

Micro:fit - Your DIY Step Counter

Overview

Ever wonder how your smartwatch knows exactly how many steps you've taken? In this exciting project, we'll demystify fitness tracking technology by creating our own step counter using a micro:bit! Unlike expensive commercial devices, our DIY fitness tracker will help you understand the basic principles of motion detection and how simple sensors can count your daily steps.

Using the micro:bit's built-in accelerometer, we'll program it to detect movement patterns that match walking or running motions. You'll learn how real fitness trackers work while building your own wearable device that displays your step count right on the LED matrix. Best of all, you'll gain hands-on experience with physical computing, data processing, and real-world sensor applications.

Whether you're interested in wearable technology, fitness tracking, or just love creating cool gadgets, this project offers a perfect blend of computer science and physical activity. By the end, you'll have your very own working step counter and a deeper understanding of the technology we use every day!

What you will Learn

How to:

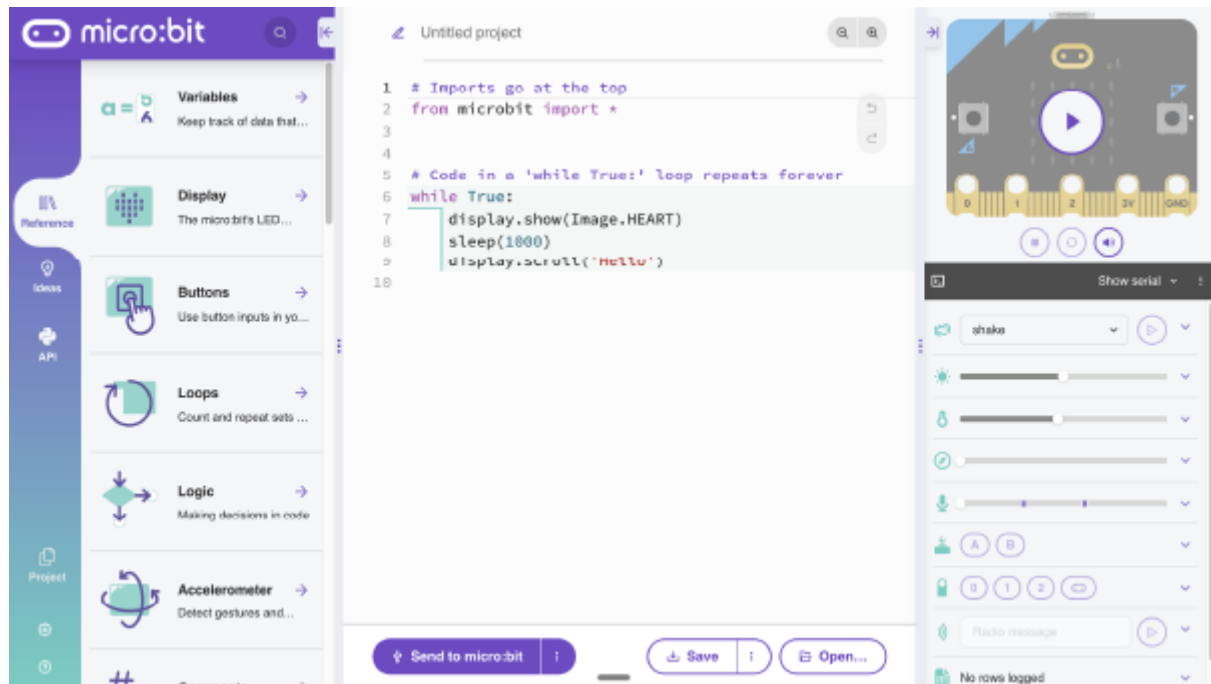
- ☐ Create and use a variable.
- ☐ The micro:bit shake function.
- ☐ Use a while True loop.
- ☐ Display a number on the LED matrix.
- ☐ Use the button function.

What you will Need

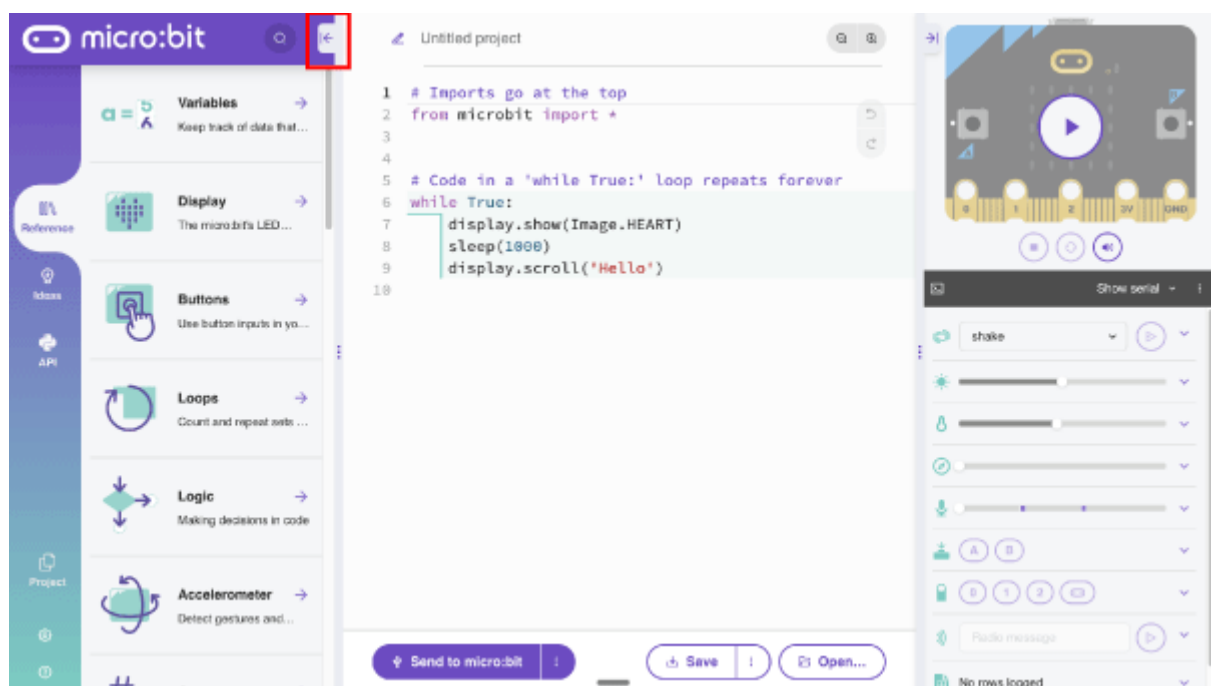
- 1 x micro:bit
- 1 x micro USB cable
- 1 x battery pack for the micro:bit (optional)

Navigating to the Python Editor

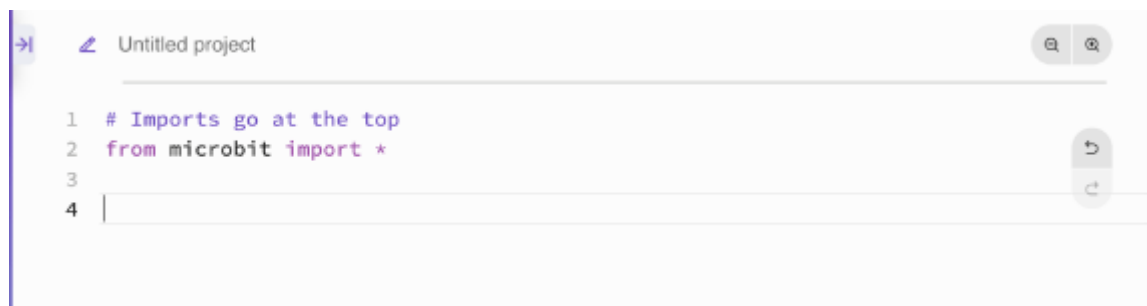
1. Open your favourite browser (we recommend Google Chrome).
2. Within the address bar of the browser type python.microbit.org.



3. Close the left hand panel by selecting the arrow pointing to the left. See the image below.



4. Delete the code from line 5 -9 on the main code area.



We are now ready to start coding!

Coding

Importing Python Libraries

On line 3 type the following code to import the time library.

```
import time
```

Press enter to start a new line of code.

Creating Variables

WHAT IS A VARIABLE

Think of a variable as a box that stores information that can be used throughout our program. We give variables a descriptive name so we and others can understand what is going on within our program.

Type the following code to create the **steps** variable and set it to **0**.

```
steps = 0
```

Creating the while True loop

Type the following line of code under `steps = 0`

```
while True:
```

Once you press enter your cursor will automatically indent. Start typing the rest of your code from here.

Adding 1 to Steps

Let's create the if function to detect when the micro:bit was shaken and add 1 to our steps variable.

```
if accelerometer.was_gesture('shake'):
    steps += 1
```

Press Enter and make sure the cursor is lined up with the if statement. (this can normally be done by pressing **delete** once)

Pausing the Code

We need to pause our code here or when we try to reset the step counter our program will be running too quickly and not detect the button press.

```
sleep(500)
```

This pauses our code for half a second.

Resetting Our Step Counter to 0

Now that we have got our steps increasing by one when the micro:bit detects it's been shaken. At the start of a new walk or day we will want to start from 0 again.

Let's create some code that will set steps back to 0 when the A button has been pressed.

```
if button_a.was_pressed():  
    steps = 0
```

Press Enter on your keyboard and make sure that the cursor is back in line with the **if statement**.

Displaying the Steps

The last thing we need to do is display the number of steps on the micro:bit LED Matrix.

To do this type:

```
display.scroll(steps)
```

Completed Code

We have now completed our code and it should look like this:

```
# Imports go at the top  
from microbit import *  
  
steps=0  
  
while True:  
    if accelerometer.was_gesture('shake'):  
        steps += 1  
        sleep(500)  
    if button_a.was_pressed():  
        steps = 0  
    display.scroll(steps)
```

Well done you have now completed the code for the rock, paper scissors game. Let's move on to learn how to download it to the micro:bit.

Downloading the Code

Pairing the micro:bit to your computer

1. Take the micro USB cable and connect the micro:bit to the computer.
2. Select the **3 little dots** next to **send to micro:bit**.



3. Select **Connect** and follow the on screen prompts.

Downloading code to the micro:bit

1. Select **Send to micro:bit** to download the code to your micro:bit.

Lets move on to see how to play.

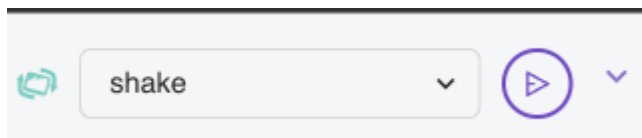
How to Play

Well done you have created your very FitBit Clone Step Counter for the micro:bit.

Web Browser

The Python Editor has a built-in micro:bit simulator so you can use this if you don't have a micro:bit handy.

Under the micro:bit simulator you will see that the shake function is selected and if you click on the play button next to it this will simulate the micro:bit being shaken and display the number of steps on the LED matrix.



Using the micro:bit

Once you have downloaded the code to your micro:bit you can shake the micro:bit and see your steps add up.

Find a way to attach the micro:bit to you and go outside for a walk or a run and see how many steps you have done at the end.