

micro:bit and Push Buttons



Contents

- Introduction to micro:bit
- Using the built-in Buttons (A and B)
- Micro:bit I/O Pins
- Using External Push Buttons (with Examples)
 - External Push Button with Pull-up Resistor
 - External Push Button with Pull-down Resistor
 - Push Button and LED

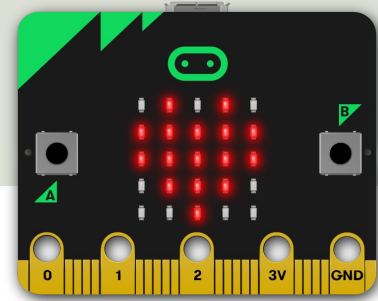


Introduction to micro:bit

Hans-Petter Halvorsen

[Table of Contents](#)

micro:bit

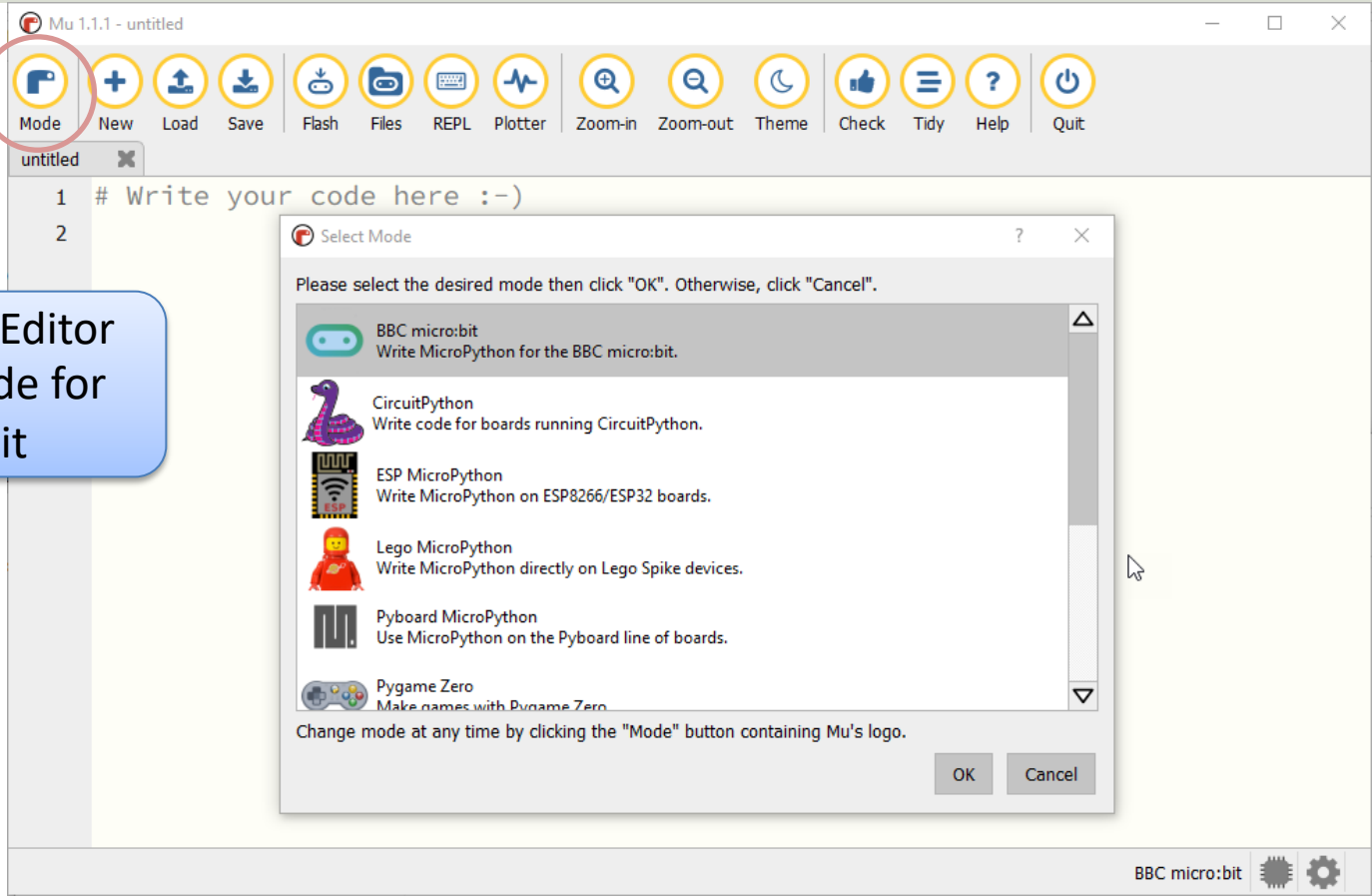


- micro:bit is a small microcontroller
- micro:bit is smaller than a credit card
- Price is about 150-400NOK (\$15-30)
- It can be used by kids and students to learn programming and technology
- micro:bit can run a special version of Python called MicroPython
- MicroPython is a down-scaled version of Python
- micro:bit Python User Guide
<https://microbit.org/get-started/user-guide/python/>
- micro:bit MicroPython documentation
<https://microbit-micropython.readthedocs.io>

Mu Python Editor

- Mu is a Python code editor for beginners
- It is tailor-made for micro:bit programming
- Mu has a “micro:bit mode” that makes it easy to work with micro:bit, download code to the micro:bit hardware, etc.
- Mu and micro:bit Tutorials:
<https://codewith.mu/en/tutorials/1.0/microbit>

Mu Python Editor



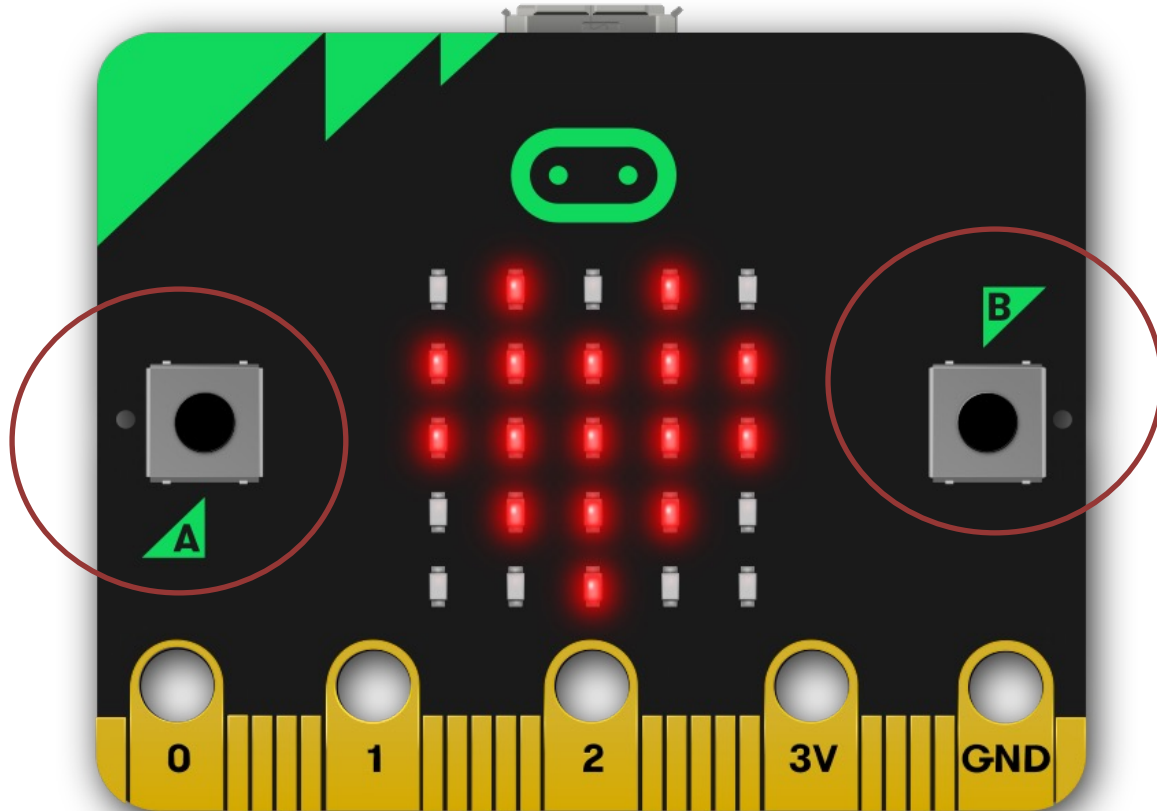
The screenshot shows the Mu Python Editor interface. The title bar reads "Mu 1.1.1 - untitled". The toolbar contains several icons, with the "Mode" icon (a blue folder with a white 'P') circled in red. Below the toolbar, the code editor shows two lines of text: "1 # Write your code here :-)" and "2". A "Select Mode" dialog box is open in the foreground, displaying a list of modes: BBC micro:bit, CircuitPython, ESP MicroPython, Lego MicroPython, Pyboard MicroPython, and Pygame Zero. The dialog box has "OK" and "Cancel" buttons at the bottom.

The Mu Python Editor has built-in Mode for the micro:bit



Using the built in Buttons (A and B)

Buttons (A and B)



Buttons (A and B) Example

```
from microbit import *

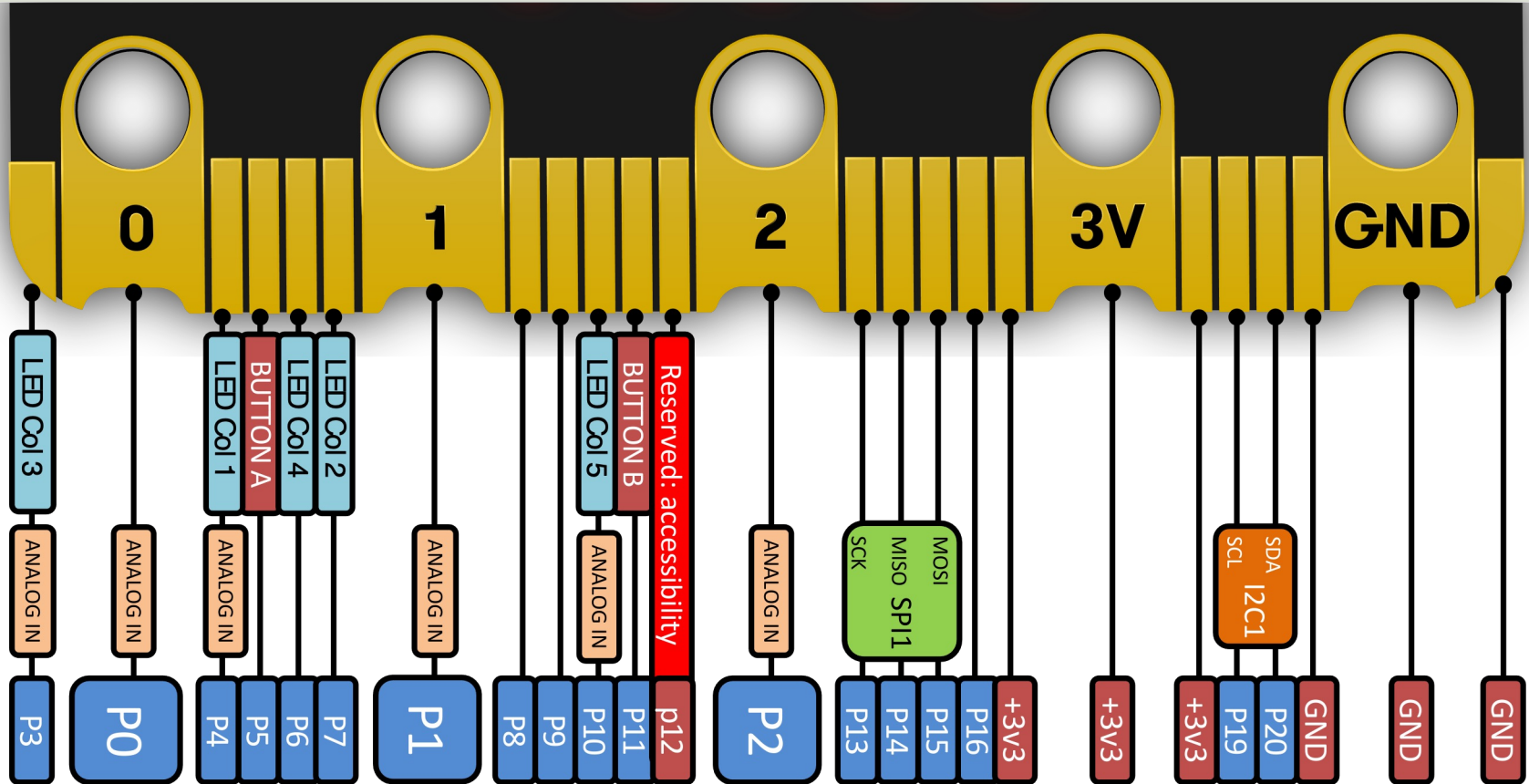
while True:
    if button_a.was_pressed():
        display.scroll("A")
    elif button_b.was_pressed():
        display.scroll("B")
    else:
        display.scroll("?")

    sleep(1000)
```

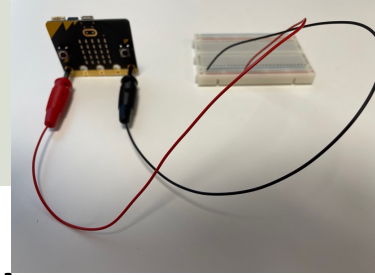


micro:bit I/O Pins

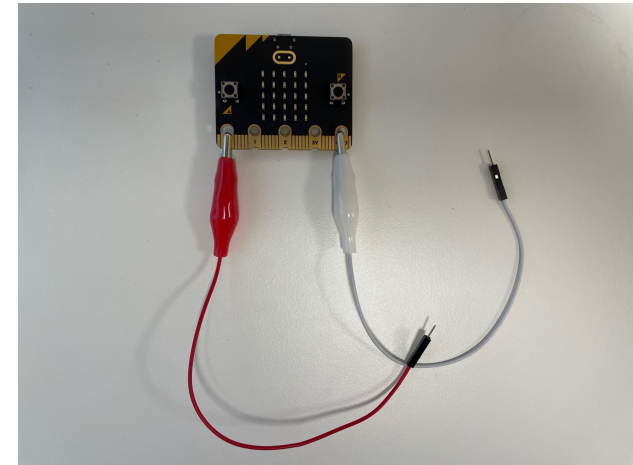
micro:bit I/O Pin Overview



I/O Pins



- We use the I/O pins to connect external components like LEDs, different types of Sensors, Push Buttons, etc.
- You can use 4mm Banana plugs or Alligator/Crocodile clips
- Typically, you also want to use a Breadboard



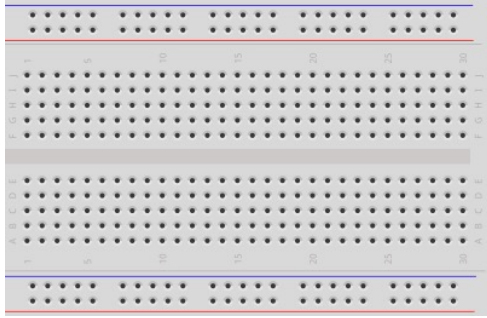
Types of I/O Pins

- **Analog/Digital Input/Output Pins**
- **Pulse Width Modulation (PWM)**
- SPI
- I2C
- UART (used for serial communication)

<https://microbit-micropython.readthedocs.io/en/latest/pin.html>

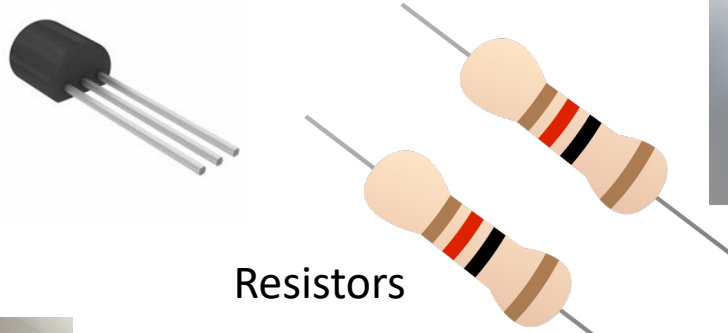
We will only use an Analog/Digital Input/Output pins in this Tutorial

Component Examples

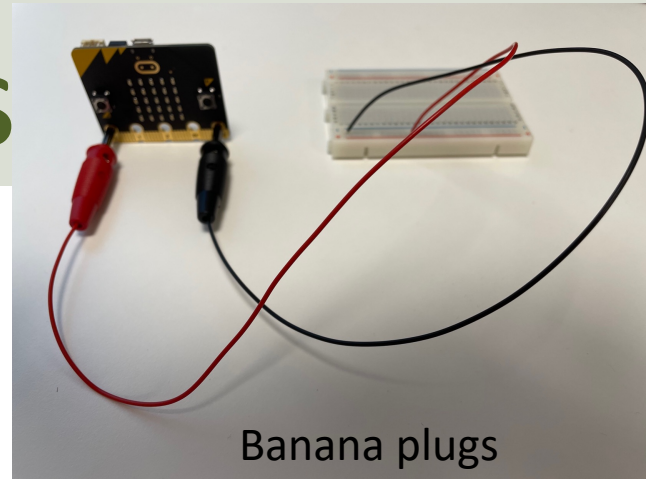


Breadboard

Temperature Sensor



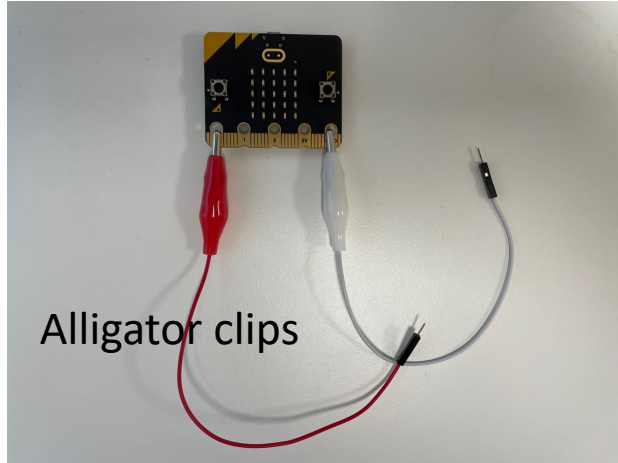
Resistors



Banana plugs

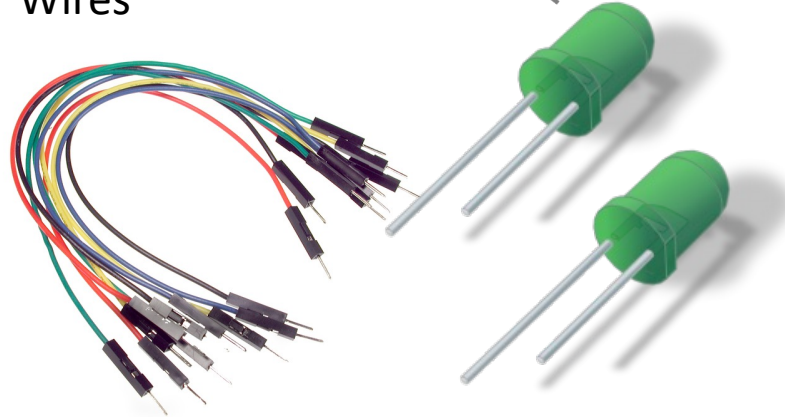
LEDs

Multimeter

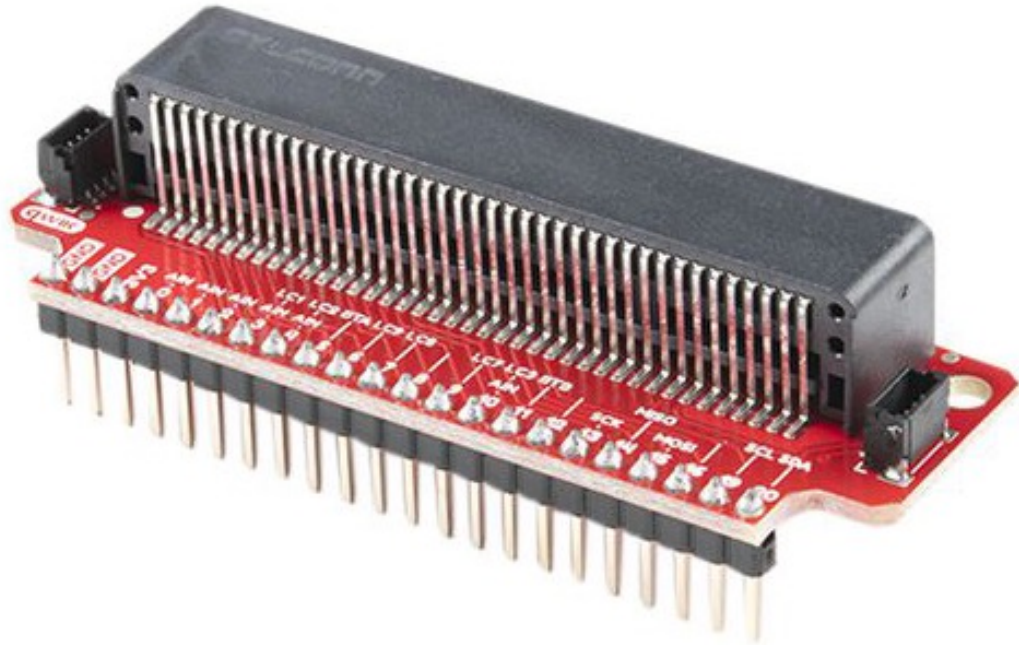


Alligator clips

Wires



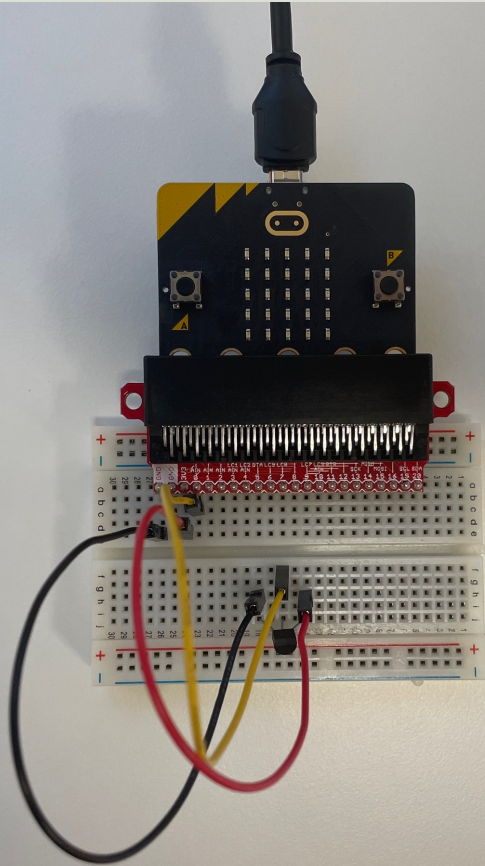
Adapter Breakout Board for micro:bit



We can also use an **Adapter Breakout Board for micro:bit** instead of Alligator/Crocodile clips

This makes it easier to wire for more advanced circuits and use of more in inputs/outputs pins

Adapter Breakout Board for micro:bit



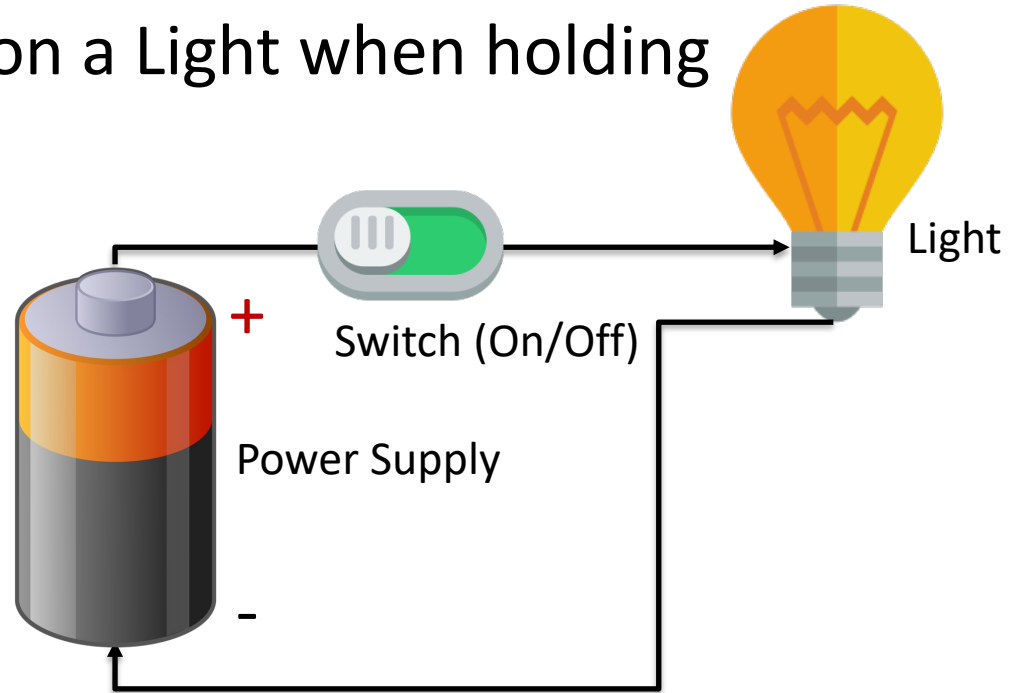
Here you see see the wirings using an Adapter Breakout Board for micro:bit



External Push Buttons

Push Button/Switch

- Pushbuttons or switches connect two points in a circuit when you press them.
- You can use it to turn on a Light when holding down the button, etc.



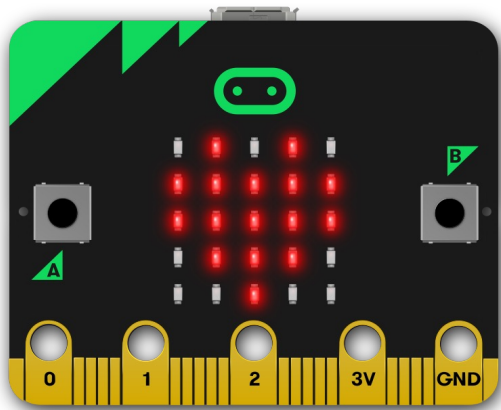
Button Examples

- Push Button using Pull-up Resistor
- Push Button using Pull-down Resistor
- Push Button + LED



External Push Button Pull-up Resistor

Wiring (Pull-up Resistor)

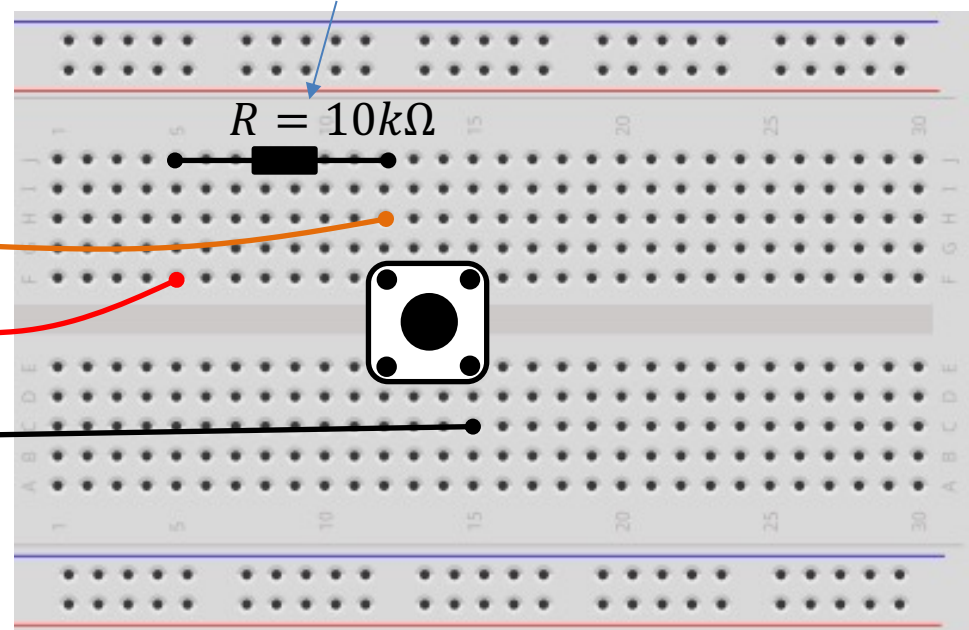


Using external Pull-up Resistor

Input,
e.g., pin0

3V

GND

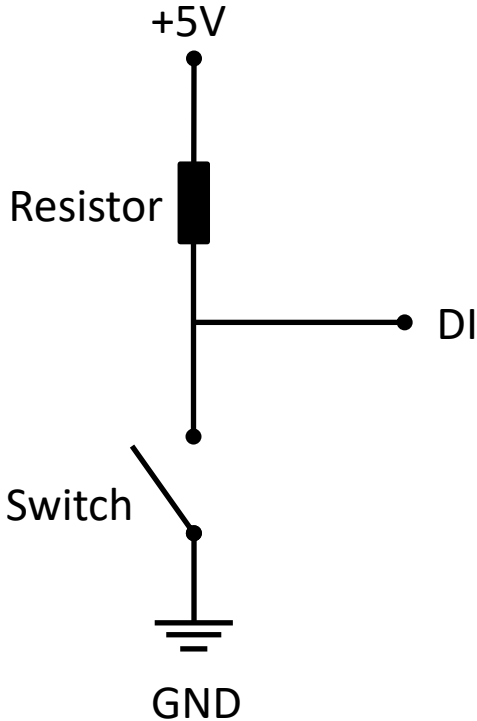


Pull-down/Pull-up Resistor

Why do we need a pull-up or pull-down resistor in the circuit?

- If you disconnect the digital I/O pin from everything, it will behave in an irregular way.
- This is because the input is "floating" - that is, it will randomly return either HIGH or LOW.
- That's why you need a pull-up or pull-down resistor in the circuit.

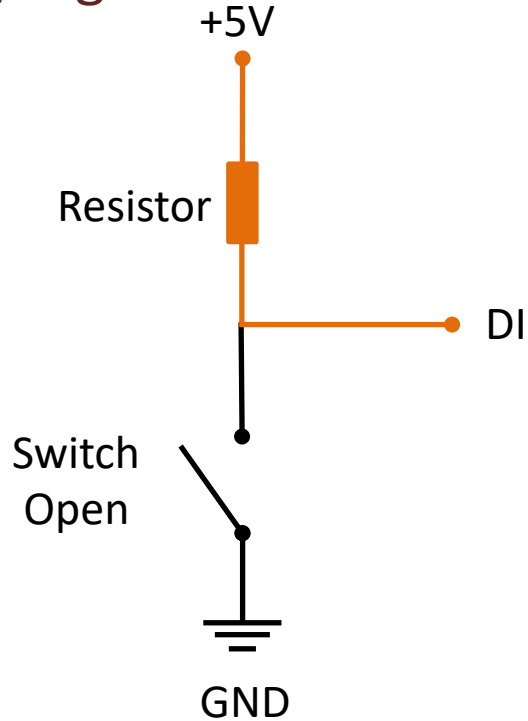
Pull-up Resistor



- When the pushbutton is open (unpressed) there is a connection between 5V and the DI pin.
- This means the default state is **True** (High).
- When the button is closed (pressed), the state goes to **False** (Low).

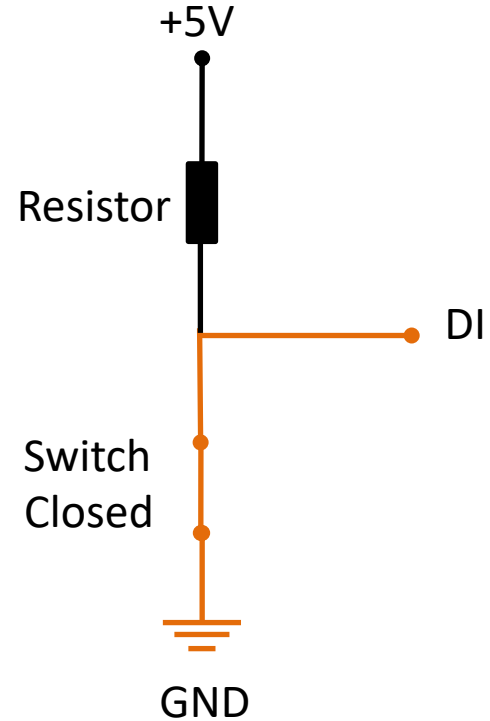
Pull-up Resistor

True/High

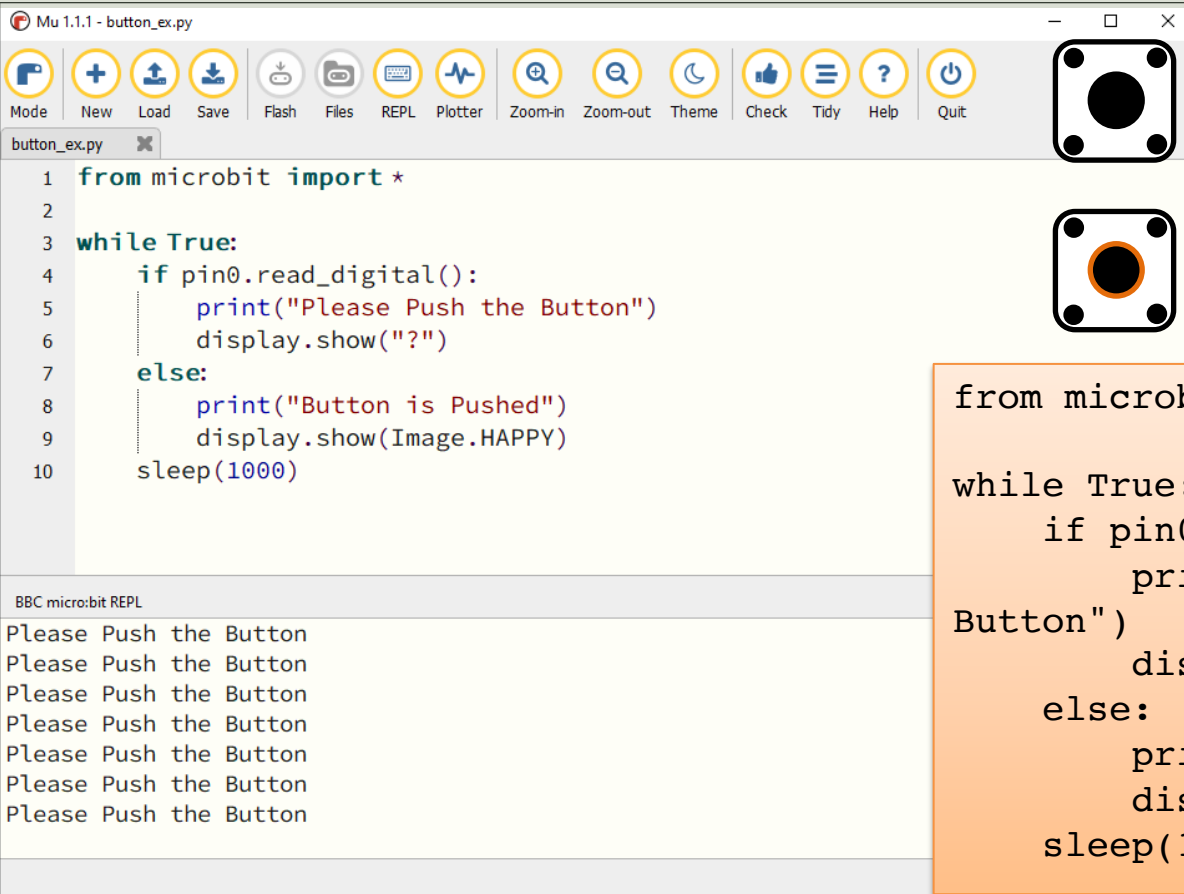


We Push the Button

False/Low



Push Button (Pull-up Resistor)



Mu 1.1.1 - button_ex.py

Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check Tidy Help Quit

```
1 from microbit import *
2
3 while True:
4     if pin0.read_digital():
5         print("Please Push the Button")
6         display.show("?")
7     else:
8         print("Button is Pushed")
9         display.show(Image.HAPPY)
10        sleep(1000)
```

BBC micro:bit REPL

Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button



Button is NOT Pushed => True/High



Button is Pushed => False/Low

```
from microbit import *

while True:
    if pin0.read_digital():
        print("Please Push the
Button")
        display.show("?")
    else:
        print("Button is Pushed")
        display.show(Image.HAPPY)
        sleep(1000)
```

Adding "NOT"

Mu 1.1.1 - button_ex2.py

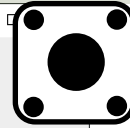
Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check Tidy Help Quit

```
1 from microbit import *
2
3 while True:
4     if not pin0.read_digital():
5         print("Button is Pushed")
6         display.show(Image.HAPPY)
7     else:
8         print("Please Push the Button")
9         display.show("?")
10
11     sleep(1000)
```

BBC micro:bit REPL

Please Push the Button
Please Push the Button
Button is Pushed
Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button
Button is Pushed

BBC micro:bit



Button is NOT Pushed => False/Low



Button is Pushed => True/High

```
from microbit import *

while True:
    if not pin0.read_digital():
        print("Button is Pushed")
        display.show(Image.HAPPY)
    else:
        print("Please Push the
Button")
        display.show("?")

    sleep(1000)
```

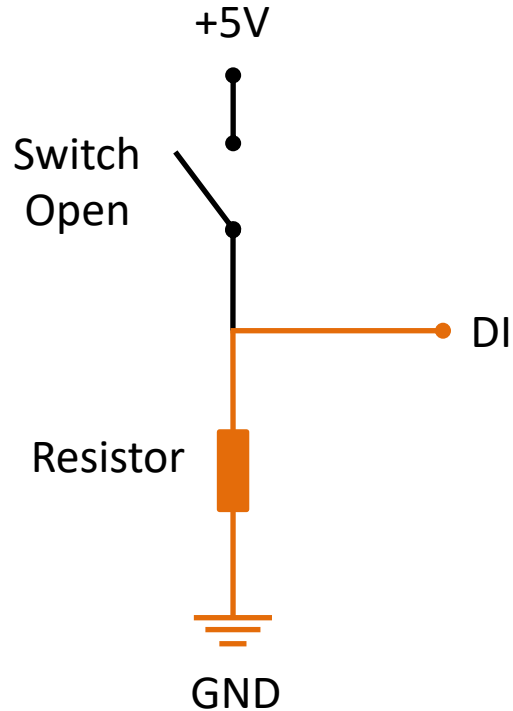


External Push Button Pull-down Resistor

Pull-down Resistor

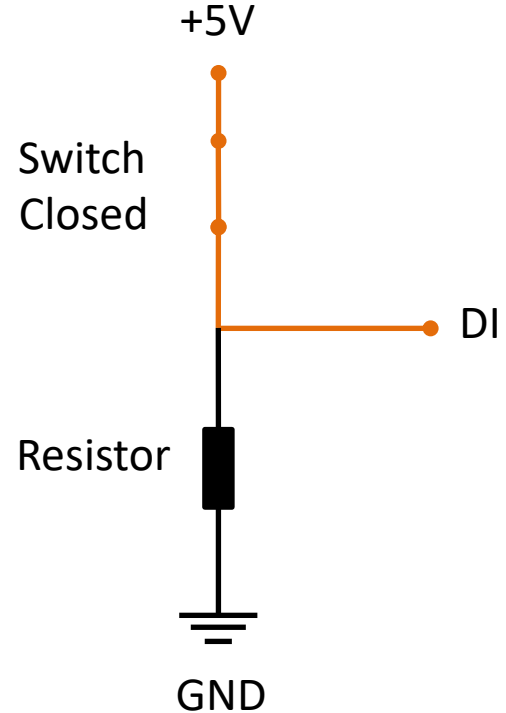
We could also have wired according to a “Pull-down” Resistor

False/Low



True/High

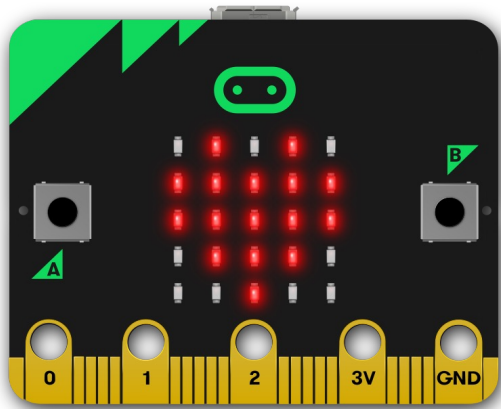
→
We Push the Button



Wiring (Pull-down Resistor)

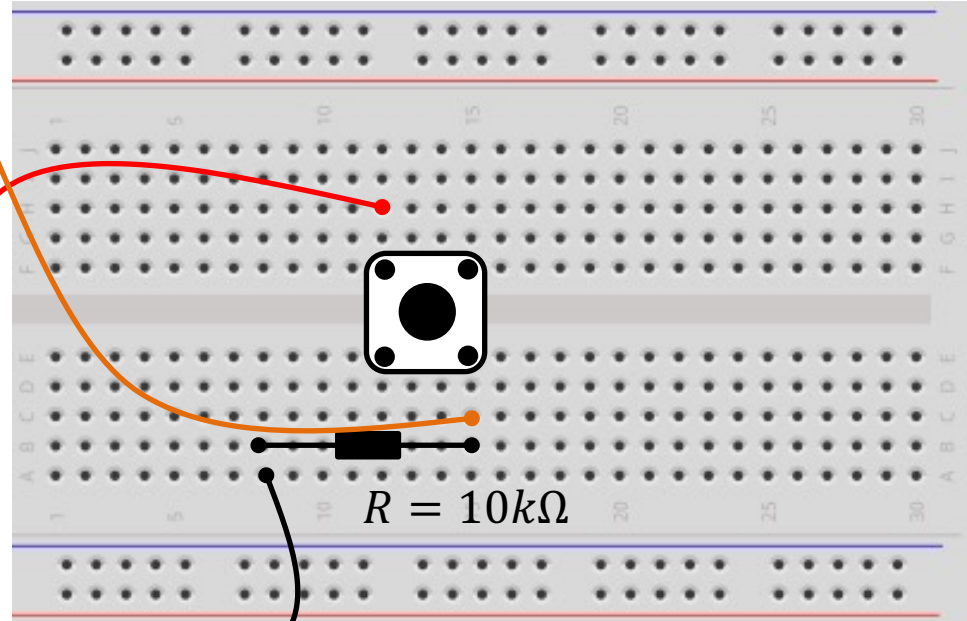
Another Digital Channel
can of course be used

Input,
e.g., pin0

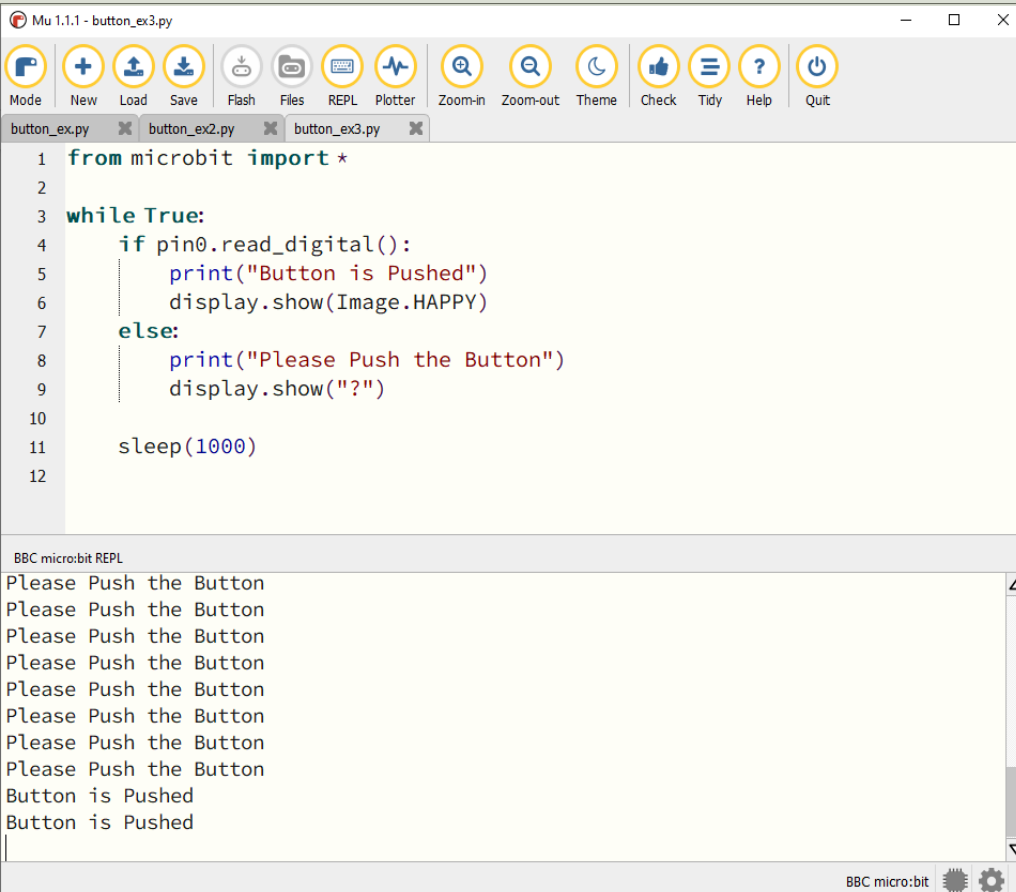


3V
GND

Using external Pull-down Resistor



Push Button (Pull-down Resistor)



The screenshot shows the Mu Python IDE interface. The top toolbar includes icons for Mode, New, Load, Save, Flash, Files, REPL, Plotter, Zoom-in, Zoom-out, Theme, Check, Tidy, Help, and Quit. The main editor displays the following Python code:

```
1 from microbit import *
2
3 while True:
4     if pin0.read_digital():
5         print("Button is Pushed")
6         display.show(Image.HAPPY)
7     else:
8         print("Please Push the Button")
9         display.show("?")
10
11     sleep(1000)
12
```

Below the code editor is the BBC micro:bit REPL window, which shows the output of the program:

```
Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button
Please Push the Button
Button is Pushed
Button is Pushed
```



Button is NOT Pushed => False/Low



Button is Pushed => True/High

```
from microbit import *

while True:
    if pin0.read_digital():
        print("Button is Pushed")
        display.show(Image.HAPPY)
    else:
        print("Please Push the
Button")
        display.show("?")

    sleep(1000)
```

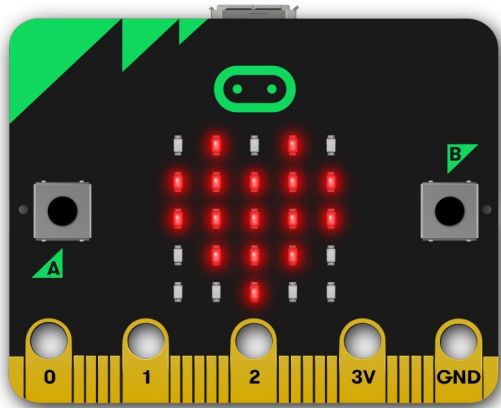


Push Button and LED

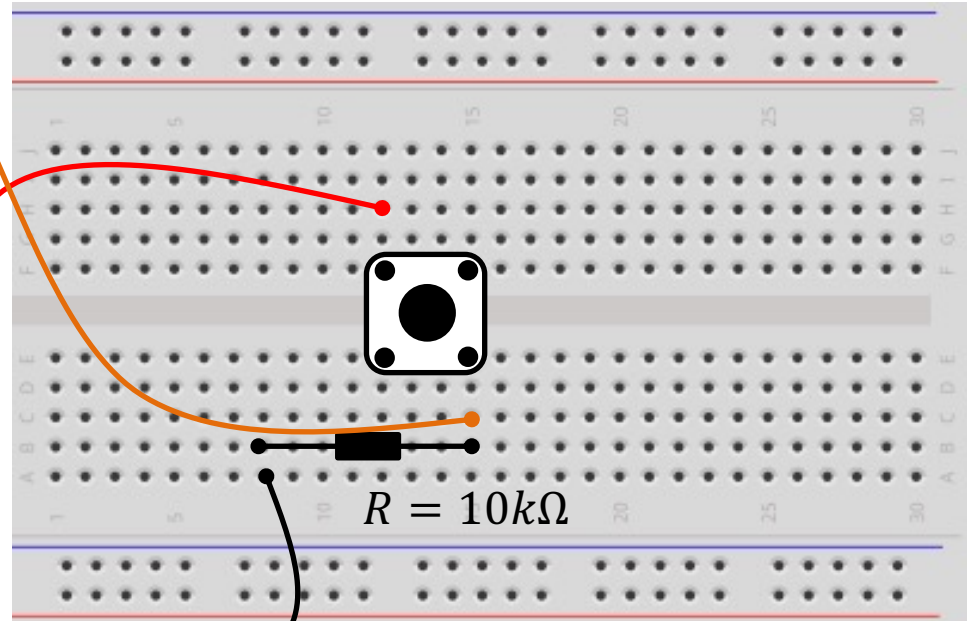
Wiring Push Button

Another Digital Channel
can of course be used

Input,
e.g., pin0



3V
GND

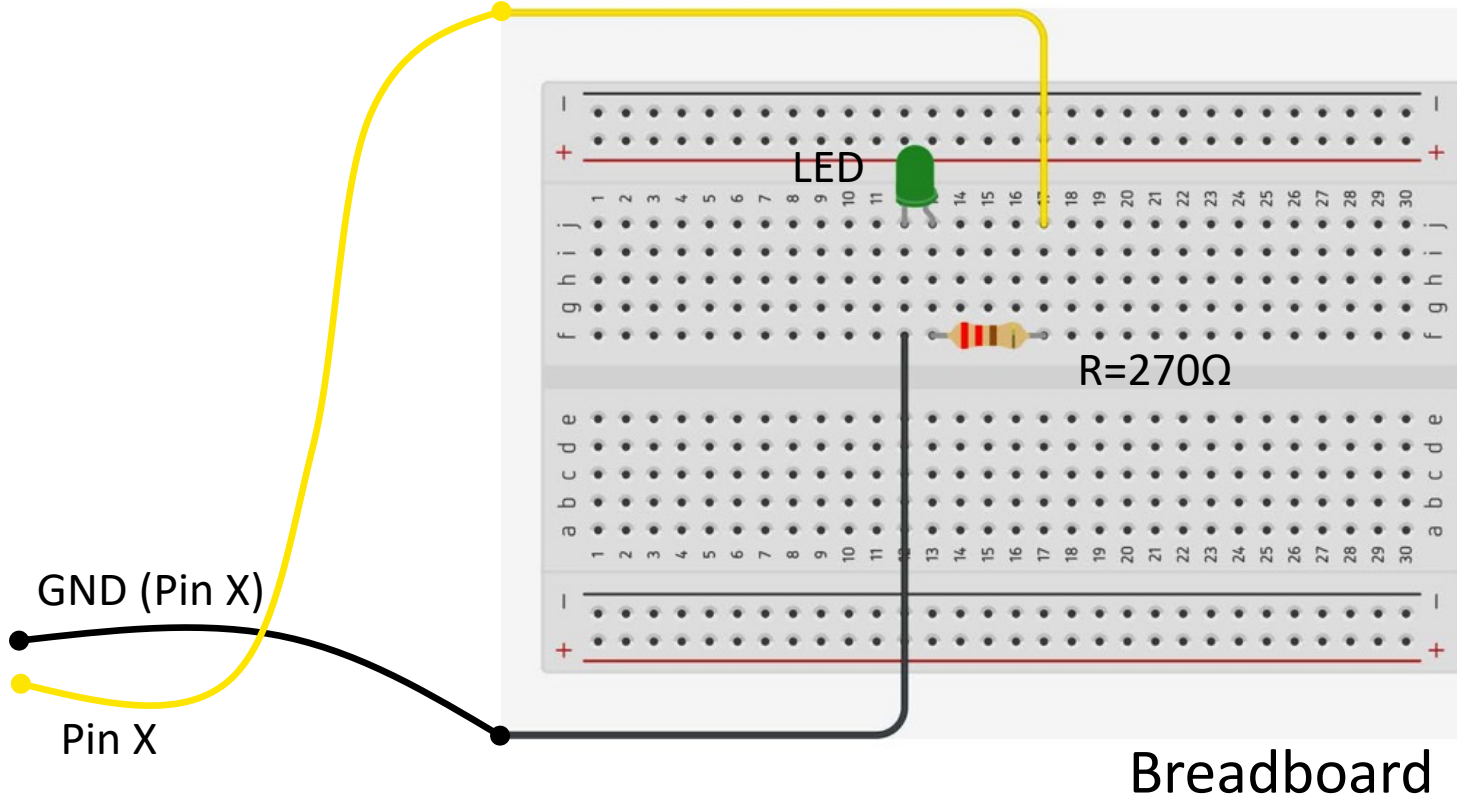


$R = 10k\Omega$

Using external Pull-down Resistor

Wiring LED

micro:bit GPIO Pins



Python

```
from microbit import *

while True:
    if pin0.read_digital():
        print("Button is Pushed")
        display.show(Image.HAPPY)
        pin1.write_digital(1)
    else:
        print("Please Push the Button")
        display.show("?")
        pin1.write_digital(0)

    sleep(1000)
```

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

