

MultiWingSpan

[Home](#) [Programming](#) [Web Design](#) [Computer Science](#) [Twisting Puzzles](#) [Arduino](#) [BBC micro:bit](#)

BBC micro:bit Beeping The Horn

Introduction

There isn't much to this. There's a buzzer. It will play a high pitched tone whilst pin 14 is driven high. That means that you don't need to import the music library to use it but that you can't make your robot play a tune. Sad face. It's still has its uses though.

Programming

Here is a function that can be used to make the robot do a double beep over half a second.

```
def Beep():
    pin14.write_digital(1)
    sleep(200)
    pin14.write_digital(0)
    sleep(50)
    pin14.write_digital(1)
    sleep(200)
    pin14.write_digital(0)
    sleep(50)
```

Another approach to using the buzzer would be in a **while True:** loop that contained a sleep statement. You can have an IF statement that checks sensors for a condition that needs a beep response and, if you find it, write a 1 to pin 14. If the condition is not met, a 0 is written. If your beeping is more of a one-off event or needs to go on longer, you can use a variable and add a set amount to it each loop. When it is equal to the amount you want, turn the beep off.

Challenges

1. The buzzer is pretty simple to use and integrate into other projects. Add it to any of the projects you have done so far.
2. You can use the buzzer as a time indicator if you write your program around a loop with a sleep statement. Work out how to get the buzzer to do a short beep every second.
3. Bump = buzz. Naturally.
4. Morse code buzzing.
5. Timely beeping synchronised with movements can be used to debug your programs as well as being a nice effect that you can add to a dancing robot program.

BBC Microbit

+ [Block Editor - The Basics](#)

+ [Block Editor - Components](#)

+ [Kodu - micro:bit Worlds](#)

+ [JavaScript Blocks](#)

+ [JavaScript Blocks - Exercises](#)

- [Blocks - Bit:Bot](#)

✖ [Meet The Bit:Bot](#)

✖ [Driving The Motors](#)

✖ [Beeping The Horn](#)

✖ [Lighting The Way](#)

✖ [Following Lines](#)

✖ [Sensing Light](#)

✖ [Radio Control](#)

+ [Blocks - Bit:Commander](#)

+ [MicroPython - Starting Off](#)

+ [MicroPython - Examples](#)

+ [MicroPython - Components](#)

+ [MicroPython - Breakout Boards](#)

+ [MicroPython - Exercises](#)

+ [MicroPython - Pi Accessories](#)

- [MicroPython - Bit:Bot](#)

✖ [Meet The Bit:Bot](#)

✖ [Driving The Motors](#)

✖ [Beeping The Horn](#)

✖ [Lighting The Way](#)

✖ [Following Lines](#)

✖ [Sensing Light](#)

✖ [Radio Control](#)

+ [MicroPython - Bit:Commander](#)

+ [MicroPython - Projects](#)

+ [MicroPython - Visual Basic](#)

+ [Other - Odds & Ends](#)

