

---

STEROWNIKI ROBOTÓW

# Laboratorium – Raspberry Pi Pico

Opługa i programowanie Raspberry Pi Pico w języku  $\mu$ Python

---

Wojciech Domski

## Spis treści

|          |                                                  |           |
|----------|--------------------------------------------------|-----------|
| <b>1</b> | <b>Wprowadzenie</b>                              | <b>3</b>  |
| <b>2</b> | <b>Opis ćwiczenia</b>                            | <b>3</b>  |
| <b>3</b> | <b>Opis środowiska</b>                           | <b>3</b>  |
| <b>4</b> | <b>Jak uruchomić płytke Raspberry Pi Pico W?</b> | <b>3</b>  |
| <b>5</b> | <b>Zadania do wykonania</b>                      | <b>8</b>  |
| 5.1      | Dostępne czujniki i moduły . . . . .             | 8         |
| 5.1.1    | Wbudowana dioda LED . . . . .                    | 8         |
| 5.1.2    | Czujnik światła BH1750 . . . . .                 | 8         |
| 5.1.3    | Wyświetlacz OLED SSD1306 . . . . .               | 8         |
| 5.1.4    | Dalmierz ultradźwiękowy HC-SR04 . . . . .        | 9         |
| 5.1.5    | Programowalne diody LED WS2812 . . . . .         | 9         |
|          | <b>Literatura</b>                                | <b>10</b> |

## 1 Wprowadzenie

Celem zadania jest zapoznanie się z podstawami programowania mikrokontrolera Raspberry Pi Pico w języku  $\mu$ Python. W ramach laboratoryjnego należy zapoznać się z obsługą mikrokontrolera RP2040 oraz komunikacji bezprzewodowej z użyciem modułu WiFi.

## 2 Opis ćwiczenia

Ćwiczenie składa się z kilku etapów, w jego ramach należy

1. obsługa wbudowanej diody LED za pomocą serwera HTTP,
2. obsługi zewnętrznego modułu,
3. uruchomienie serwera HTTP na RPI Pico do obsługi modułu.

## 3 Opis środowiska

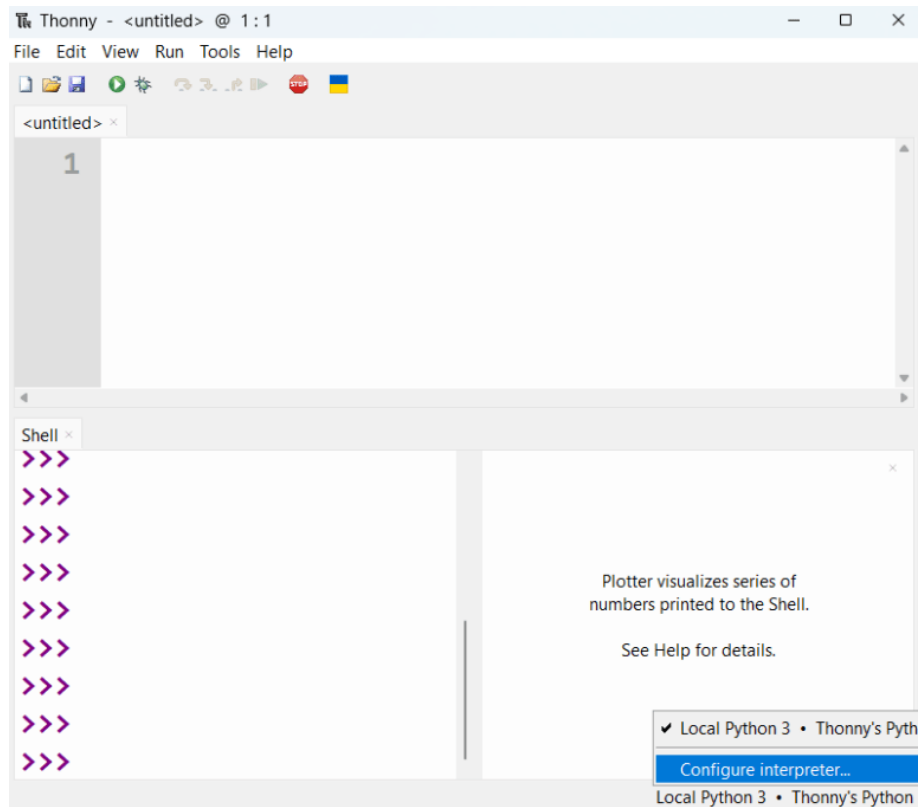
Informacje na temat Raspberry Pi Pico można znaleźć w książce *Get Started with MicroPython on Raspberry Pi Pico* [1]. Opis języka Python wykorzystywanego na Raspberry Pi Pico znajduje się pod adresem *Raspberry Pi Pico Python SDK* [4]. Natomiast opis wykorzystania funkcji związanych z bezprzewodową łącznością znajduje się pod adresem *Connecting to the Internet with Raspberry Pi Pico W* [2].

Informacje na temat Raspberry Pi Pico zostały opisane w [3].

Plik MicroPython UF2 dla wersji Raspberry Pi Pico W można znaleźć w [5].

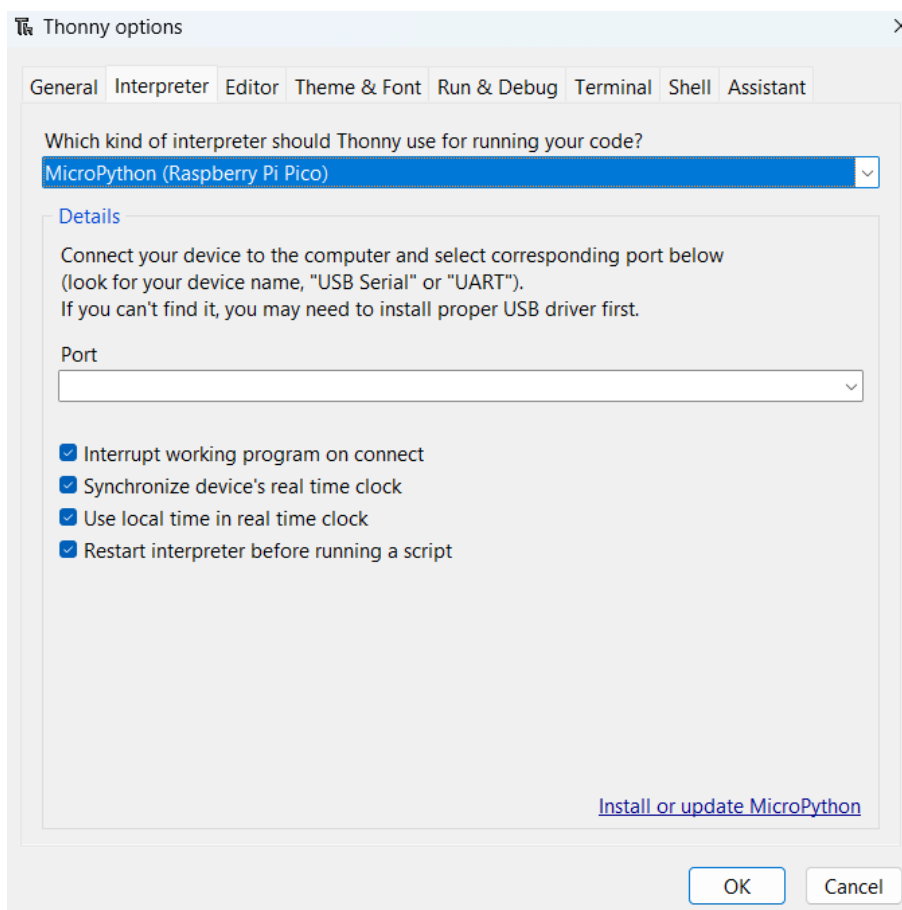
## 4 Jak uruchomić płytkę Raspberry Pi Pico W?

1. Należy ściągnąć plik UF2 [5]. Upewnij się, że plik ma rozmiar powyżej 1MB i został poprawnie pobrany.
2. Wciśnij i przytrzymaj przycisk BOOTSEL na płytce Raspberry Pi Pico. Trzymając przycisk podłącz płytkę do komputera. Odczekaj 2 sekundy oraz puść przycisk.
3. Płytką uruchomiła się w trybie ładowania programu (z ang. bootlader mode).
4. Wgraj plik UF2 na płytkę poprzez jego skopiowanie do nowego urządzenia, które pojawiło się w systemie. Odczekaj około 20 sekund i odłącz płytkę od komputera. Jeśli nowe urządzenie nie pojawiło się w systemie to odłącz płytkę od komputera i powtórz procedurę od pkt. 2.
5. Odłącz płytkę od portu USB i podłącz ją ponownie.
6. Otwórz terminal i uruchom środowisko thonny poprzez wpisanie komendy thonny.
7. W przypadku pierwszego uruchomienia środowiska wybierz tryb standardowy. Później będzie to można zmienić poprzez kliknięcie linku w aplikacji, który znajduje się w prawym górnym rogu (może być wymagane powiększenie okienka aplikacji).
8. Wybierz interpreter języka Python. W tym celu w prawym dolnym rogu należy skonfigurować interpreter (Rys. 1).



Rysunek 1: Środowisko Thonny

9. W zakładce Interpreter wybierz rodzaj interpretera. Należy wybrać MicroPython (Raspberry Pi Pico). W polu Port należy wybrać urządzenie szeregowe, które odpowiada za podłączoną płytkę Raspberry Pi Pico (Rys. 2).



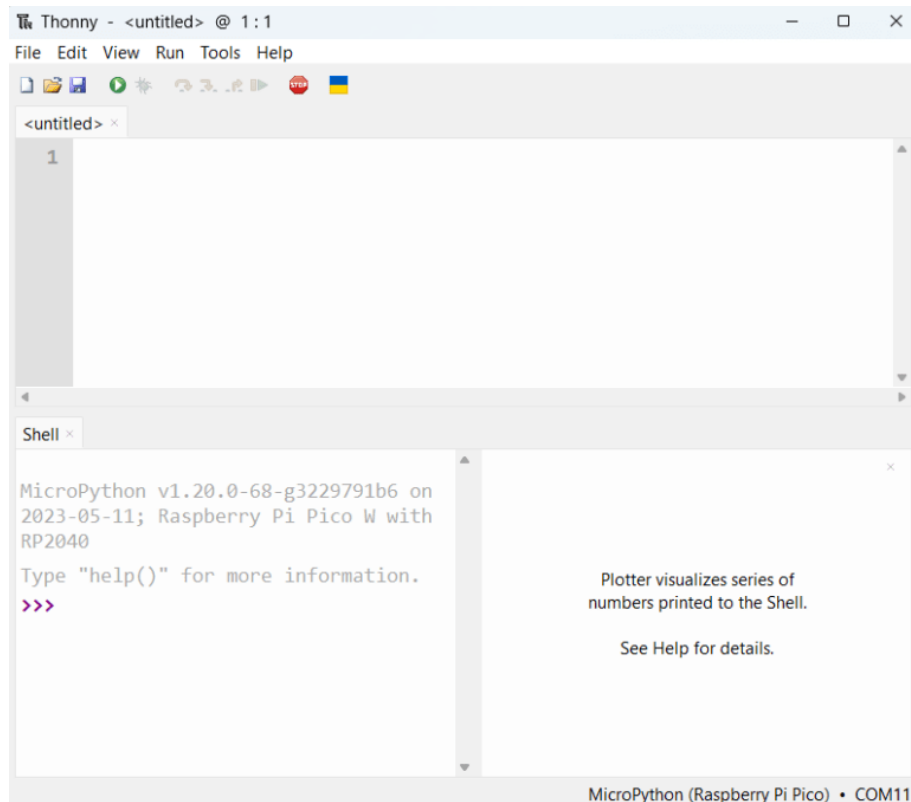
Rysunek 2: Wybór interpretera języka Python w środowisku Thonny

10. Po poprawnym skonfigurowaniu aplikacji i zakładając poprawne wgranie pliku UF2 w lewym dolnym panelu aplikacji Thonny (konsola, (z ang. Shell)) powinna ukazać się informacja podobna do tej jak na Rys. 3.

```

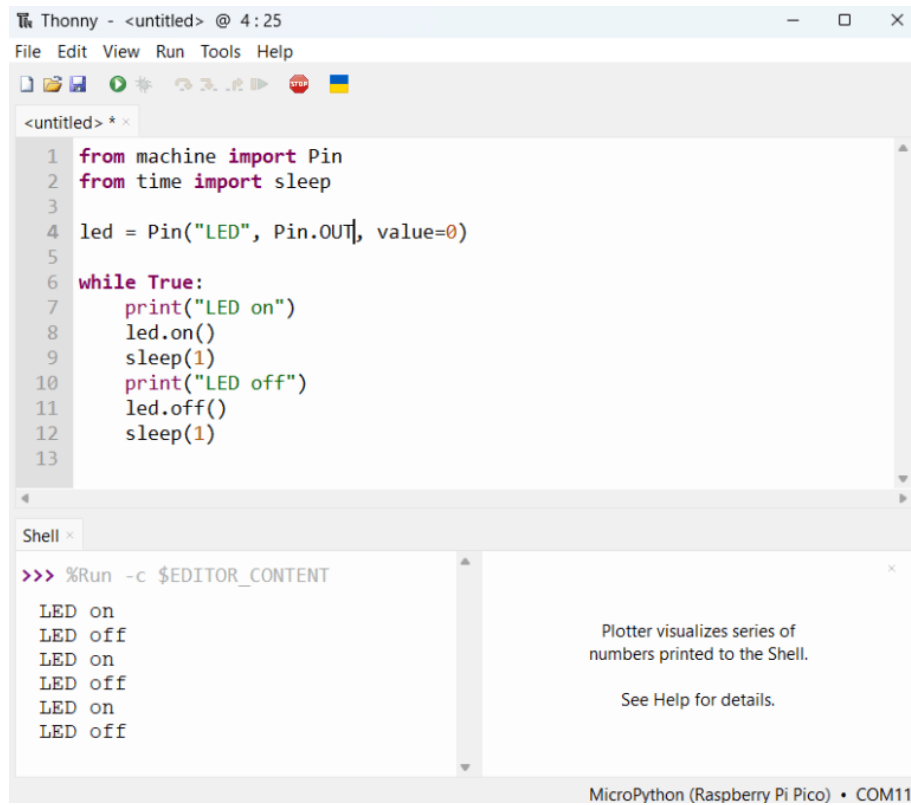
1 MicroPython v1.20.0-68-g3229791b6 on 2023-05-11; Raspberry Pi Pico W with RP2040
2 Type ,,help()'' for more information.

```



Rysunek 3: Uruchomiony interpretera języka Python w środowisku Thonny

11. Uruchom prostą aplikację, która będzie migać diodą LED. Uwaga! z racji tego, że jest to płytka Raspberry Pi Pico W diodę należy skonfigurować podając łańcuch znaków „LED” zamiast numeru portu. Przykład działania aplikacji został pokazany na Rys. 4.



```
Thonny - <untitled> @ 4:25
File Edit View Run Tools Help

<untitled> * x
1 from machine import Pin
2 from time import sleep
3
4 led = Pin("LED", Pin.OUT, value=0)
5
6 while True:
7     print("LED on")
8     led.on()
9     sleep(1)
10    print("LED off")
11    led.off()
12    sleep(1)
13

Shell x
>>> %Run -c $EDITOR_CONTENT
LED on
LED off
LED on
LED off
LED on
LED off

Plotter visualizes series of
numbers printed to the Shell.

See Help for details.

MicroPython (Raspberry Pi Pico) • COM11
```

Rysunek 4: Przykładowy program w środowisku Thonny

## 5 Zadania do wykonania

Każde z zadań będzie składać się z trzech części. Dotyczyć będą one:

obsługa wbudowanej diody LED za pomocą serwera HTTP, obsługi zewnętrznego modułu, uruchomienie serwera HTTP na RPI Pico do obsługi modułu,

W trakcie laboratoriów będzie należało uruchomić przynajmniej jeden z modułów:

czujnik natężenia światła BH1750 komunikujący się po magistrali I2C, wyświetlacz OLED oparty o sterownik SSD1306, dalmierz ultradźwiękowy HC-SR04, programowalne diody LED WS2812

Nazwa sieci oraz hasło do niej zostaną podane na zajęciach.

Uwaga!

W przypadku występowania błędu `OSError: [Errno 98] EADDRINUSE` należy płytkę odłączyć i podłączyć ponownie. Jednakże, aby problem nie występował ponownie warto ustawić odpowiednie opcje dla gniazdka *socket*, które zostało stworzone

```
1 s = socket.socket()
2 s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
```

Spowoduje to, że do wcześniej wykorzystanego adresu będzie można ponownie wymusić przypisanie tego samego portu.

Uwaga!

Próbując podłączyć się do płytki Raspberry Pi Pico i uruchomionego na niej serwera HTTP należy zwrócić szczególną uwagę na adres jaki będzie wpisywany w przeglądarce. Powinien być wykorzystywany protokół HTTP zamiast HTTPS.

Zatem w przeglądarce należy wpisywać adres URL, który jest podobny do:

```
1 http://xxx.yyy.zzz.qqq/
```

### 5.1 Dostępne czujniki i moduły

#### 5.1.1 Wbudowana dioda LED

W przypadku płytek Raspberry Pi Pico W dioda LED obecna na płytce nie jest dostępna pod pinem numer 25. Należy nią sterować podając jako parametr łańcuch znaków „LED”.

```
1 led = Pin("LED", Pin.OUT, value = 0)
```

Aby uruchomić serwer HTTP, który pozwoli na zmianę stanu diody LED należy wykorzystać materiały dostępne w Connecting to the Internet with Raspberry Pi Pico W [2]. Dane potrzebne do podłączenia się do lokalnej sieci WLAN zostaną podane na zajęciach.

#### 5.1.2 Czujnik światła BH1750

Uruchomienie czujnika natężenie światła BH1750. Należy skonfigurować magistralę I2C na pinach GP2 i GP3, a prędkość transmisji ustawić na 400kHz. W kolejnym kroku należy wyszukać dostępne urządzenia na magistrali I2C. Pozwoli to na pozyskanie adresu urządzenia podpiętego do magistrali. Następnie należy rozpocząć odczyty pomiaru natężenia światła wyrażone w luxach.

W ramach serwera HTTP należy utworzyć prostą stronę HTTP, którą będzie zawierać aktualną wartość wykonanego pomiaru.

#### 5.1.3 Wyświetlacz OLED SSD1306

Przed przystąpieniem do ćwiczenia należy pobrać i zainstalować bibliotekę SSD1306. Można to zrobić przez wybranie plug-ins i wyszukanie `micropython-ssd1306`. Do komunikacji z wyświetlaczem zostanie wykorzystana magistrala I2C na pinach GP2 i GP3, a częstotliwość transmisji powinna zostać ustawiona na 400kHz. Następnie na ekranie należy wyświetlić napis np. „Hello World! xxx”, gdzie xxx to inkrementowana liczba co jedną sekundę.

W przypadku problemów z instalacją biblioteki należy wykonać następujące kroki:

Ściągnąć plik `ssd1306.py` z repozytorium `micropython-ssd1306`. Stworzyć nowy plik w środowisku Thonny. Zapisać nowo utworzony plik na urządzeniu Raspberry Pi Pico pod nazwą `ssd1306.py`. Można używać biblioteki do obsługi wyświetlacza. Stworzyć nowy plik, w którym będzie wykonane ćwiczenie. Pozwoli to na uniknięcie problemu z nadpisywaniem pliku biblioteki.

W drugiej części ćwiczenia należy stworzyć serwer HTTP, do którego przesyłając tekst zostanie on następnie wyświetlony na ekranie.

#### 5.1.4 Dalmierz ultradźwiękowy HC-SR04

Do wykonania pomiaru odległości należy wykorzystać dwa dowolne piny mikrokontrolera RP2040. Jeden z pinów wyzwala pomiar, natomiast drugi pin w momencie osiągnięcia stanu wysokiego informuje o wykryciu odbitej fali dźwiękowej. Czas pomiędzy wyzwoleniem, a wykryciem powrotnej fali dźwiękowej pozwala na określenie odległości. Odległość powinna być mierzona w centymetrach.

Serwer HTTP powinien wyświetlać zmierzoną odległość. Strona powinna odświeżać się co 2 sekundy. Do tego celu można wykorzystać fragment kodu napisany w JavaScript.

#### 5.1.5 Programowalne diody LED WS2812

W ramach ćwiczenia należy stworzyć aplikację, która będzie zapalać diody WS2812 w określonym kolorze i z danym natężeniem. Do tego celu należy wykorzystać bibliotekę `neopixel`. Nie jest wymagane jej instalowanie, ponieważ jest ona bezpośrednio dostępna w firmware.

Strona internetowa powinna pozwalać na zdalne wybranie diody, której stan ma być zmodyfikowany, zmianę koloru w formacie RGB oraz poziomu jasności (od 0 do 1).

## Literatura

- [1] G. Halfacree, B. Everard. *Get Started with MicroPython on Raspberry Pi Pico*. Raspberry Pi Press 2021. <https://magpi.raspberrypi.com/books/micropython-pico>.
- [2] Raspberry Pi. *Connecting to the Internet with Raspberry Pi Pico W-series. Getting online with C/C++ or MicroPython on W-series devices*. Raspberry Pi Ltd. 2024. <https://datasheets.raspberrypi.com/picow/connecting-to-the-internet-with-pico-w.pdf>.
- [3] Raspberry Pi. Pico-series Microcontrollers. <https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html>, 2024.
- [4] Raspberry Pi. *Raspberry Pi Pico-series Python SDK A MicroPython environment for Raspberry Pi microcontrollers*. Raspberry Pi Ltd. 2024. <https://datasheets.raspberrypi.com/pico/raspberry-pi-pico-python-sdk.pdf>.
- [5] Raspberry Pi. Raspberry Pi Pico W UF2 file (latest). <https://micropython.org/download/rp2-pico-w/rp2-pico-w-latest.uf2>, 2024.