

Música predefinida o crea tu música

SALIDAS DE AUDIO

La placa Microbit v2 tiene un altavoz incorporado que se puede anular o activar con la instrucción **speaker.on()** o **speaker.off()**

La salida de audio también sale por el **pin0**, de tal manera que si conectamos un altavoz o headphone, y tenemos speaker.on() se oirá por los sitios, si lo tienes en off sólo por el buzzer/headphones exterior:

[435692553_1.png](#)
[headphones_1.png](#)

Fuente <https://www.teachwithict.com/microbit-music-python.html>

BBC micro:bit "Magic 8-Ball" lesson is licenced under a [Creative Commons Attribution 4.0 International License](#).

Ojo, hay que conectar un **buzzer pasivo**, es decir, que reproduce la seña analógica en sonido (o sea un altavoz normal y corriente)

si quieres conectar un **buzzer activo**, reproduce un tono (predefinido) al suministrarle un 1, el siguiente código sonaría una alarma:

```
from microbit import *
while True:
    pin0.write_digital(1)
    sleep(500)
    pin0.write_digital(0)
    sleep(500)
```

En los siguientes ejemplo usaremos siempre **buzzer pasivo**. Si no te queda claro lo que es un buzzer activo y un pasivo, mira [esta página](#)

CREA TUS EFECTOS

Puedes crear los efectos utilizando rangos de frecuencias, aquí en el ejemplo la función pitch reproduce durante 6mseg las frecuencias de medio 880 , aguda 1760 y grave 16 y luego lo mismo pero en orden decreciente, y así sucesivamente para dar el efecto de sirena.

```
import music
from microbit import *
display.show(Image.GHOST)
while True:
    for freq in range(880, 1760, 16):
        music.pitch(freq, 6)
    for freq in range(1760, 880, -16):
        music.pitch(freq, 6)
```

Extraído de <https://microbit-micropython.readthedocs.io/en/v2-docs/tutorials/music.html#sound-effects>

<https://www.youtube.com/embed/tSkmgffox2A>

MUSICA PREDEFINIDA

En Reference- Sound tienes muchos tonos predefinidos para experimentar:

[2024-09-24 13_13_12-micro_bit Python Editor.png](#)

Si tienes la microbit v2 tienes otros en "**Expressive sounds**" como `audio.play(Sound.TWINKLE)`

También puede hablar, arrastra el código que tienes en Reference-Sound- Speech, pero no lo intentes en español, no se entiende nada

```
import speech
from microbit import *
```

```
display.show(Image.FABULOUS)
speaker.on()
set_volume(255)
speech.say('Hello, How are you? Do you sign up in online course in CATEDU.ES ?')
audio.play(Sound.TWINKLE)
```

<https://www.youtube.com/embed/EcV3aeYo6Vs>

CONSTRUYENDO TU MÚSICA

Tienes que añadir la librería **music** y componer las notas según la notación americana :

2024-09-24 12_45_51-Making music with the micro_bit - MicroPython version - teachComputing.png

Fuente <https://www.teachwithict.com/microbit-music-python.html>

BBC micro:bit "Magic 8-Ball" lesson is licenced under a [Creative Commons Attribution 4.0](#)

[International License](#).

La duración (si no se pone, sigue con la duración anterior)

2024-09-24 12_46_47-Making music with the micro_bit - MicroPython version - teachComputing.png

Fuente <https://www.teachwithict.com/microbit-music-python.html>

BBC micro:bit "Magic 8-Ball" lesson is licenced under a [Creative Commons Attribution 4.0](#)

[International License](#).

Se puede poner incluso sostenidos, por ejemplo C#4:4 o f#5:4

Si quieres poner descansos es con la letra r seguido de su duración por ejemplo r:4 r:2

Un ejemplo

Tono Nokia (arriba la duración)

[image.png](#)

El código sería:

```
from microbit import *  
import music  
  
tune = ["e5:2", "d5", "f#4:4", "g#4", "c#5:2", "b4", "d4:4", "e4", "b4:2", "a4", "c#4:4", "e4", "a4:12"]  
  
music.play(tune)
```

Otro ejemplo, el código lo tienes en <https://microbit-micropython.readthedocs.io/en/v2-docs/music.html>

<https://www.youtube.com/embed/0u3hjXXEZYg>

Revision #7

Created 2024-09-24 12:33:13 CEST by Javier Quintana

Updated 2024-09-26 14:39:46 CEST by Javier Quintana