establecer pitch para rotación (°) timbre
  establecer roll para rotación (°) girar
  si
    absoluto de pitch > 10 y absoluto de roll > 10
  entonces
    mostrar LEDs [
      [0, 1, 0, 1, 0],
      [1, 0, 0, 0, 1],
      [0, 0, 1, 0, 0],
      [1, 0, 0, 0, 1],
      [0, 1, 0, 1, 0]
    ]
  si no
    mostrar LEDs [
      [1, 0, 1, 0, 1],
      [0, 1, 1, 1, 0],
      [1, 1, 0, 1, 1],
      [0, 1, 1, 1, 0],
      [1, 0, 1, 0, 1]
    ]
```

### – Ejemplo

#### • Horizontal

- Este ejemplo pone una cara alegre si la tarjeta se mantiene horizontal y triste si se inclina un poco en cualquiera de los dos ejes (pitch y roll)



# Tarjeta micro:bit

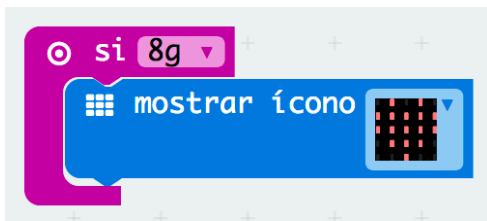


## Lectura de acelerómetros

- **Detección de gestos**
  - **Identifica algunos gestos ya especificados**



- **Ejercicio**
  - **Comprueba la diferencia en comportamiento entre estos dos programas**





## Propuesta de ejercicios básicos

- **Cambiando el estado de ánimo**
  - Realiza un programa que al pulsar el Pulsador A visualice una cara alegre y reproduzca un sonido agudo mientras que al pulsar el Pulsador B visualice una cara triste y suene un sonido grave.
- **Cara o Cruz**
  - Haz un programa que al pulsar cualquiera de los dos pulsadores aleatoriamente pinte una cara o una cruz.
- **Piedra – Papel - Tijera**
  - Haz un programa que al pulsar cualquiera de los dos pulsadores aleatoriamente pinte una de las tres opciones.
  - Modifícalo para que, en vez de aparecer al pulsar, lo haga al dejar de agitar.
- **Dado**
  - Haz un programa que al agitar la tarjeta no visualice nada y que al dejar de agitar represente la imagen de una cara de un dado elegida aleatoriamente.



## Propuesta de ejercicios básicos

millis (ms)

- **Medidor de reflejos**
  - En el menú Avanzado → Control se encuentra la variable millis que contiene el numero de milisegundos que han pasado desde que se ha encendido la tarjeta.
  - Haz un programa que, pasado un tiempo aleatorio, visualice algo en la pantalla y mida el tiempo que pasa desde que aparece hasta que se pulsa uno de los pulsadores. Tras pulsar deberá visualizarse el tiempo.
- **Contador de pasos**
  - Se propone hacer un programa que haga que la tarjeta cuente el número de pasos que va dando una persona. Para ello se debe conectar la tarjeta a una pila y colocarla en el tobillo con una goma.
  - Con cada paso que se dé se debe visualizar el número de pasos.
  - Se puede hacer al pulsar el Pulsador A, se visualice el número y que con el Pulsador B se ponga a cero.



Universidad  
de Alcalá

DEPARTAMENTO DE ELECTRÓNICA

# Sistema de desarrollo micro:bit

## Entradas y salidas al exterior

**D. Julio Pastor Mendoza** ([julio.pastor@uah.es](mailto:julio.pastor@uah.es))

Profesor del Departamento de Electrónica (UAH)

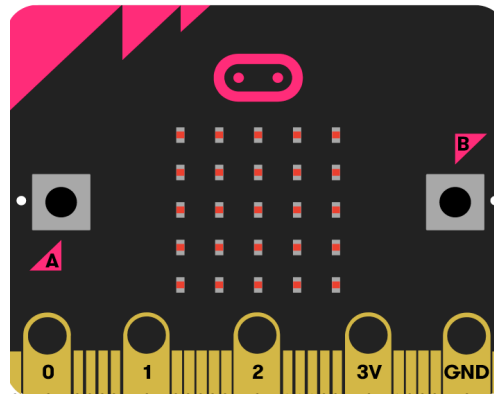


# Tarjeta micro:bit



## Conexiones externas de la tarjeta

- La tarjeta tiene **25 conexiones externas** que se denominan pines
  - 5 pines grandes
  - 20 pines pequeños.
- Están en un lateral de la tarjeta y la parte dorada son las conexiones.
- Hay 5 pines grandes que tienen un orificio para facilitar la conexión.
- Hay otros 20 pines más pequeños que se pueden usar cuando se conectan en una tarjeta de expansión.





# Tarjeta micro:bit



## Pines grandes

- Hay 5 pines grandes que tienen un orificio para facilitar la conexión.
  - Marcados como 0, 1, 2, 3V y GNG
  - Se pueden conectar fácilmente pinzas de cocodrilo o bananas de 4mm
  - Los pines 0, 1 y 2
    - Son pines multifunción de entrada y salida (GPIO – General Purpose Input Output)
      - » Entradas y salidas digitales
      - » Entradas analógicas
  - 3V
    - Es la salida de 3V si es alimentada con USB
    - Puede ser la entrada de 3V si no se alimenta por USB
  - GND
    - Es la masa o referencia de tensión.
- Si se toca el pin de GND con una mano, y un pin con la otra, se detecta la pulsación.





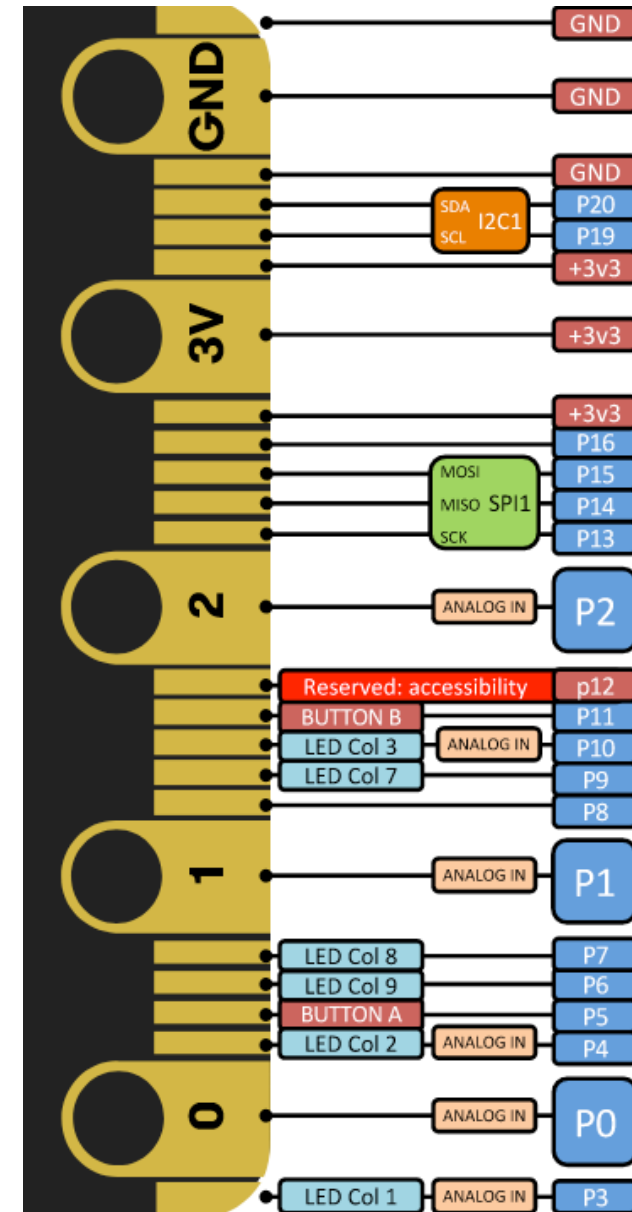


# Tarjeta micro:bit



## Pines pequeños

- Hay otros 20 pines más pequeños que se pueden usar cuando se conectan en una tarjeta de expansión.
- No están numerados en la tarjeta. Hay que verlo en el esquema.
- Algunos de los pines están conectados también a dispositivos internos de la tarjeta.
  - Pin 3, 4, 6, 7, 9, 10 están conectados a los LEDs. Pueden utilizarse como I/O digitales o ADC si LEDs apagados.



<https://makecode.microbit.org/device/pins>



# Tarjeta micro:bit



## Utilizando las entradas y salidas digitales

📌 lectura digital pin P0 ▾

```
function digitalReadPin(name: DigitalPin): number;
```

📌 escritura digital pin P0 ▾ a 📌 0

```
function digitalWritePin(name: DigitalPin, value: number): void;
```



# Tarjeta micro:bit



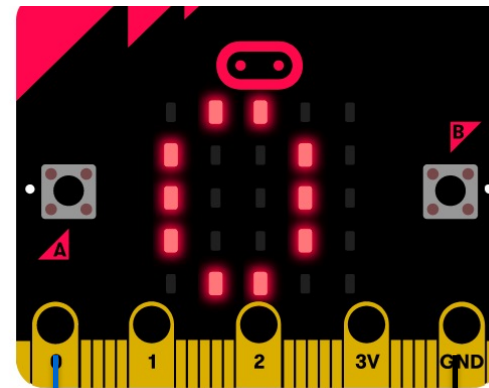
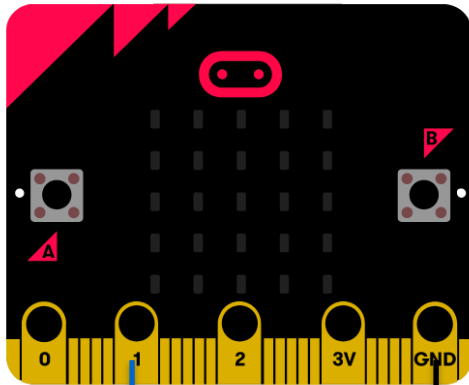
## Utilizando las entradas y salidas digitales

- Ejemplo de interconexión entre dos tarjetas

```
al presionarse el botón B
  escritura digital pin P1 a 1
  pausa (ms) 500
  escritura digital pin P1 a 0
```

```
para siempre
  si
    lectura digital pin P0 = 1
  entonces
    cambiar score por 1
    escritura digital pin P2 a 1
    mostrar número score
    pausa (ms) 1000
    escritura digital pin P2 a 0
```

```
al iniciar
  mostrar número score
```



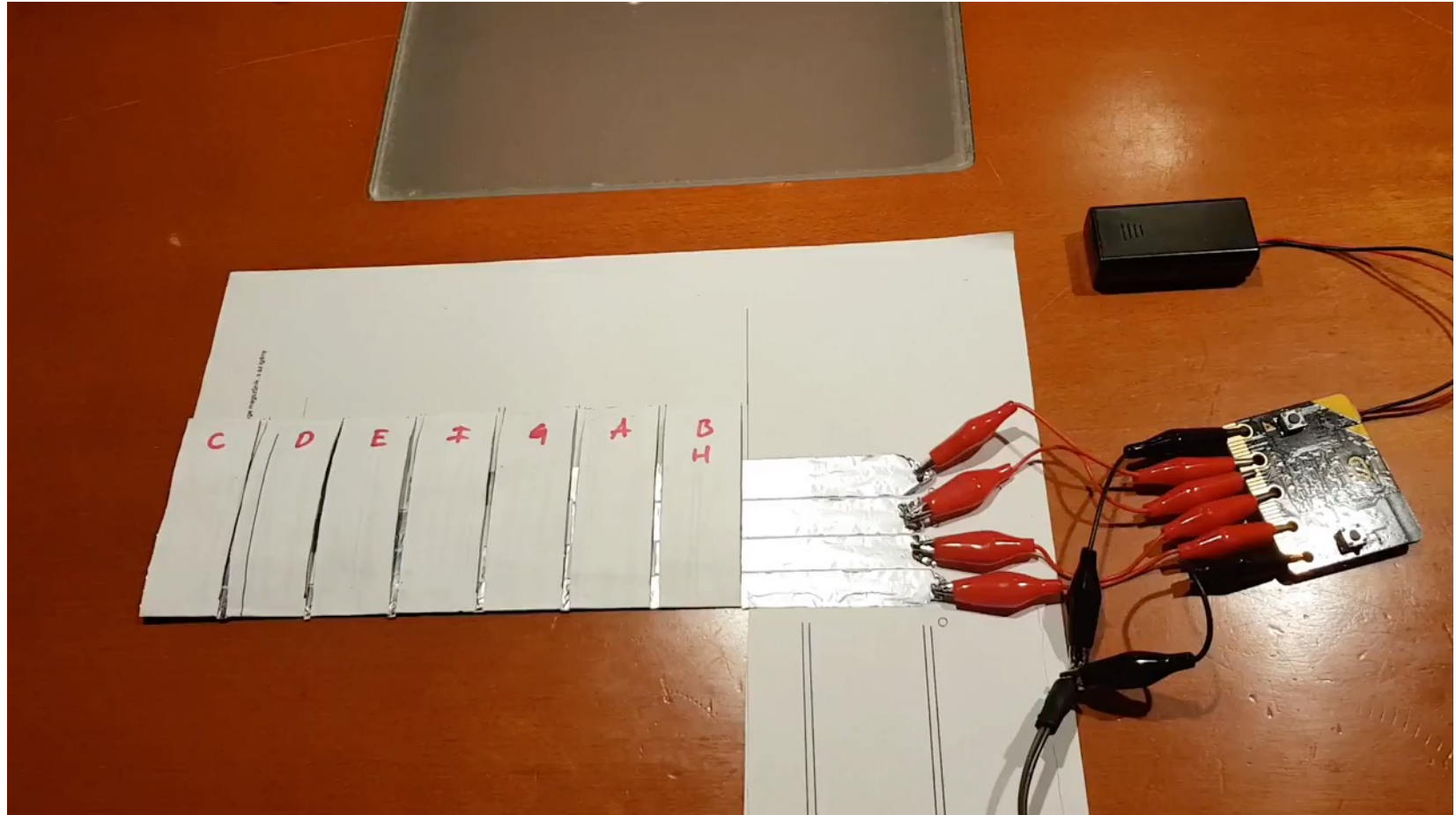


# Ejemplos de proyectos



S  
i  
s  
t  
e  
m  
a  
  
m  
i  
c  
r  
o  
  
b  
i  
t

## Paper Piano



<https://drive.google.com/file/d/1dgUUIEMWrP4NXYAxSSu5Rrd2ygb8OVcj/view>

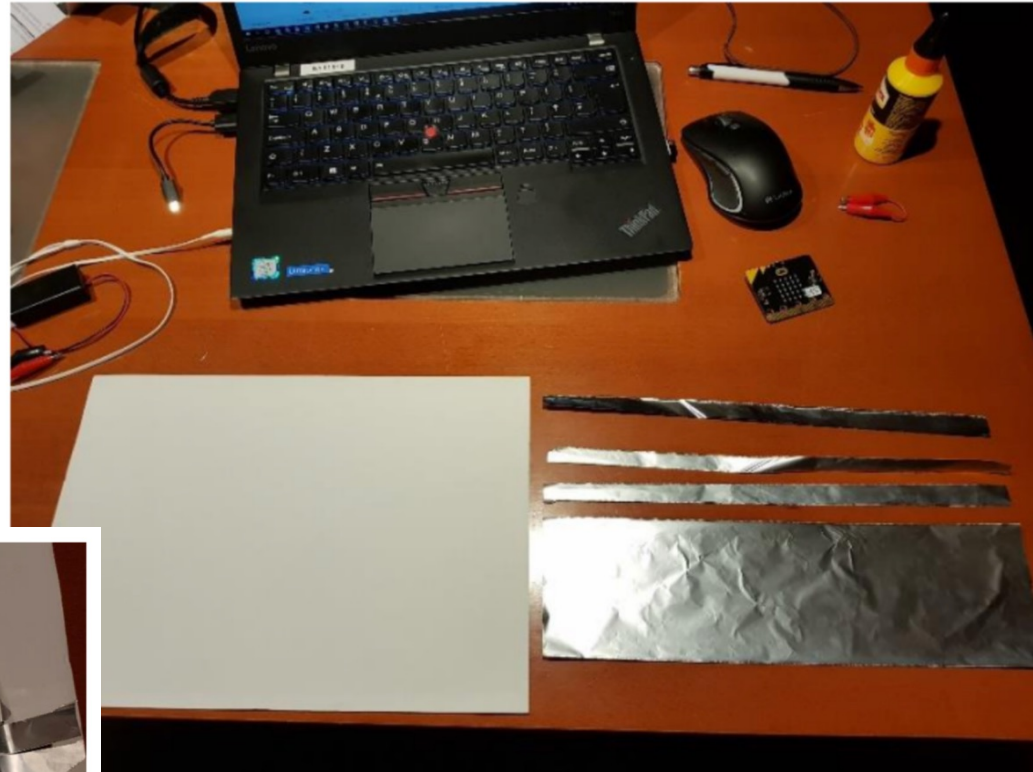


# Ejemplos de proyectos



## Paper Piano

Key	code	Note
-	000	silence
Key1	001	C
Key2	010	D
Key3	011	E
Key4	100	F
Key5	101	G
Key6	110	A
Key7	111	B





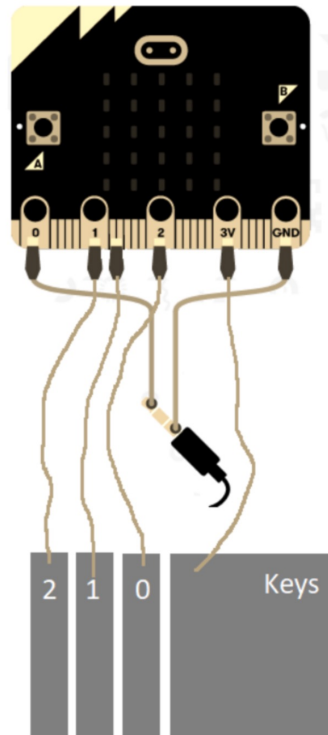


# Ejemplos de proyectos



## Paper Piano

Key	code	Note
-	000	silence
Key1	001	C
Key2	010	D
Key3	011	E
Key4	100	F
Key5	101	G
Key6	110	A
Key7	111	B





# Tarjeta micro:bit



## Utilizando las entradas analógicas

- Toma valores de 0 a 1023 (10 bits)
- Acepta tensiones de 0V a 3,3V

Ⓞ lectura analógica pin P0 ▾

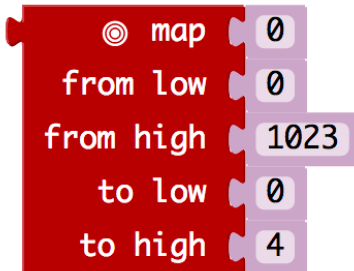
```
function analogReadPin(name: AnalogPin): number;
```

```
para siempre
  establecer value para Ⓞ lectura analógica pin P1 ▾
  mostrar número value ▾
```

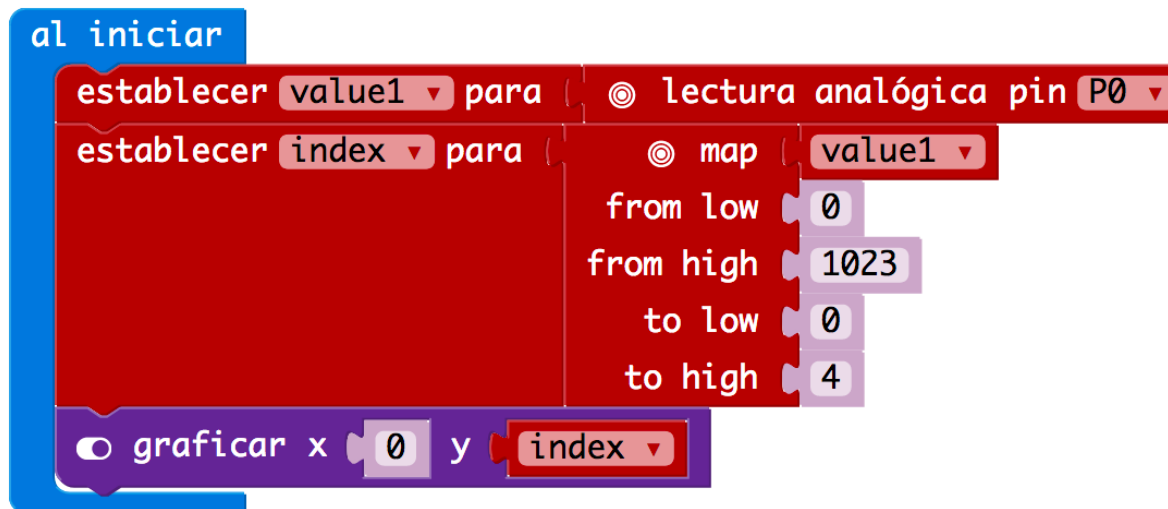


## Utilizando las entradas análogicas

- La función "map" permite adaptar un margen de valores en otro
- Implementa la ecuación de una recta (similar a una regla de tres)
- No limita los valores de entrada ni de salida



```
function map(value: number, fromLow: number, fromHigh: number, toLow: number, toHigh: number): number;
```







# Tarjeta micro:bit



## Utilizando las salidas analógicas

- Configura un pin como salida analógica.
- Acepta valores de 0 a 1023

© escritura analógica pin P0 a 1023

```
function analogWritePin(name: AnalogPin, value: number): void;
```

al iniciar

© escritura analógica pin P0 a 1023

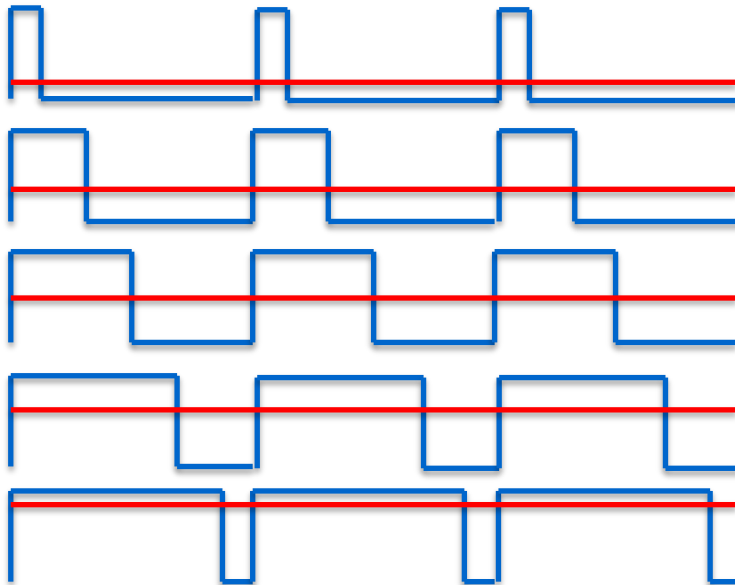


# Tarjeta micro:bit



## Utilizando las salidas analógicas

- La salida es una señal PWM (Pulse Wide Modulation) cuyo valor medio corresponde con el valor establecido.
- Se puede configurar el periodo de la señal de salida.
- Se pueden añadir componentes externos que obtenga de la señal el valor medio (Filtro paso bajo).



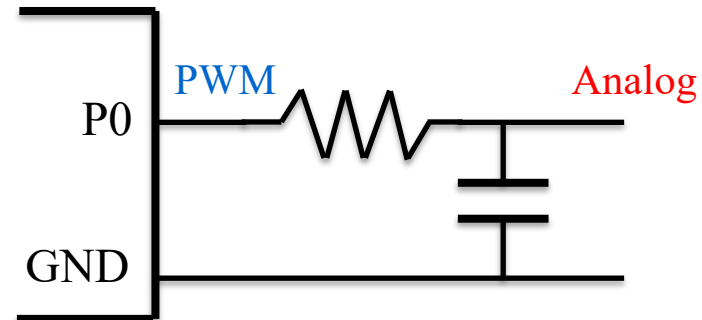
⊙ establecer período analógico pin P0 en (μs) 20000

```
function analogSetPeriod(name: AnalogPin, micros: number): void;
```

al iniciar

⊙ escritura analógica pin P0 a 512

⊙ establecer período analógico pin P0 en (μs) 20000





## Comunicación serie

- **La tarjeta permite mandar información al ordenador que puede servir para:**
  - Visualizar el valor de una variable.
  - Enviar información del ordenador a la tarjeta.
  - Comunicarse con una aplicación que se ejecute en el ordenador.
  - Registrar una serie de datos para su posterior tratamiento o visualización.
  - Etc.



# Tarjeta micro:bit



## Comunicación serie

- Se puede leer desde el propio editor con Chrome
- También se puede configurar en el ordenador un programa terminal que permita visualizar los mensajes y añadir nuevos.
  - **En Windows**
    - Si es inferior a Windows 10 es necesario instalar un driver.
      - » [https://docs.mbed.com/docs/mbed-os-handbook/en/latest/getting\\_started/what\\_need/](https://docs.mbed.com/docs/mbed-os-handbook/en/latest/getting_started/what_need/)
    - Programa Terminte ([https://www.compuphase.com/software\\_termite.htm](https://www.compuphase.com/software_termite.htm))
    - Programa PuTTY (<https://www.putty.org/>)
  - **En Mac OS**
    - Abrir el terminal (en utilidades)
    - `ls /dev/cu.*`. → Proporciona la lista de conexiones
    - `screen /dev/cu.usbmodem141222 115200`
    - Para salir: Ctrl a+ Ctrl d
    - Para almacenar en un fichero y parar el almacenamiento: Ctrl a + Ctrl H



# Tarjeta micro:bit



## Comunicación serie

- **Se puede:**
  - Enviar una cadena de caracteres (mensaje de texto)
  - Enviar una cadena de caracteres (mensaje de texto) con salto de línea
  - Enviar un número (en texto)
  - Enviar el valor de una variable ” variable = <valor>”
  - Recibir una línea de texto. Espera a recibir el final de la línea.
  - Leer la información recibida..
  - Habilitar un evento que se ejecute cuando se recibe un determinado carácter.
  - Redireccionar la comunicación por unos pines (puerto y velocidad)
  - Redireccionar al USB





# Tarjeta micro:bit



## Comunicación serie

- **Serial write string**
  - Envía una cadena de caracteres (mensaje de texto) sin salto de línea
- **Serial write line**
  - Envía una cadena de caracteres (mensaje de texto) con salto de línea
- **Serial write number**
  - Envía un número en forma de cadena de caracteres.
- **Serial write value**
  - Envía un texto seguido de un número separado por un =
  - El ejemplo envía la temperatura y el nivel de iluminación detectado.

```
forever
  serial write string " JUMBO "
  pause (ms) 1000
```

```
forever
  serial write line " BOFFO "
  pause (ms) 5000
```

```
forever
  serial write number 1234567890
  pause (ms) 5000
```

```
forever
  serial write value " temp " = temperature (°C)
  serial write value " light " = light level
  pause (ms) 10000
```



## Comunicación serie

### – Serial read string

- Según se va recibiendo texto se visualiza en el display.

```
forever
  show string ( serial read string )
```

### – Serial read line

- Espera hasta recibir un carácter de salto de línea por la comunicación serie devolviendo el texto recibido antes

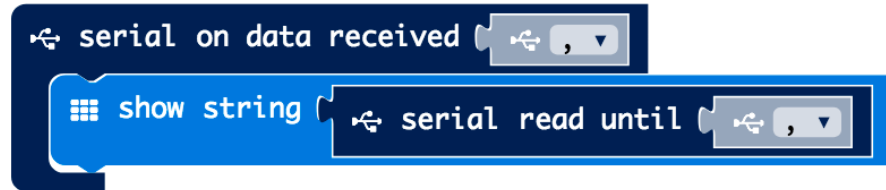
```
forever
  serial write line ( " What is your name? " )
  set answer to ( serial read line )
  serial write string ( " Hello, " )
  serial write line ( answer )
```



## Comunicación serie

### – Serial on data received

- Evento que se ejecuta cuando se recibe un determinado carácter por la comunicación serie.



### – Serial read until

- Espera hasta recibir un determinado carácter devolviendo el texto recibido hasta ese momento.





# Tarjeta micro:bit



## Comunicación serie

### – Serial redirect to USB

- Por defecto la comunicación serie está configurada para ser generada por el puerto USB.
- En caso de estar configurado para enviarlo por otros pines con esta función se vuelve al envío por USB.

serial redirect to USB

### – Serial redirect to

- En vez de relizar la transmisión por USB se puede configurar para utilizar dos pines de la tarjeta, uno de transmisión o salida (TX) y otro de recepción o entrada (RX)
- La velocidad de comunicación se puede configurar entre diferentes valores estándar: 9600, 19200, 38400, 57600 ó 115200

on button A pressed

serial  
redirect to  
TX P1  
RX P2  
at baud rate 9600



# Tarjeta micro:bit



## Comunicación serie

### – Visualización de datos utilizando “Serial Plotter” de Arduino

- Instala la última versión del IDE de Arduino
  - <https://www.arduino.cc/en/Main/Software>
- Carga el programa en la tarjeta micro:bit
- Ejecuta el Arduino IDE
- En Herramientas → Puerto, selecciona el puerto asociado al micro:bit
- En Herramientas → Serial Monitor (eligiendo 115200 de velocidad) deberían verse los valores separados por comas.
- En Herramientas → Serial Plotter deberían verse las gráficas de datos.

```
on start
  set item to 0

forever
  if (item < 100)
    then change item by 1
  else set item to 0

  serial write number (analog read pin P1)
  serial write string (",")
  serial write number (item)
  serial write line (" ")
```



Universidad  
de Alcalá

DEPARTAMENTO DE ELECTRÓNICA

# Sistema de desarrollo micro:bit

## Conexión de sensores

**D. Julio Pastor Mendoza** ([julio.pastor@uah.es](mailto:julio.pastor@uah.es))

Profesor del Departamento de Electrónica (UAH)



## Wiki Keystudio

- [https://wiki.keystudio.com/KS0361\(KS0365\)\\_keystudio\\_37\\_in\\_1\\_Starter\\_Kit\\_for\\_BBC\\_micro:bit](https://wiki.keystudio.com/KS0361(KS0365)_keystudio_37_in_1_Starter_Kit_for_BBC_micro:bit)





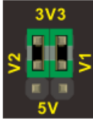
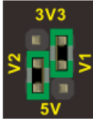
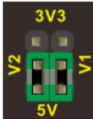
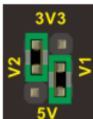
# Mico:bit



S  
i  
s  
t  
e  
m  
a  
  
m  
i  
c  
r  
o  
  
b  
i  
t

## Wiki Keystudio



Picture	V1 Interface voltage/V	V2 Interface voltage/V
	3.3	3.3
	3.3	5
	5	5
	5	3.3

Ejemplo	Entrada Digital	Salida Digital	Entrada Analógica	Salida Analógica	Comunicaciones	Otros
<a href="#">7.2 Project 2: LED Blink</a>					X	
<a href="#">7.3 Project 3: Breath</a>				X		
<a href="#">7.4 Project 4: Blink and Breath</a>		X		X		
<a href="#">7.5 Project 5: Make a Sound</a>		X				
<a href="#">7.6 Project 6: Play Music</a>		X				X
<a href="#">7.7 Project 7: Changing Color</a>				X		
<a href="#">7.8 Project 8: Button Control</a>	X					
<a href="#">7.9 Project 9: Tilt Control</a>	X					
<a href="#">7.10 Project 10: Light Interrupter</a>	X					
<a href="#">7.11 Project 11: Capacitive Touch</a>	X					
<a href="#">7.12 Project 12: Traffic Light</a>		X				
<a href="#">7.13 Project 13: Magnetic Detection</a>	X					
<a href="#">7.14 Project 14: Follow Black Line</a>	X					
<a href="#">7.15 Project 15: Obstacle Avoidance</a>	X					
<a href="#">7.16 Project 16: Someone Comes</a>	X					
<a href="#">7.17 Project 17: Fire Alarm</a>	X	X				
<a href="#">7.18 Project 18: To The Top</a>	X	X				
<a href="#">7.19 Project 19: Magnetic Switch</a>	X					
<a href="#">7.20 Project 20: Relay</a>		X				
<a href="#">7.21 Project 21: Ultrasonic Ranging</a>	X				X	X
<a href="#">7.22 Project 22: Light Brightness</a>			X		X	
<a href="#">7.23 Project 23: Analog Temperature</a>			X		X	
<a href="#">7.24 Project 24: Hear Footstep</a>			X		X	
<a href="#">7.25 Project 25: Rotary Potentiometer</a>			X		X	
<a href="#">7.26 Project 26: Alcohol Content in the Air</a>			X		X	
<a href="#">7.27 Project 27: Flammable Gas in the Air</a>			X		X	
<a href="#">7.28 Project 28: Ambient Temperature Detection</a>			X		X	
<a href="#">7.29 Project 29: Ambient Light</a>			X		X	
<a href="#">7.30 Project 30: Automatic Water Your Plant</a>			X		X	
<a href="#">7.31 Project 31: Water Level Alarm</a>		X	X		X	
<a href="#">7.32 Project 32: Ultraviolet Detection</a>			X		X	
<a href="#">7.33 Project 33: Vapor in the Air</a>			X		X	
<a href="#">7.34 Project 34: Pressure Detection</a>			X		X	
<a href="#">7.35 Project 35: Light Controlled By Vibration</a>	X	X				
<a href="#">7.36 Project 36: Joystick</a>			X		X	
<a href="#">7.37 Project 37: Micro Servo</a>						X
<a href="#">7.38 Project 38: Add LCD</a>						X