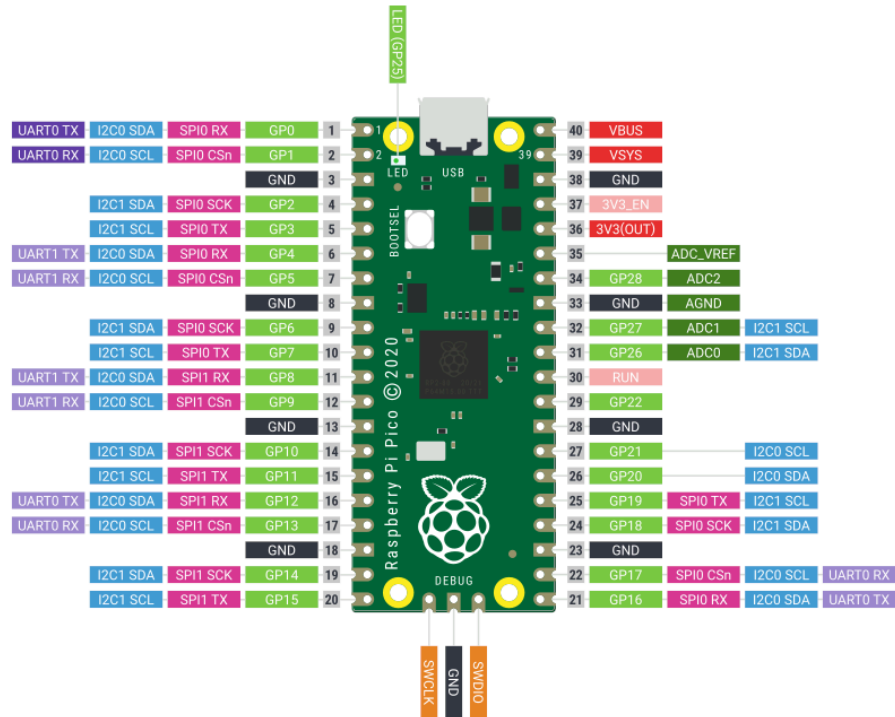


Ćwiczenie nr 1: Wejścia/wyjścia cyfrowe



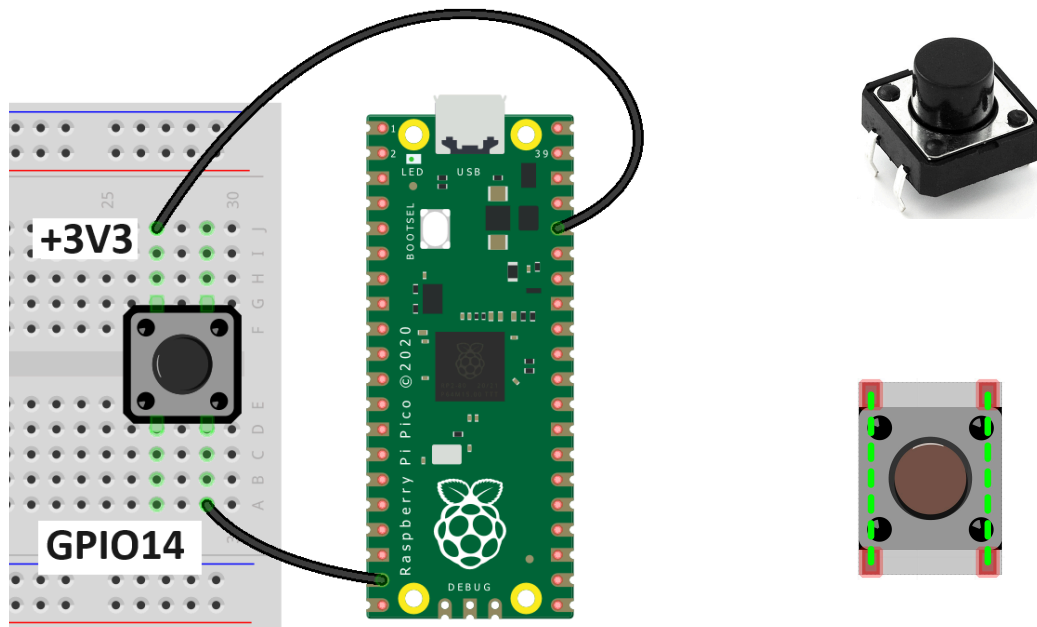
Zad.1. Napisać program, który co pół sekundy będzie zmieniał stan LED na płytce Raspberry Pi Pico. Dioda elektroluminescencyjna jest podłączona do GPIO 25.

```
import machine
import utime

led_builtin = machine.Pin(25, machine.Pin.OUT)

while True:
    led_builtin.toggle()
    utime.sleep(0.5)
```

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:



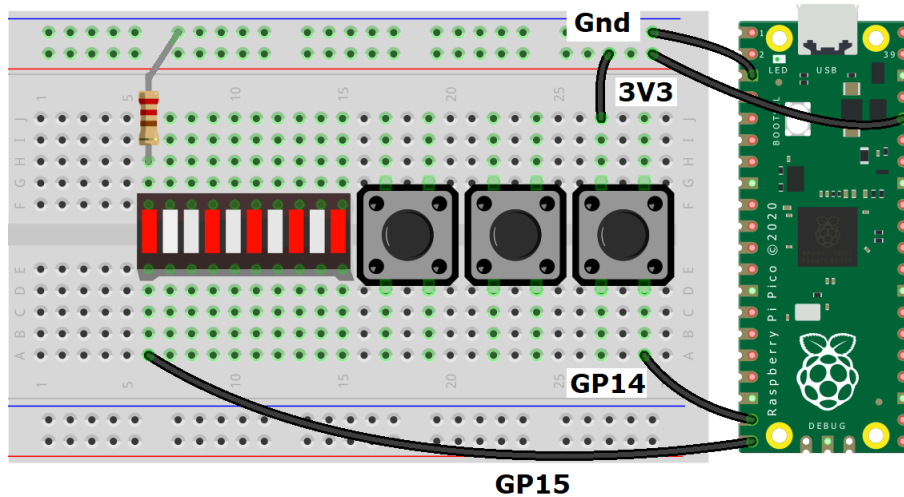
```
from machine import Pin
import utime

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

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

Zad.3. Połączyć układ jak na rysunku. Jeżeli przycisk jest naciśnięty LED na płytce powinna być wyłączona. W przeciwnym wypadku LED powinna być włączona.

Ćwiczenie nr 1: Wejścia/wyjścia cyfrowe



```
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)
```

Task.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)
```

Ćwiczenie nr 1: Wejścia/wyjścia cyfrowe

```
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
```

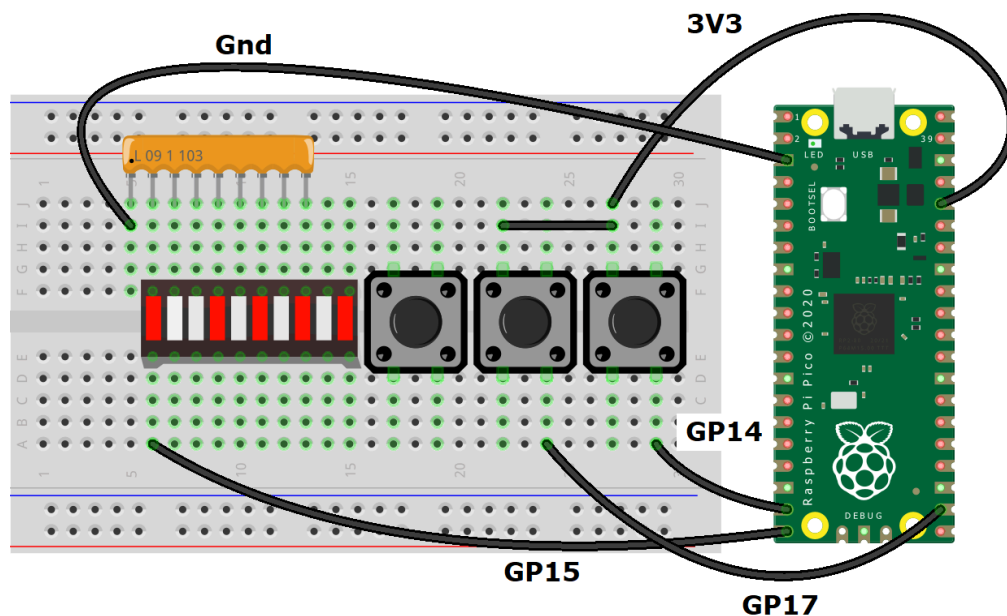
Ćwiczenie nr 1: Wejścia/wyjścia cyfrowe

```
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 płytce 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. Napisać program, który wyłączy LED 3s po naciśnięciu przycisku. 2 z przycisków powinien pełnić funkcję reset.

Zad.8. Podłączyć trzeci przycisk do płytki. Samodzielnie napisać program, który pozwala na załączanie i wyłączanie 3 wybranych segmentów.

Zad.9. Po naciśnięciu przycisku należy wysłać do komputera komunikat ile razy dany przycisk został naciśnięty.

Zad.10. Napisać program, który pozwala na sterowanie położeniem jednego załączonego segmentu.

Zad.11. Zbudować prototyp urządzenia do pomiaru czasu reakcji.

Dla zainteresowanych:

1. Strona MicroPython:

micropython.org/download/rp2-pico/

2. Python wprowadzenie:

www.programiz.com/python-programming/first-program

3. Wprowadzenie do Raspberry Pi Pico:

projects.raspberrypi.org/en/projects/getting-started-with-the-pico/0