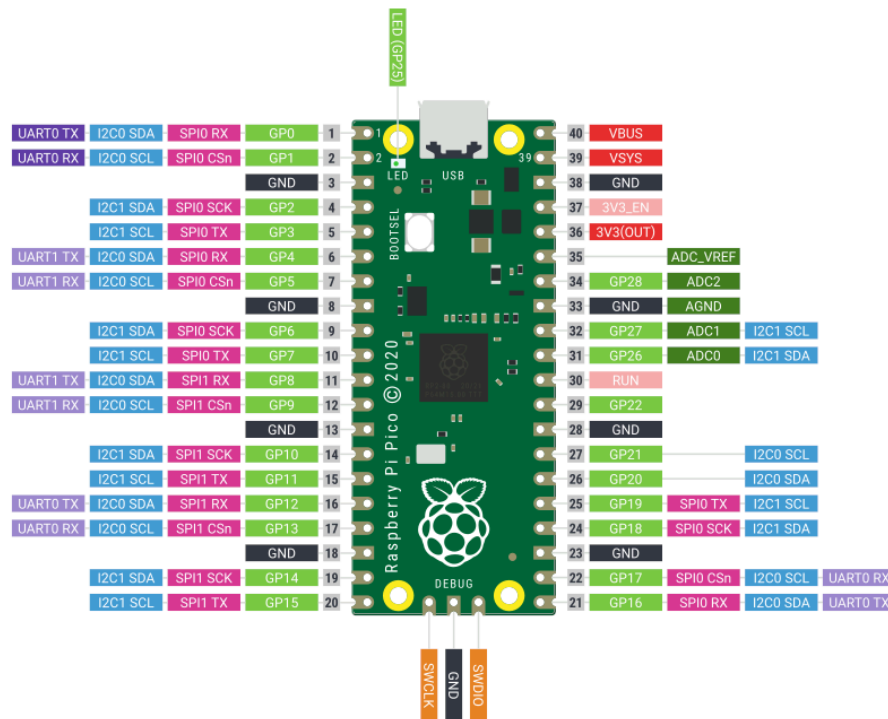


## Ćwiczenie nr 1: Wejścia/wyjścia cyfrowe

MicroPython jest implementacją języka Python 3, która może być uruchomiona w systemach wbudowanych takich, jak Raspberry Pi Pico lub Raspberry Pi Pico W. Przed rozpoczęciem pracy z modułem Raspberry Pi Pico należy zapoznać się z instrukcją *Drag-and-Drop MicroPython*, która jest dostępna pod linkiem:

[www.raspberrypi.com/documentation/microcontrollers/micropython.html](http://www.raspberrypi.com/documentation/microcontrollers/micropython.html)



**Zad 1.** Napisać program, który co pół sekundy będzie zmieniał stan LED na płytce Raspberry Pi Pico. Dioda elektroluminescencyjna jest podłączona do GPIO 25.

```
import machine
import utime as t

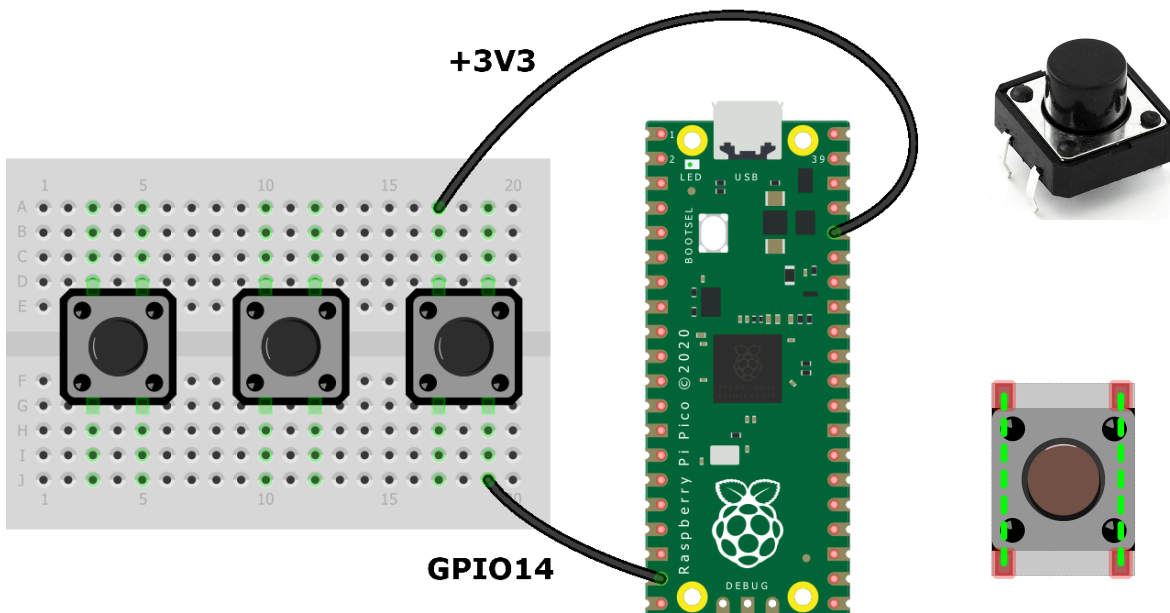
led_builtin = machine.Pin(25, machine.Pin.OUT)
i = 0
```

## Ćwiczenie nr 1: Wejścia/wyjścia cyfrowe

---

```
while True:
    led_builtin.toggle()
    t.sleep(0.5)
    i+=1
    print(f'Current interation is {i}')
```

**Zad 2.** Połączyć układ jak na rysunku. Przycisk jest podłączony do GPIO 14 (GP14). Napisać program, który wyśle komunikat *"Button pressed"* do komputera po naciśnięciu przycisku:



```
from machine import Pin
import utime

button = Pin(14, Pin.IN, Pin.PULL_DOWN)

while True:
    if button.value() == 1:
        print("Button pressed")
        utime.sleep(0.5)
```

## Ćwiczenie nr 1: Wejścia/wyjścia cyfrowe

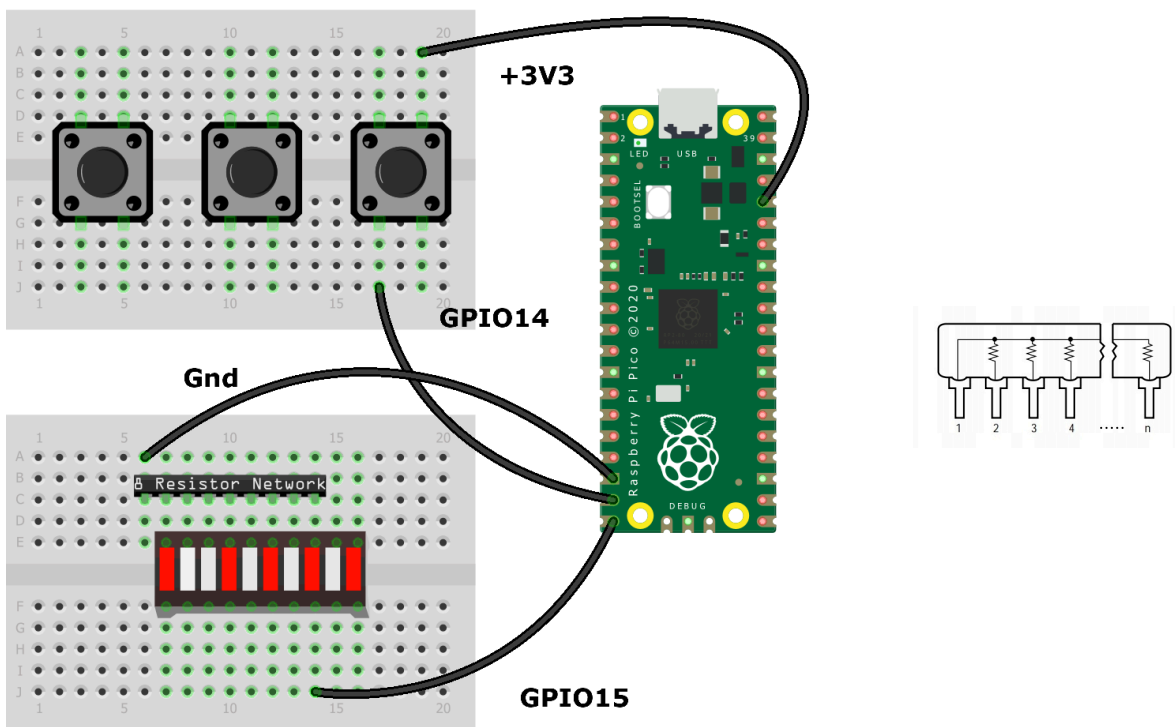
Pozwól użytkownikowi na podjęcie decyzji o treści komunikatu.

```
from machine import Pin
from utime import sleep

button = Pin(14, Pin.IN, Pin.PULL_DOWN)
message = input("Enter the message: ")
print('This is the message: ' + message)

while True:
    if button.value() == 1:
        print(message)
        sleep(0.5)
```

**Zad 3.** Połączyć układ jak na rysunku.



Jeżeli przycisk jest naciśnięty LED na płytce powinna być wyłączona. W przeciwnym wypadku LED powinna być włączona. Drabinka rezystorowa jest elementem, który zawiera w swojej strukturze 8 rezystorów, które posiadają 1 z wyprowadzeń połączone ze sobą.

```
from machine import Pin
import utime

button = Pin(14, Pin.IN, Pin.PULL_DOWN)
led = Pin(15, Pin.OUT)
led.value(1)

while True:
    if button.value() == 1:
        led.value(0)
    else:
        led.value(1)
        utime.sleep(0.1)
```

**Zad 4.** Korzystając z układu z zadania 3 napisać program, który pozwoli na wygenerowanie 4 impulsów światła po każdym naciśnięciu przycisku.

```
from machine import Pin
from utime import sleep

button = Pin(14, Pin.IN, Pin.PULL_DOWN)
led = Pin(15, Pin.OUT)

def led_blink(pin, times):
    for i in range(times):
        pin.toggle()
        sleep(0.5)
        pin.toggle()
        sleep(0.5)
```

```
led.value(0)

while True:
    if button.value() == 1:
        led_blink(led, 5)
    sleep(0.1)
```

**Zad 5.** Korzystając z układu, z zadania 3 napisać program implementujący funkcjonalność typowego włącznika oświetlania z przyciskiem monostabilnym.

```
from machine import Pin
from utime import sleep

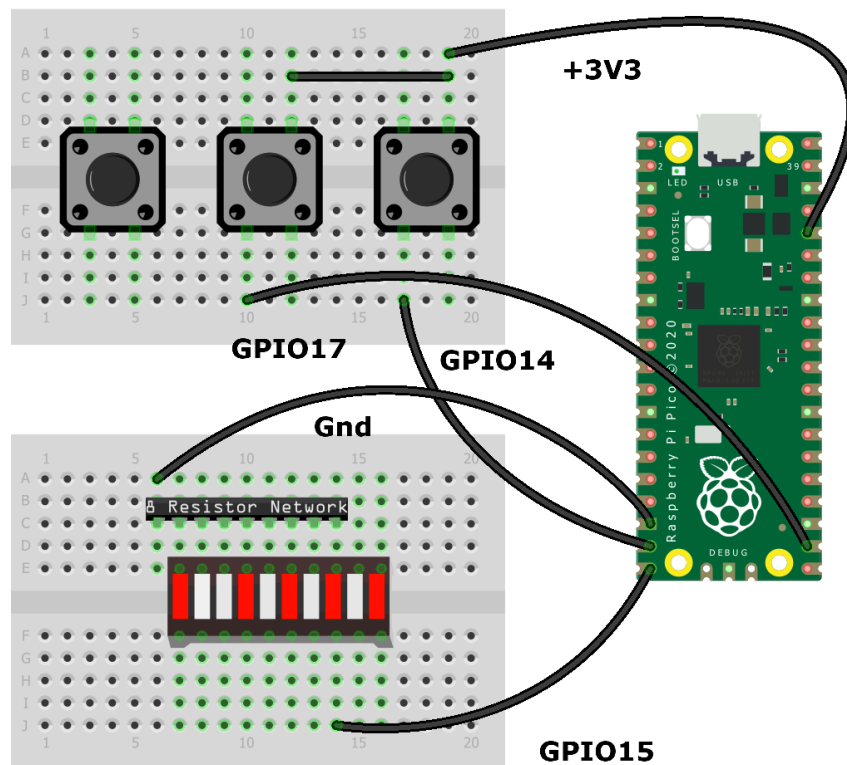
button = Pin(14, Pin.IN, Pin.PULL_DOWN)
light = Pin(15, Pin.OUT)

def button_clicked(pin):
    if pin.value() == 1:
        sleep(0.1)
        while pin.value() == 1:
            pass
        return True
    else:
        return False

light.value(0)

while True:
    if button_clicked(button):
        light.toggle()
```

**Zad 6.** Połączyć układ jak na rysunku. 1 z przycisków powinien załączać podłączoną do płytki Raspberry Pi Pico LED, a 2 wyłączać. Dioda na płycie powinna zmieniać swój stan 1 raz na sekundę.



```
from machine import Pin
from utime import sleep
import _thread

button_L = Pin(17, Pin.IN, Pin.PULL_DOWN)
button_R = Pin(14, Pin.IN, Pin.PULL_DOWN)
buildin_led = Pin(25, Pin.OUT)
external_led = Pin(15, Pin.OUT)

def led_blink():
    while True:
        buildin_led.toggle()
        sleep(0.5)

_thread.start_new_thread(led_blink, ())
```

```
def button_clicked(pin):
    if pin.value() == 1:
        sleep(0.1)
        while pin.value() == 1:
            pass
        return True
    else:
        return False

while True:
    if button_clicked(button_R):
        external_led.value(1)
        print("R click")
    if button_clicked(button_L):
        external_led.value(0)
        print("L click")
```

**Zad 7. (0.5 punktu)** Podłączyć trzeci przycisk do płytki. Samodzielnie napisać program, który pozwala na załączanie i wyłączanie 3 wybranych segmentów. Załączenie / wyłączenie 1 segmentu powinno nastąpić 1 sekundę po naciśnięciu przycisku. 2 przycisk załącza / wyłącza 2 segmenty 2 sekundy po naciśnięciu przycisku. 3 przycisk załącza / wyłącza 3 segmenty 3 sekundy po naciśnięciu przycisku.

**Zad 8. (1 punkt)** Po naciśnięciu przycisku należy wysłać do komputera komunikat ile razy dany przycisk został naciśnięty. Użytkownik powinien mieć możliwość ustalenia komunikatu po każdym resecie urządzenia. Drugi przycisk powinien resetować licznik zliczający ilość naciśnień przycisku.

**Zad 9. (1 punkt)** Napisać program, który pozwala na sterowanie położeniem jednego załączonego segmentu.

**Zad 10. (1.5 punktu)** Zbudować prototyp urządzenia do pomiaru czasu reakcji.

**Do zaliczenia zadania wymagana jest prezentacja i przesłanie rozwiązania na adres wskazany w stopce instrukcji. Przed wysłaniem plik należy spakować lub zmienić rozszerzenie na txt.**

**Dla zainteresowanych:**

1. Strona MicroPython:

[micropython.org/download/rp2-pico/](https://micropython.org/download/rp2-pico/)

2. Python wprowadzenie:

[www.programiz.com/python-programming/first-program](https://www.programiz.com/python-programming/first-program)

3. Wprowadzenie do Raspberry Pi Pico:

[projects.raspberrypi.org/en/projects/getting-started-with-the-pico/0](https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico/0)