



**Matematyczne echa
lessowych wąwozów**



Zobacz matematykę

Robotyka z Arduino Alvik

Małgorzata Cudna
UMCS 2025

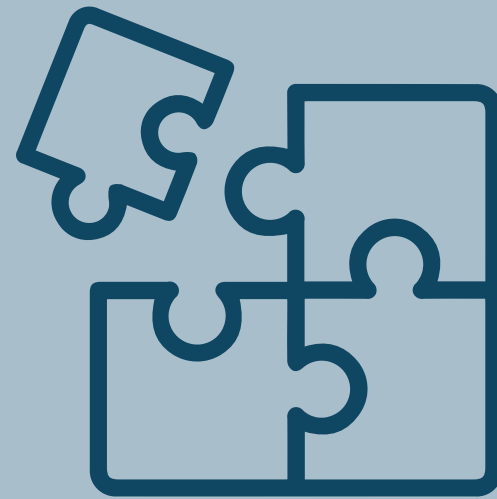


Co dziś robimy?



POZNAJEMY ROBOTA

- Testujemy trzy wbudowane programy



PROGRAMUJEMY

- Budujemy własne programy w języku MicroPython




TESTUJEMY

- Uruchamiamy i testujemy programy, sterując robotem

Programy testowe





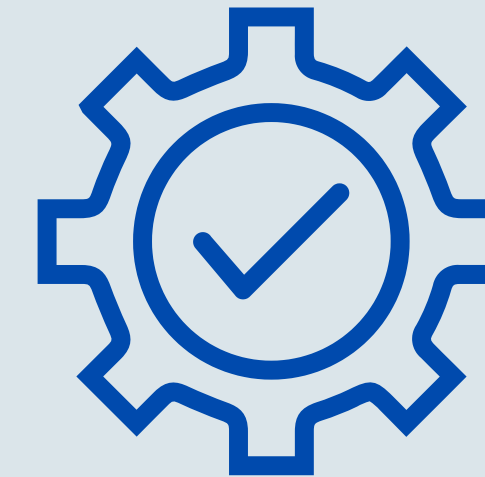
CZERWONY

- Programujemy ruch naciskając odpowiednie strzałki
- Potwierdzamy 





ZIELONY

- Robot utrzymuje odległość od obiektu z przodu
- Start 
- Koniec 



NIEBIESKI

- Robot porusza się wzdłuż czarnej linii o szer. 2-3 cm
- Start 
- Koniec 

Program 1: mrugnij!

Chrome: lab-micropython.arduino.cc

```
from arduino import *  
from arduino_alvik import ArduinoAlvik  
  
alvik = ArduinoAlvik()
```

Na początku programu importujemy potrzebne biblioteki i tworzymy obiekt, który będzie kontrolować robota i jego elementy



Program 1: mrugnij!

```
def setup():  
    alvik.begin()  
  
def loop():  
    print('mrugam...')  
    delay(1000)  
  
def cleanup():  
    alvik.stop()  
  
start(setup, loop, cleanup)
```



Definiujemy trzy funkcje:

- setup uruchomi się raz na początku
- loop będzie działać w pętli
- cleanup zostanie wywołana raz na końcu działania programu



Program 1: mrugnij!

```
def loop():  
    print('mrugam...')  
    alvik.left_led.set_color(1, 0, 0)  
    alvik.right_led.set_color(1, 0, 0)  
    delay(1000)  
    alvik.left_led.set_color(0, 0, 0)  
    alvik.right_led.set_color(0, 0, 0)  
    delay(1000)
```

W funkcji loop dodajemy instrukcje, które będą ustawiały kolor obu światełek LED umieszczonych na górze robota. Kolory definiujemy w modelu RGB w zakresie 0-1.

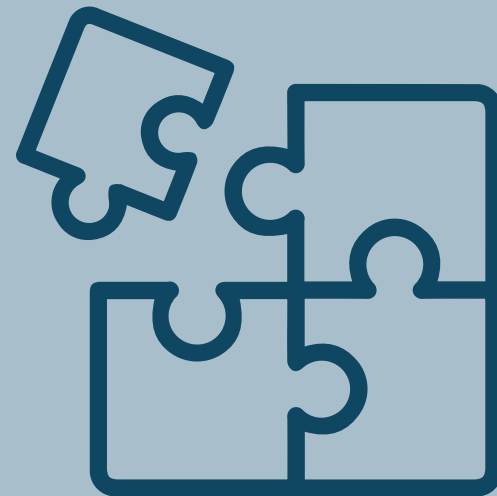


Testujemy!



Przetestuj program

- Czy światełka robota mrugają?
- Jaka jest przerwa między błyskami świateł?



Modyfikuj

- Spróbuj zmodyfikować kod programu, aby światełka mrugały naprzemiennie na niebiesko i zielono



Eksperymentuj

- Spróbuj stworzyć swój własny wielokolorowy wzór mrugających światełek :)

Program 2: rusz się!

```
def loop():  
    alvik.move(1)  
    delay(500)  
  
    alvik.rotate(45)  
    delay(20)  
    alvik.rotate(-45)  
    delay(20)
```

W funkcji loop mamy:

- jazdę do przodu
- skręty
- Jakie kąty skręcają w lewo a jakie w prawo?
- Jak można pojechać do tyłu?

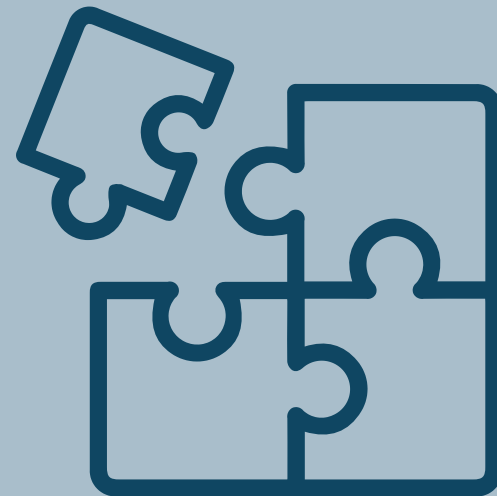


Testujemy!



Przetestuj program

- Podnieś robota i zastopuj program, jeśli chciałby uciec i spaść!



Modyfikuj

- Aby program zadziałał po odłączeniu od komputera, zmień nazwę pliku importowanego w **main.py**



Eksperymentuj

- Wyłącz robota, połóż go na podłodze, włącz i gotowe!

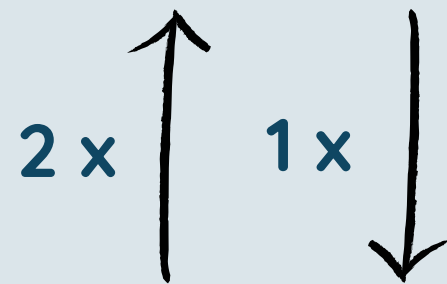
Program: dziwne kroki

Zaprogramuj krok robota godny **Ministerstwa Dziwnych Kroków**
w hołdzie dla **Latającego Cyrku Monty Pythona!**



Program: dziwne kroki

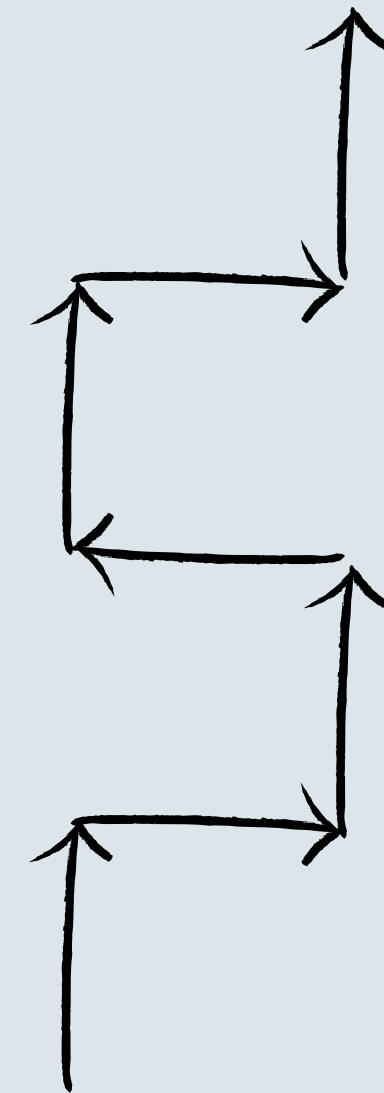
Krok 1



Krok 2

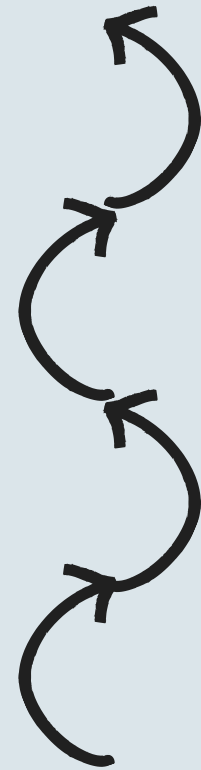


Krok 3



Program: dziwne kroki

Krok 4



Krok 5



Eksperymentuj

- A jaki byłby Twój własny dziwny krok? :)

Program: dziwne kroki

```
alvik.move(distance_in_cm)
alvik.rotate(angle_in_deg) # 90 stopni to skręt w lewo!

alvik.set_wheels_speed(left_rpm, right_rpm)

alvik.get_touch_left() # right, up, down, ok, center
alvik.on_touch_cancel_pressed(alvik.stop)
```

W naszym programie
można dodać obsługę
przycisków, aby
sterować
startem/stopem
programu



Program: dziwne kroki (krok 1)

```
from arduino import *  
from arduino_alvik import ArduinoAlvik
```

```
alvik = ArduinoAlvik()
```

```
def setup():  
    alvik.begin()  
    while not alvik.get_touch_ok():  
        delay(100)
```

```
def cleanup():  
    alvik.stop()
```

```
def loop():  
    alvik.move(10)  
    delay(20)  
    alvik.move(10)  
    delay(20)  
    alvik.move(-10)  
    delay(20)
```


```
alvik.on_touch_cancel_pressed(cleanup)  
start(setup, loop, cleanup)
```

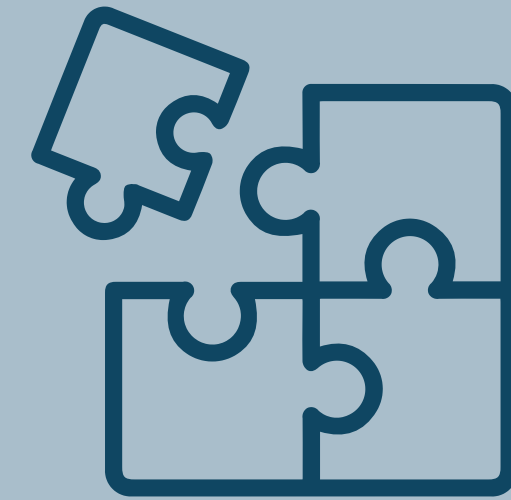
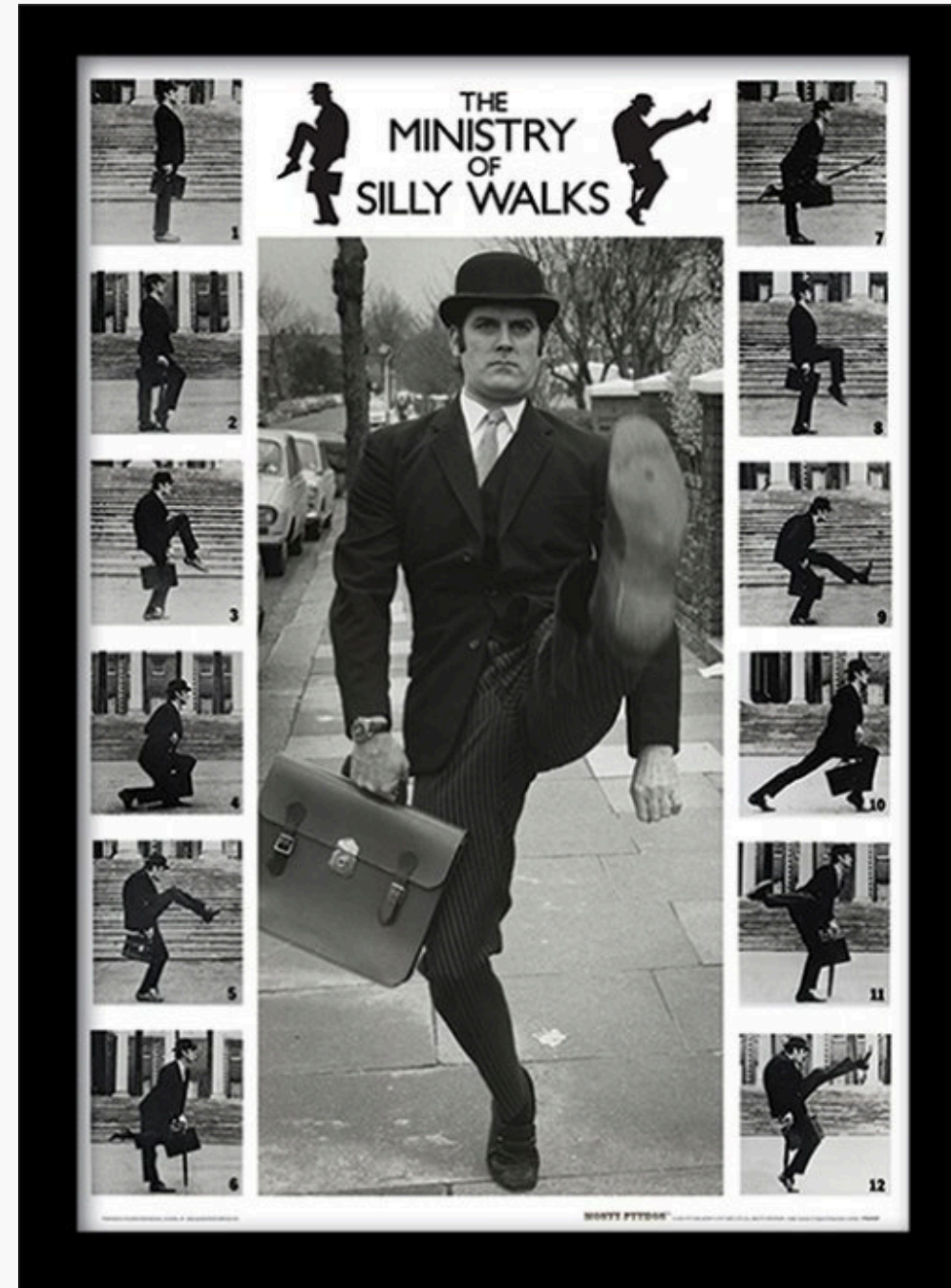


Testujemy!



Przetestuj program

- Czasami przycisk stopu  trzeba nacisnąć parę razy, zanim zadziała, dlaczego?



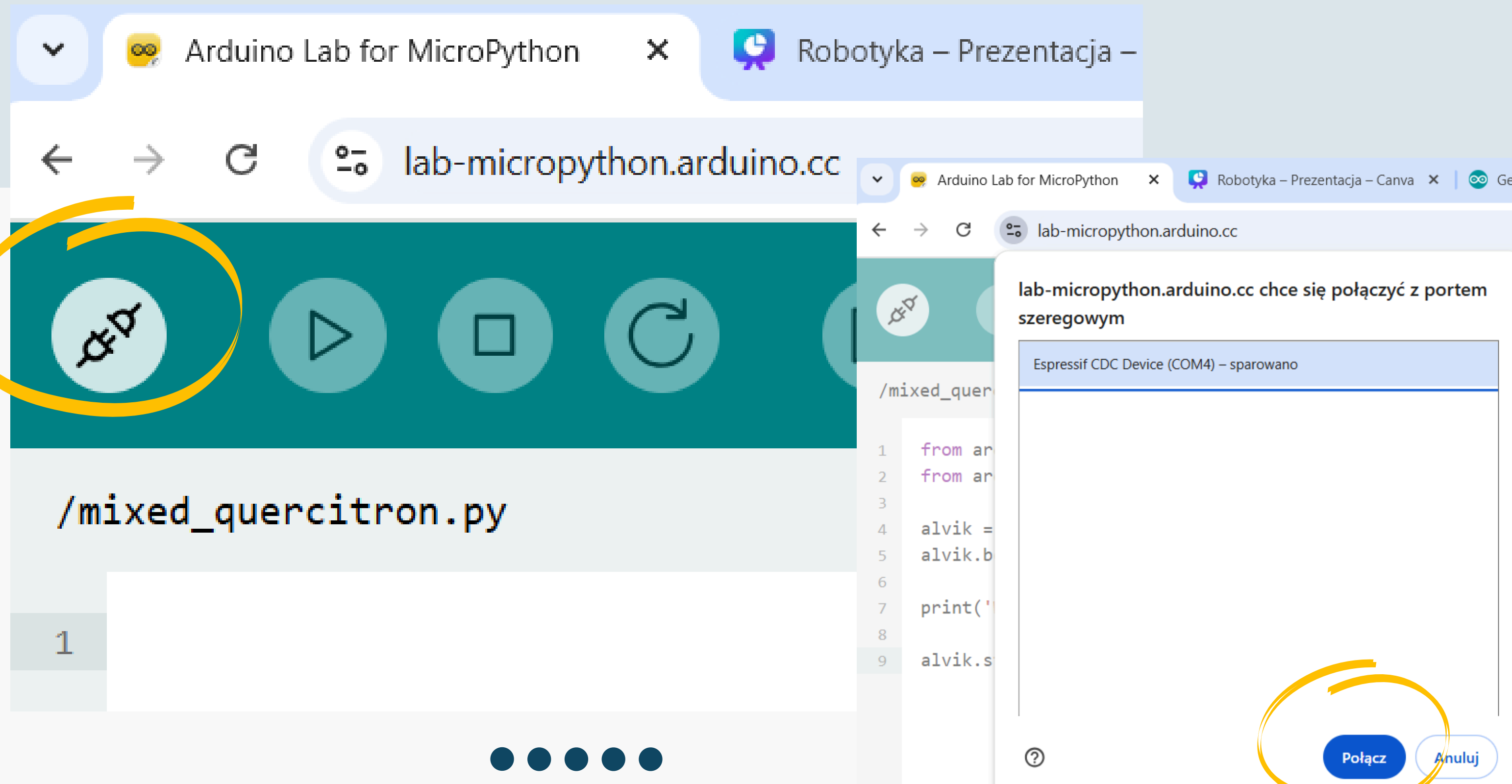
Modyfikuj i eksperymentuj

- Stwórz własny dziwny krok dla robota!



MicroPython online 1

Po podłączeniu Alvika do USB naciskamy ikonę **connect**, a następnie wybieramy **Espressif Device**



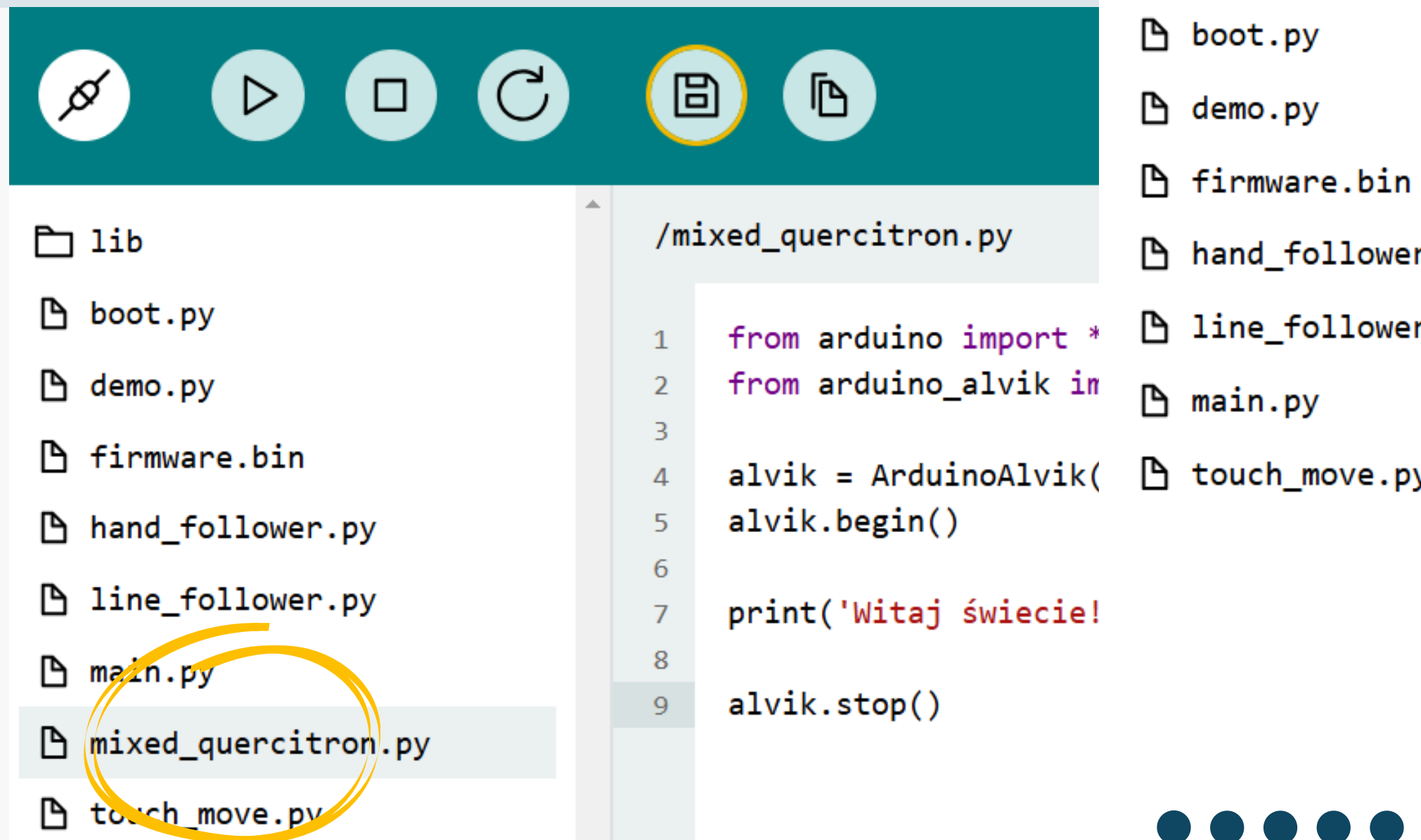
The screenshot shows the Arduino Lab for MicroPython web interface. The browser tabs include "Arduino Lab for MicroPython" and "Robotyka – Prezentacja –". The address bar shows "lab-micropython.arduino.cc". The main interface has a teal header with several icons: a "connect" icon (a USB symbol) circled in yellow, a play button, a square button, and a refresh button. Below the header, the file path "/mixed_quercitron.py" is displayed. A code editor shows the following Python code:

```
1 from ar
2 from ar
3
4 alvik =
5 alvik.b
6
7 print('
8
9 alvik.s
```

On the right side, a dialog box is open with the title "lab-micropython.arduino.cc chce się połączyć z portem szeregowym". It lists the selected device: "Espressif CDC Device (COM4) – sparowano". At the bottom of the dialog, the "Połącz" (Connect) button is circled in yellow, along with an "Anuluj" (Cancel) button.

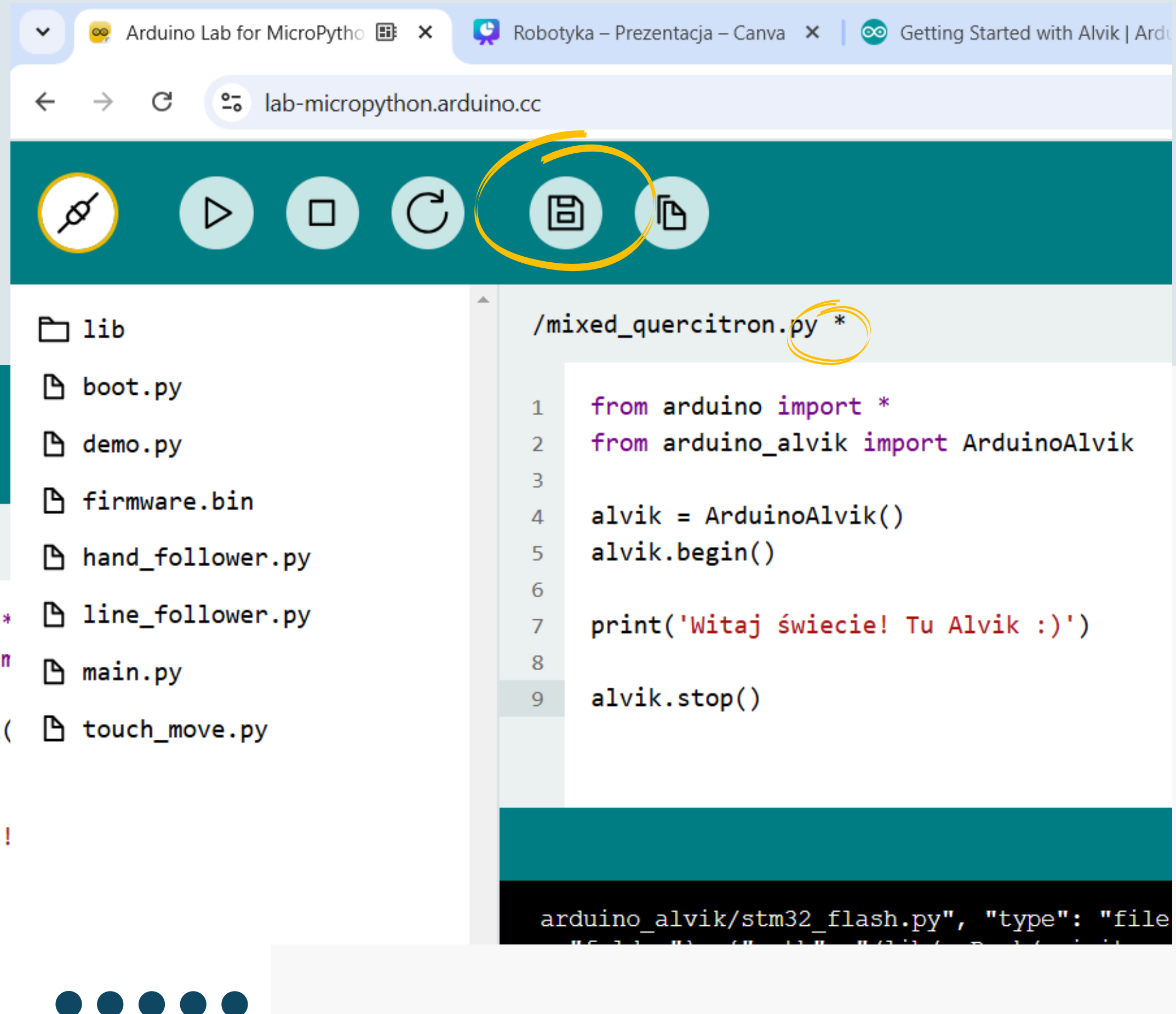
MicroPython online 2

Edytujemy kod programu i zapisujemy zmiany **save**. Uaktualni się lista plików.



```
lib
boot.py
demo.py
firmware.bin
hand_follower.py
line_follower.py
main.py
mixed_quercitron.py
touch_move.py
```

```
/mixed_quercitron.py
1 from arduino import *
2 from arduino_alvik in
3
4 alvik = ArduinoAlvik(
5 alvik.begin()
6
7 print('Witaj świecie!
8
9 alvik.stop()
```



```
lib
boot.py
demo.py
firmware.bin
hand_follower.py
line_follower.py
main.py
touch_move.py
```

```
/mixed_quercitron.py *
1 from arduino import *
2 from arduino_alvik import ArduinoAlvik
3
4 alvik = ArduinoAlvik()
5 alvik.begin()
6
7 print('Witaj świecie! Tu Alvik :)')
8
9 alvik.stop()
```

MicroPython online 3

The screenshot displays the MicroPython online IDE interface. The top toolbar features several icons: a play button (run), a square (stop), a circular arrow (refresh), a floppy disk (save), and a USB symbol (upload). The left sidebar shows a file explorer with a 'lib' folder and several Python files, including 'mixed_quercitron.py'. The main editor shows Python code for initializing an ArduinoAlvik object and printing a message. Two terminal windows are overlaid: the top one shows a warning to turn on the Arduino Alvik, and the bottom one shows the successful execution of the code, printing 'Witaj świecie! Tu Alvik :)'. Yellow circles highlight the run icon in the toolbar and the terminal output.

```
1 from arduino import *
2 from arduino_alvik import ArduinoAlvik
3
4 alvik = ArduinoAlvik()
5 alvik.begin()
6
7 print("Witaj świecie! Tu Alvik :)")
8
9 alvik>
```

MicroPython v1.22.0 on 2024-02-22; Arduino Nano ESP32 with ESP32S3
Type "help()" for more information.
>>>

Uruchamiamy program ikoną **run** i czytamy komunikaty terminala. Po włączeniu Alvika zobaczymy komunikat funkcji **print**.

Programy działające dłużej możemy zatrzymać ikoną **stop** (jest obok run).





Świetna robota!

- <https://docs.arduino.cc/tutorials/alvik/getting-started/>
- <https://lab-micropython.arduino.cc/>
- <https://courses.arduino.cc/explore-robotics-micropython/>
- <https://docs.arduino.cc/tutorials/alvik/api-overview/>

- Obrazek 1: icelandnews.is
- Obrazek 2: cdn.europosters.eu





**DOFINANSOWANO
ZE ŚRODKÓW
BUDŻETU PAŃSTWA**

W RAMACH PROGRAMU
Społeczna odpowiedzialność nauki II
projekt: Matematyczne echa lessowych
wąwozów

DOFINANSOWANIE
[125 730 zł]

CAŁKOWITA WARTOŚĆ
[139 700 zł]

DATA PODPISANIA UMOWY
[08.2023]

