

MultiWingSpan

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

BBC micro:bit Potentiometer (MicroPython)

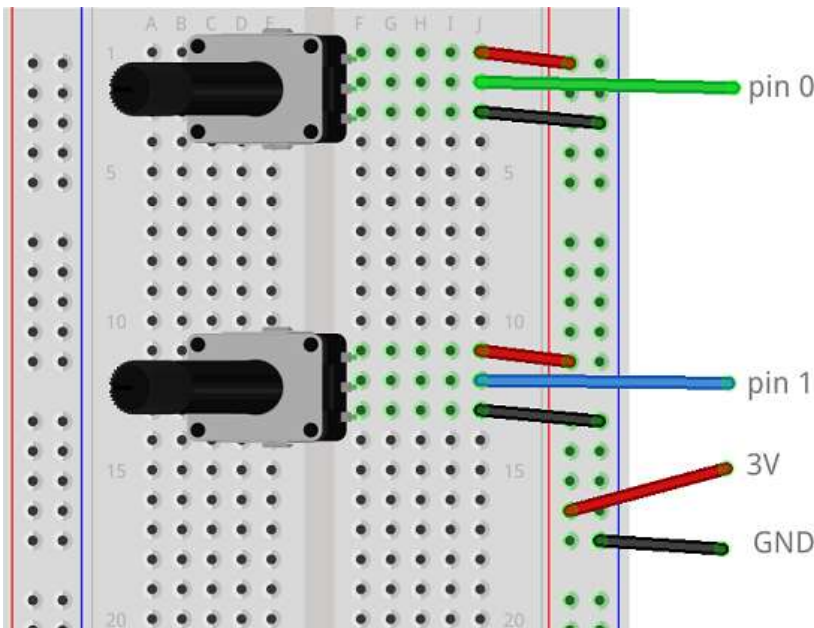
Introduction

The more powerful range of statements that you have in MicroPython makes it much easier to use simple components like potentiometers.

On this page, you will make a circuit with two potentiometers. We will take the readings from the potentiometers and convert them into numbers that represent coordinates on the LED matrix. You will have to do the rest of the work.

Circuit

With two potentiometers, we really need to use a breadboard. The alligator clips are fine for a single component but it becomes fiddly with any more than that.



The 3V and GND connections can be swapped around if you want the readings to be reversed.

Programming

This code is just the start. It is a simple way to test both the connections and the values you are getting. The program consists of an infinite while loop. The next two lines read the wiper pins of the potentiometers and store the values in variables **a** and **b**. The next two lines round the lines to make a number from 0 to 4. After a brief pause, the values are printed. If you press the REPL button, you will see the readings scrolling down the screen. Turn your knobs and you should be able to make them change from 0 to 4.

```
from microbit import *

while True:
    a = pin0.read_analog()
    b = pin1.read_analog()
    a = int(4 * (a / 1023))
    b = int(4 * (b / 1023))
    sleep(50)
    print("x:" + str(a) + " y:" + str(b))
```

Challenges

1. Obviously you need to map this onto some sort of sprite and use the potentiometers as inputs for a game.
2. Maybe you are more interested in using the potentiometer value for something else. Perhaps you could use it to select a frequency to send to a buzzer or some headphones you have hacked.
3. You could use the potentiometer readings to make some sort of display on the LED matrix. You could make something approximating a volume up/down image (a triangle) that fills as you turn the potentiometer.
4. A potentiometer could also be wired up in series with any speaker you are using to make a volume adjuster.

BBC Microbit

- + [Block Editor - The Basics](#)
- + [Block Editor - Components](#)
- + [Kodu - micro:bit Worlds](#)
- + [JavaScript Blocks](#)
- + [JavaScript Blocks - Exercises](#)
- + [Blocks - Bit:Bot](#)
- + [Blocks - Bit:Commander](#)
- + [MicroPython - Starting Off](#)
- + [MicroPython - Examples](#)
- [MicroPython - Components](#)
 - ✦ [Introduction](#)
 - ✦ [Buzzer With MicroPython](#)
 - ✦ [LEDs With MicroPython](#)
 - ✦ [Connecting micro:bits Together](#)
 - ✦ [Extra Buttons](#)
 - ✦ [Knock Sensor](#)
 - ✦ [Rotary Encoder](#)
 - ✦ [Potentiometer](#)
 - ✦ [Soft Potentiometer](#)
 - ✦ [Flex Sensor](#)
 - ✦ [Tilt Sensor](#)
 - ✦ [Reed Switch](#)
 - ✦ [More Buttons](#)
 - ✦ [Temperature Sensor](#)
 - ✦ [7 Segment Display](#)
 - ✦ [Reflectance Sensor](#)
 - ✦ [Driving A Motor](#)
 - ✦ [Shift Register](#)
 - ✦ [Shifting In](#)
 - ✦ [Neopixels](#)
 - ✦ [IR Break Beam Sensor](#)
 - ✦ [DIY MIDI Out](#)
 - ✦ [PCF8574A Port Expander](#)
 - ✦ [16x2 Character LCD Display](#)
 - ✦ [SNES Controller](#)
- + [MicroPython - Breakout Boards](#)
- + [MicroPython - Exercises](#)
- + [MicroPython - Pi Accessories](#)
- + [MicroPython - Bit:Bot](#)
- + [MicroPython - Bit:Commander](#)
- + [MicroPython - Projects](#)
- + [MicroPython - Visual Basic](#)
- + [Other - Odds & Ends](#)

