

# TRADUCTOR DE MORSE USANDO MICROBIT

Marcos Rico Guerra

## Introducción

Como bien indica el título del documento, este trabajo ha consistido en la implementación de un traductor de Morse al alfabeto latino(El que usamos en España).

Esto ha sido realizado para una placa BBC micro:bit, un pequeño ordenador que puede alojar programas realizados en otra computadora para realizar ciertas funciones.

Gracias a las posibilidades que esta placa nos ofrece, como botones o ciertos leds que pueden ser usados para mostrar caracteres, realizar ciertas tareas como la realizada en esta práctica resulta una tarea medianamente sencilla de hacer.

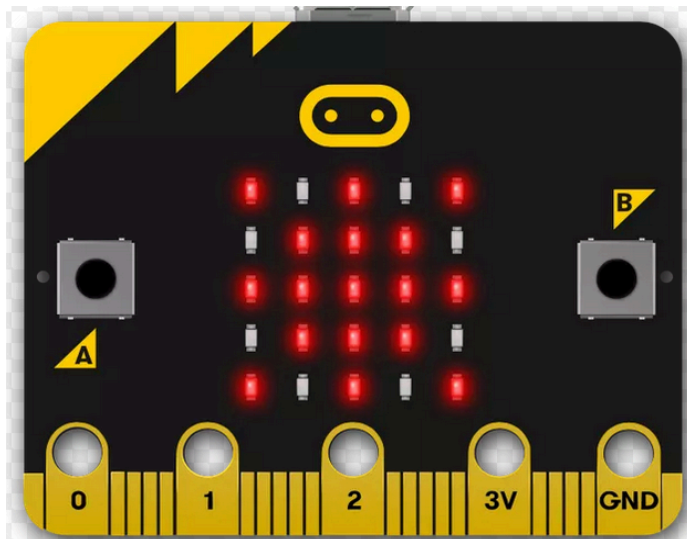


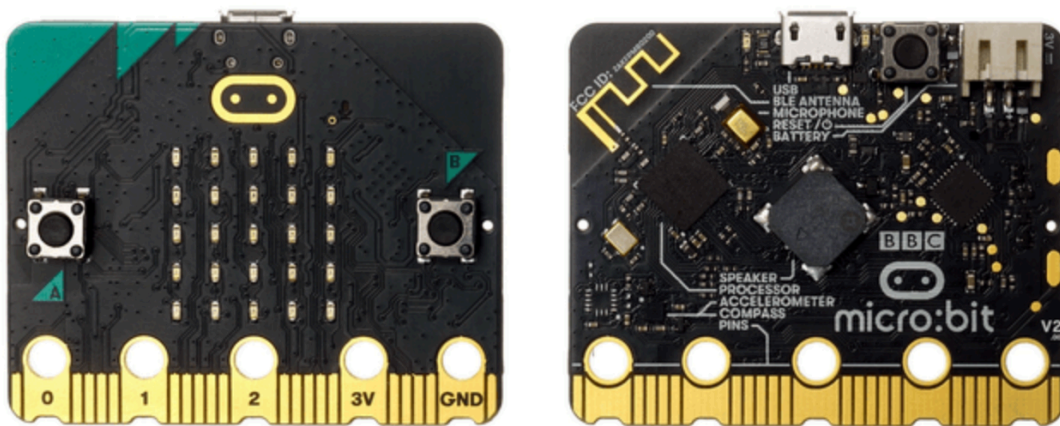
Imagen de una placa micro:bit con 2 botones y pantalla led 5x5

# MicroBit

A continuación, hablaremos un poco sobre la placa Micro:bit.

Técnicamente hablando, micro:bit es un sistema embebido de hardware libre que se basa en ARM. Su origen se debe a la BBC (Servicio público británico de radio y televisión), cuyo fin era la educación informática del Reino Unido.

Actualmente existen dos versiones de esta, v1 y v2; siendo la segunda una versión mejorada en cuanto a las prestaciones hardware que contiene (Por ejemplo, integra un sensor de temperatura o un micrófono), pero la práctica ha sido realizada en la v1.

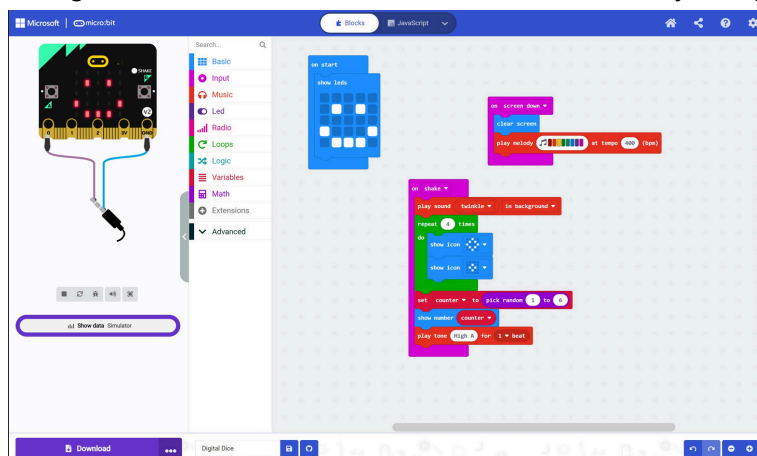


Placa Micro:bit v2 por ambos lados

En cuanto al software, existen dos editores de código oficiales para esta placa: Microsoft Makecode y MicroPython.

El primero ha sido el usado en el proyecto, y permite programación por bloques o usando JavaScript.

El segundo, como su nombre indica, hace uso de Python para la programación.



Ejemplo de programa en bloques usando Microsoft Makecode

Cabe destacar como Makecode, además de tener bloques para usar expresiones aritméticas, lógicas, bucles, etc. Tiene dos bloques especiales llamados “on start”, que se ejecuta siempre que se inicia la placa, y otro llamado “forever”, que se ejecuta en bucle sin parar.

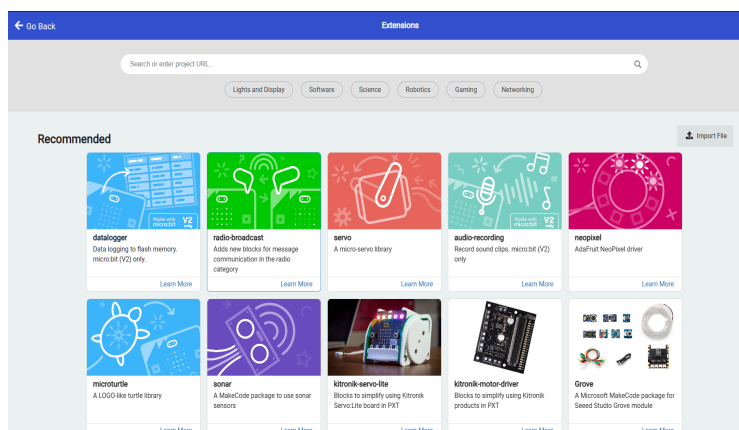
Además, esta incluye un simulador donde podremos visualizar el código que acabamos de implementar en una placa microbit virtual.

## Traductor de Morse

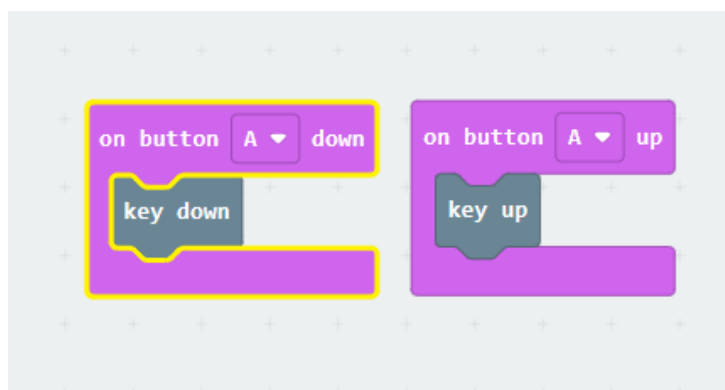
Como ya se ha mencionado, en esta práctica se ha implementado un programa usando Microsoft Makecode.

Este programa integra un traductor de código morse en la placa.

Para ello, se usa el botón izquierdo etiquetado como “A” para el receptor del código morse. De base, la placa solo reconoce la pulsación del botón(Pero no el momento en que se suelta este), por lo que fue necesario buscar una forma de reconocer los . y los -  
Tras investigar, encontré una biblioteca que era capaz de reconocer cuando el botón es soltado, por lo que se simplificó el programa.



Buscador de extensiones de Makecode



## Funciones para detectar pulsado y soltado de botones

Además, entre las extensiones encontré una para implementar programas que empleen código morse.

Esta biblioteca añade varias cosas, destacando los bloques “key down” y “key up” y el bloque “on code (sequence) selected”

## Extensión Morse Code

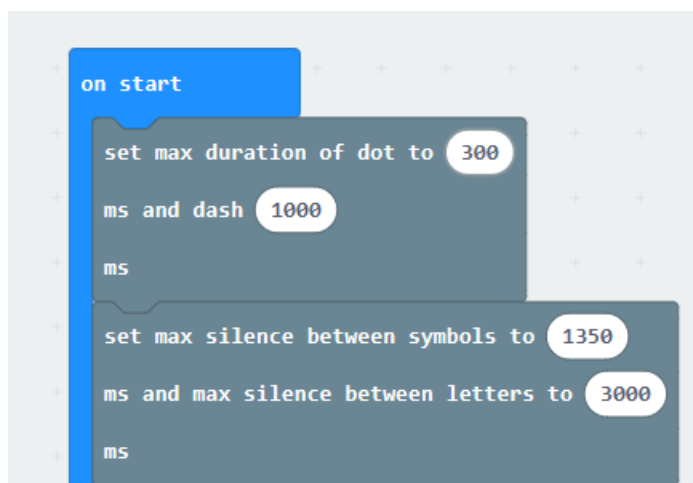
Hablaremos brevemente sobre esta extensión y como funciona.

Esta extensión tiene como elemento principal los bloques “key down” y “key up”. Estos representan la pulsación (key down) y soltar(key up) un botón que transmita código morse.

La extensión registra el tiempo que pasa entre un key down y un key up sucesivo. Si es menor que “max duration of dot”, registra un punto; si está entre “max duration of dot” y “max duration of slash” registra un guión; y en otro caso se considera un elemento inválido y elimina los puntos y guiones registrados anteriormente.

Si tras un key up no se vuelve a pulsar key down en un intervalo de tiempo mayor a “max time between symbols” y menor a “max time between letters”, se guarda la secuencia actual de puntos y guiones en una variable llamada sequence. Si esta secuencia corresponde a una letra, se asigna esta letra en otra variable code. En caso de que sea una secuencia incorrecta, se guarda en code el valor “?”.

Si dejamos pasar aún más el tiempo entre pulsaciones, se registra el fin de una palabra, dándole a code el valor “ ”, representando el espacio entre palabras.



Registro de los tiempos para puntos, guiones, letras y palabras en código

Cada vez que se guarda un valor en code (Ya sea por registrar una letra o una palabra) el programa llama al bloque “on code (sequence) selected”, donde podremos incluir el código que queramos.

Existen otras funciones que pueden ser de gran utilidad, pero en este caso solo usamos otra llamada “on symbol newsymbol entered” que es llamada cada vez que introducimos un punto o un guión(o un símbolo incorrecto).

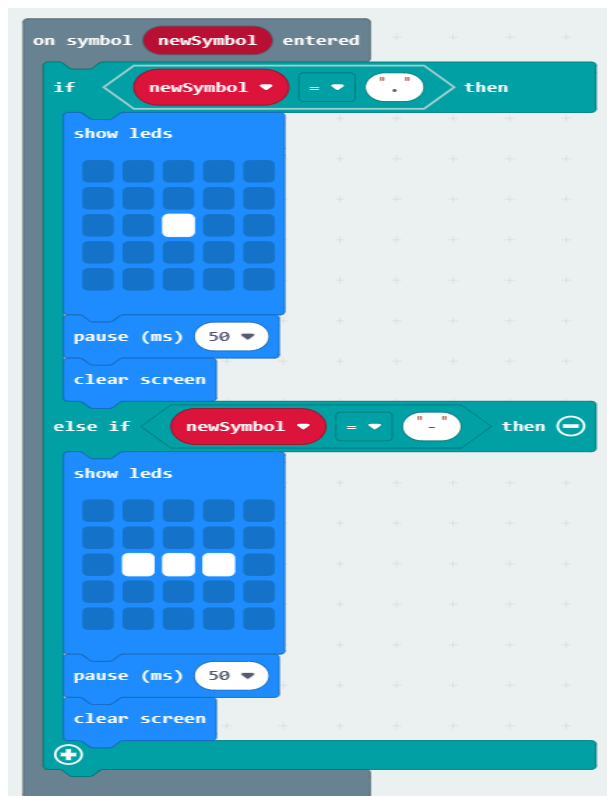
## Código

Una vez vistas las funciones de la biblioteca, veremos como las hemos usado para implementar nuestro programa.

Lo primero es fijar los tiempos máximos para registrar puntos, guiones, letras o palabras, cosa que vimos antes.

Además, es necesario inicializar una nueva variable a la que llamo palabra. En esta, iremos guardando la palabra que se va generando, ya que la biblioteca no lo hace por sí misma.

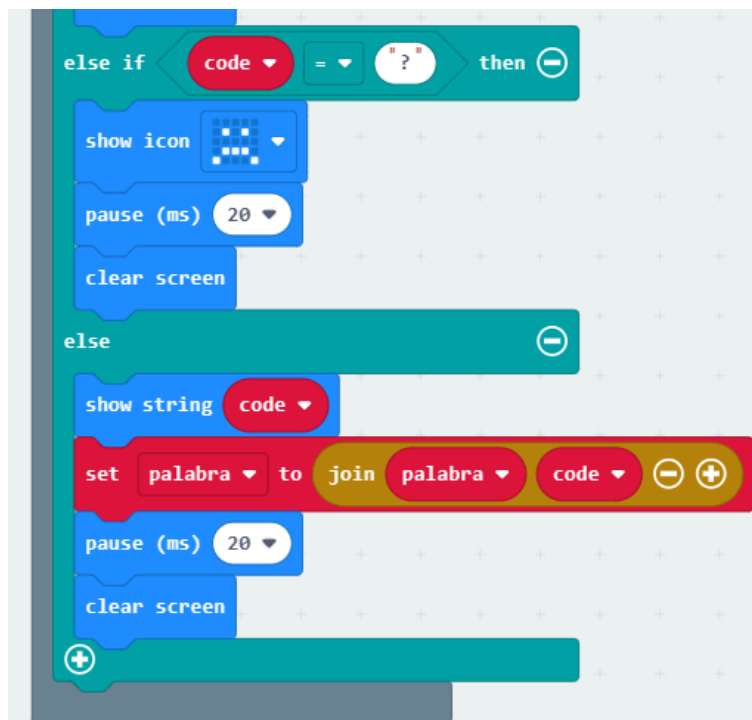
Tras esto, vamos guardando las pulsaciones según si se pulsa el botón A o si se suelta. Cuando leemos un punto o un guión, este se muestra en los leds brevemente, para verificar lo que hemos pulsado



## Código que muestra el símbolo pulsado

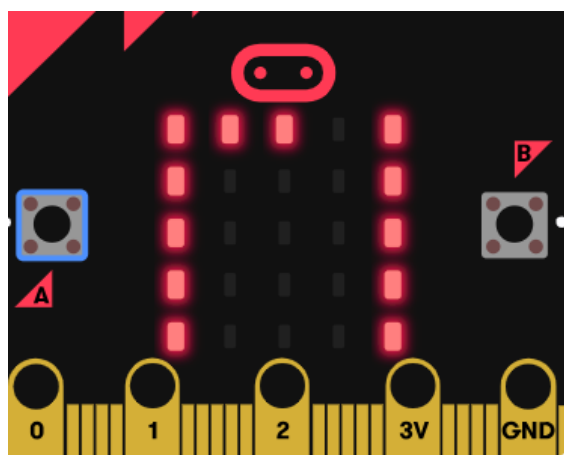
Como vemos, se hace uso de un bloque llamado show leds y una matriz 5x5. Esto enciende los leds que indicamos de color blanco(En la placa se ven de color rojo).

Cuando se registra una letra, esta es mostrada brevemente y se añade a la palabra. Si la letra no existe, se muestra una cara triste y se ignora



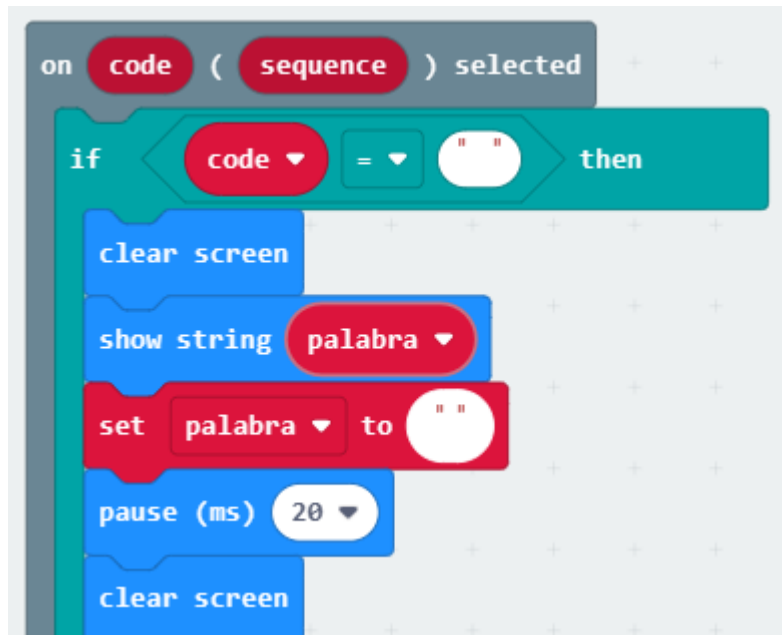
Código que se ejecuta tras recibir una letra(o carácter incorrecto)

Se emplea la función show string, que es capaz de mostrar letras o frases enteras en los leds del microbit. En caso de que sea algo demasiado largo para incluirlo en los leds, se hace una especie de scroll de izquierda a derecha por todas los caracteres



Aunque no se distinga del todo bien, en la imagen la placa está mostrando la palabra "TE". En este caso, vemos la parte derecha de la T y la parte izquierda de la E

Cuando registramos una palabra, simplemente mostramos la palabra completa en la placa, y le damos a la variable "palabra" el valor vacío.



Código ejecutado tras recibir una palabra entera

## Conclusión

Tras la realización de la práctica, puedo concluir que trabajar con una placa microbit es bastante cómodo, tanto por su hardware que hace todo más cómodo y evita tener que ampliar la placa con otro tipo de hardware en ciertas ocasiones; como por su software, debido a la gran cantidad de funciones existentes que facilitan la programación, además de todo el número de extensiones que podemos encontrar.

Al final, el trabajo resultó más un trabajo de investigación sobre la propia placa para conocer como usar la pantalla led, buscar las bibliotecas necesarias para facilitar el trabajo e investigar dichas bibliotecas para conocer cómo funcionan por dentro y como usarlas.

En cuanto al proyecto, encuentro que sería difícil encontrar mejoras, ya que realmente funciona perfectamente como un traductor morse. Lo único que podría mejorarse es poder guardar frases enteras y mostrar toda la frase cuando se desee, en vez de poder mostrar sólo palabra a palabra.