

Getting Started with the BBC micro:bit

The BBC micro:bit is a programmable device that allows a hands-on approach to coding, utilising and enabling digital systems. The new micro:bit with sound has a number of features and ways of interaction (Figure 1).

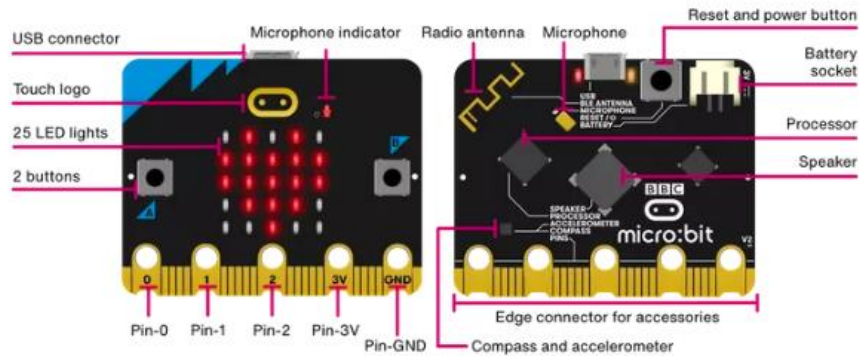


Figure 1: BBC micro:bit Front and Back

To start: Connect your micro:bit to your home computer/laptop by attaching the included USB lead to the USB connector on the micro:bit and the USB port on your computer.

In these example applications, the programming language Python (or more specifically MicroPython) is going to be used. An Integrated Development Environment (IDE) is needed to write code and transfer it to the micro:bit. Thonny is the IDE that will be used. This can be downloaded from <https://thonny.org/> for your particular machine.

Once downloaded and installed, open Thonny and look for the Options menu choice. Choose this and click on the interpreter tab.

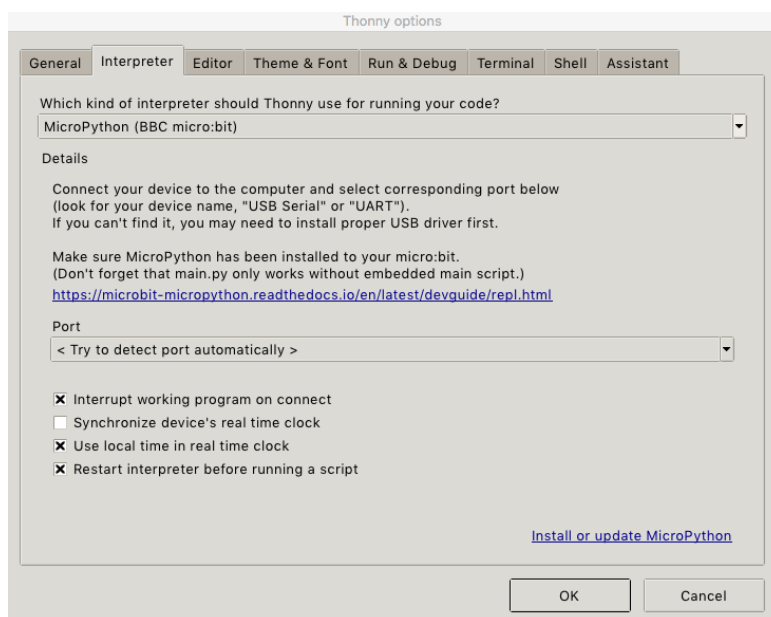


Figure 2: The Thonny Options Dialogue

Pick MicroPython (BBC micro:bit) from the 'Which kind of interpreter should Thonny use for running your code?' drop down menu. Then click on 'Install or update MicroPython'. Make sure you choose Variant: *BBC • micro:bit v2 (original simplified API)*

When completed programming can now begin. In Thonny enter the code as shown in Figure 3 in the top window.

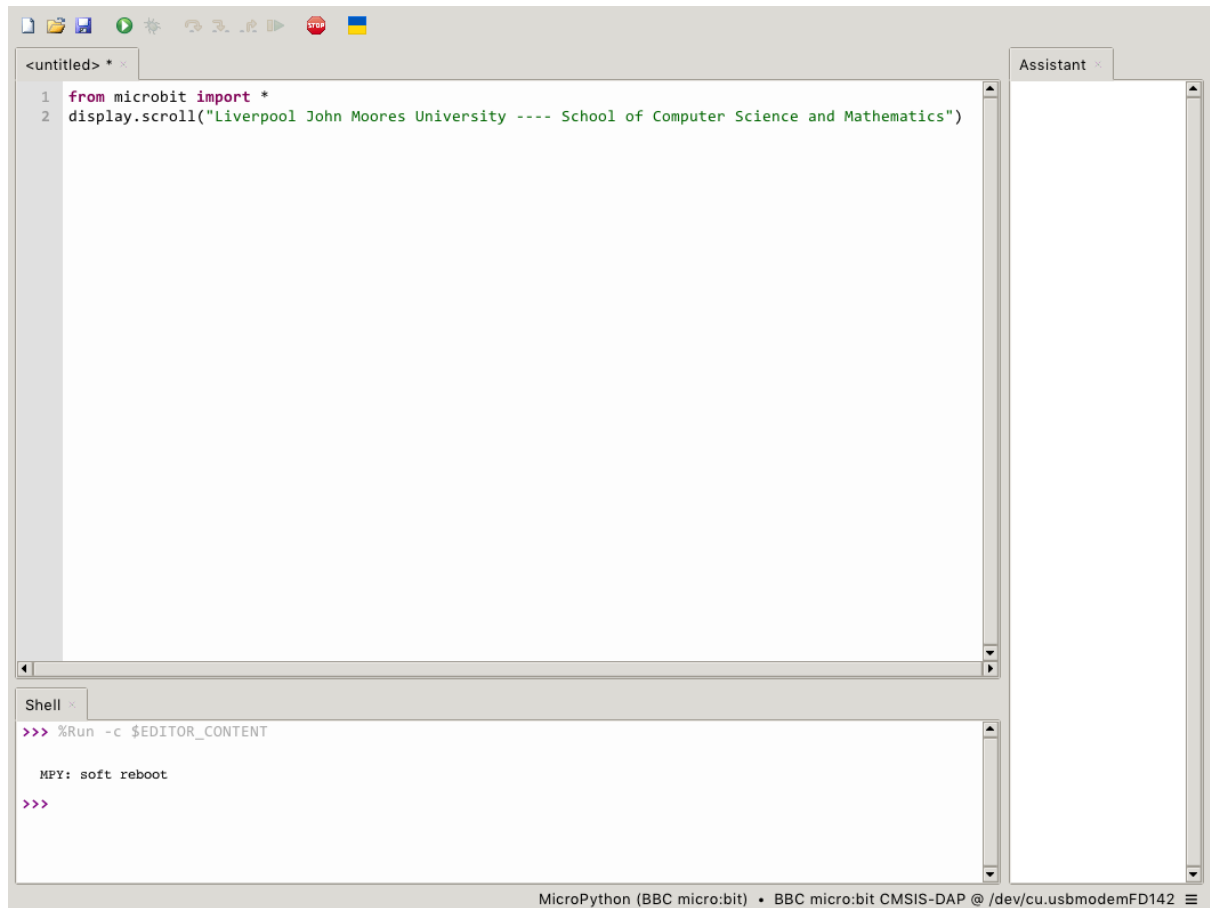
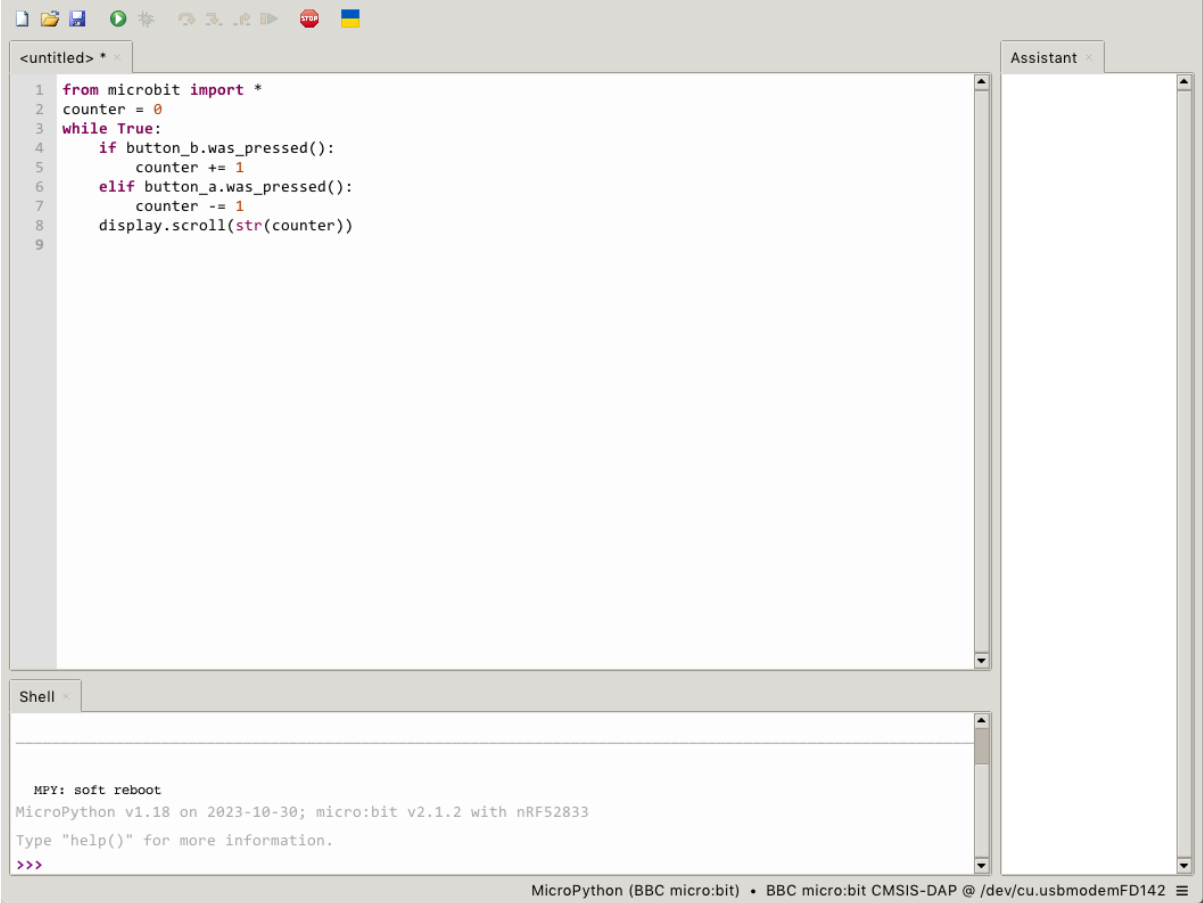


Figure 3: Entering Code into Thonny

Now press the Run (current script) Button (green button with the white arrow at the top of the window).

You can change the message to anything you want to scroll across your micro:bit.

We can use variables to capture and store information. In the example shown in Figure 4, *counter* is a variable that holds a number, that is incremented by button b and decremented by button a, with an initial value of 0. The while loop is always enacted, as the condition is true, and runs until you stop it. The value of the variable *counter* is then scrolled across the micro:bit. Enter the code as shown in Figure 4 and press the Run button again



```
<untitled> * x
1 from microbit import *
2 counter = 0
3 while True:
4     if button_b.was_pressed():
5         counter += 1
6     elif button_a.was_pressed():
7         counter -= 1
8     display.scroll(str(counter))
9

Assistant x

Shell x

MPY: soft reboot
MicroPython v1.18 on 2023-10-30; micro:bit v2.1.2 with nRF52833
Type "help()" for more information.
>>>
```

MicroPython (BBC micro:bit) • BBC micro:bit CMSIS-DAP @ /dev/cu.usbmodemFD142

Figure 4: Using a Variable in MicroPython

This is not intended to be a guide to programming or even learning MicroPython. You can, however, visit the website https://microbit-micropython.readthedocs.io/en/v2-docs/microbit_micropython_api.html to obtain more details and help with the language.

Finally, in this 'Getting Started section: Can you, firstly, put the tasks completed so far together and write a programme that displays the message when you press button a. Secondly, have a variable hold the message and scroll this across the device