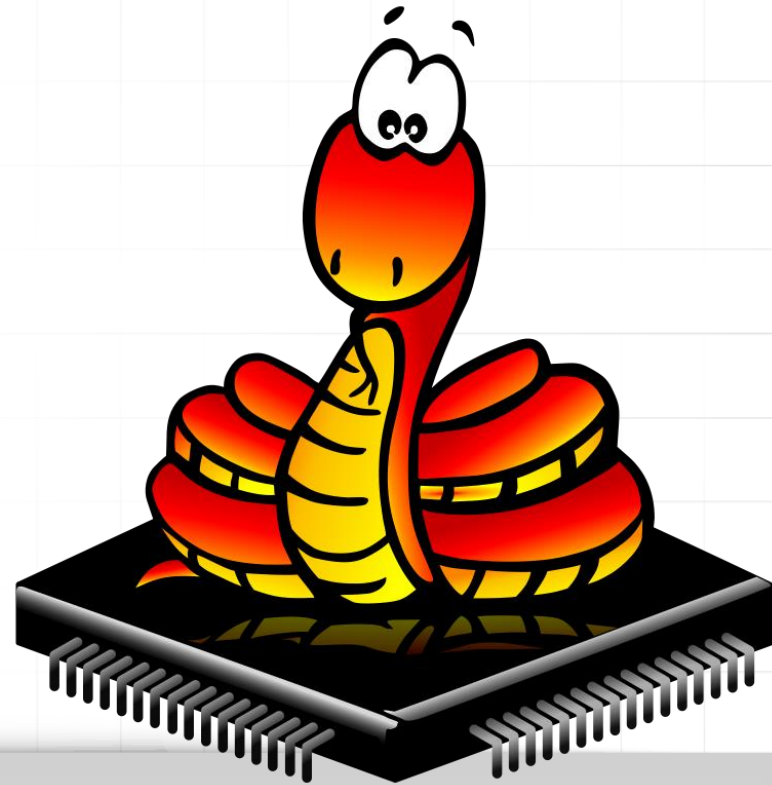


# Wielozadaniowość w języku MicroPython

Systemy operacyjne mikrokontrolerów

Agata Lebek  
Artur Bogacz  
Jacek Jaszcuk  
Grzegorz Woźny



HR EXCELLENCE IN RESEARCH



Politechnika Wroclawska

# Plan prezentacji

- **Cel i założenia projektu**
- **Definicja współprogramu**
- **Wykorzystane elementy**
- **Oprogramowanie**
- **Efekt końcowy**



HR EXCELLENCE IN RESEARCH



Politechnika Wroclawska

# Cel projektu

- Sterowanie w czasie rzeczywistym silnikiem krokowym poprzez stronę WWW
- Wykorzystanie mechanizmu współprogramów (coroutines)



# Założenia projektu

- Stworzenie dwóch współprogramów. Program pierwszy wystawia stronę WWW oraz zarządza obsługą klienta WWW. Program drugi kręci silnikiem krokowym.
- Komunikacja między programami przy użyciu zmiennych globalnych. Program WWW informuje silnik o kierunku obrotu. Na stronie WWW znajdą się dwa odnośniki, dzięki którym będzie można ustawić kierunek kręcenia się silnika.
- Możliwe będzie również określenie prędkości i liczby kroków (zarówno wczytywanie z pliku, jak i wpisywanie parametrów na stronie WWW).

# Współprogram w Pythonie

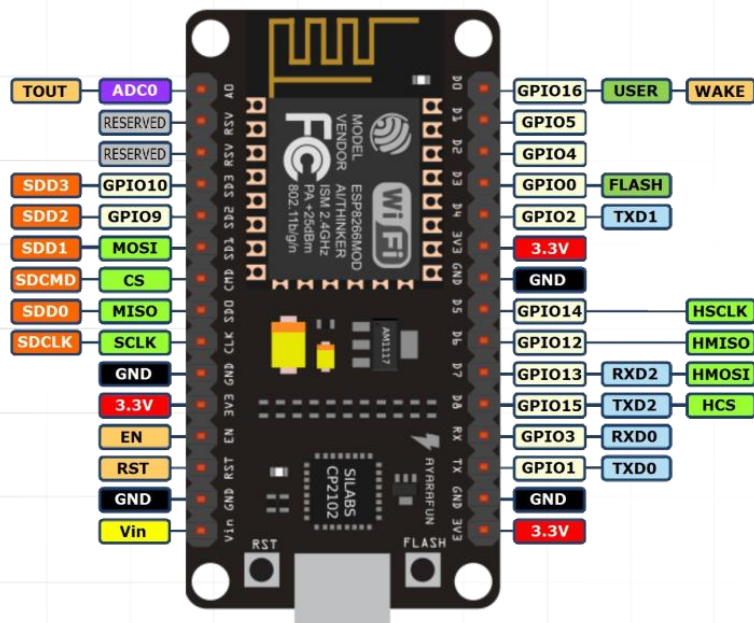
- Współprogram cechuje się posiadaniem ciągu instrukcji do wykonania i ponadto możliwością zawieszania wykonywania jednego współprogramu *A* i przenoszenia wykonywania do innego współprogramu *B*.
- W języku Python współprogram jest uogólnieniem podprogramu. Jest on używany do wielozadaniowej współpracy, w której proces dobrowolnie oddaje kontrolę okresowo aby umożliwić jednoczesne uruchamianie wielu aplikacji.
- Współprogram może zawiesić swoje wykonanie i przekazać kontrolę do innego modułu i może wznowić wykonywanie od miejsca, w którym zostało przerwane.

# Wykorzystane elementy

- Mikrokontroler z WiFi ESP8266 V3 NodeMCU
- Silnik krokowy 28BYJ-48 5VDC + sterownik ULN2003



# Mikrokontroler z WiFi ESP8266 V3 NodeMCU



Specyfikacja modułu NodeMCU:

Moduł zbudowany w oparciu o układ ESP8266-12E

Zasilanie napięciem 3,3V (lub 5V przez USB)

Zawiera konwerter USB-UART – CH340

10 portów GPIO (PWM, I2C, SPI, 1-Wire)

Konwerter ADC – 10-bitowy

Złącze micro USB

# Silnik krokowy 28BYJ-48 i sterownik ULN2003

## Specyfikacja silnika

- Napięcie znamionowe: 5 V
- Pobór prądu na cewkę: 100 mA
- Przełożenie: 64:1
- Rezystancja cewki: 50  $\Omega$
- Moment trzymający na wyjściu przekładni: 0,03 Nm
- Pięć wyprowadzeń
- Wymiary:  $\phi 28 \times 19$  mm (bez wału)

## Specyfikacja sterownika

- Napięcie pracy: 5 V
- Diody LED wskazujące aktualną fazę
- Siedem pinów wejściowych - IN1...IN7
- Wymiary płytki: 35 x 32 mm





# Oprogramowanie

## Front-end

```
function funName(argument){  
    let req = new XMLHttpRequest();  
    req.open("POST", "tu_slij", true);  
    var mySuperVar = "Hello world 2.0";  
    //...  
    req.send(JSON.stringify({myTxt: mySuperVar}))  
}}
```

# Oprogramowanie

## Back-end – Tworzenie tasków

```
loop = asyncio.get_event_loop()
loop.create_task(motor_run())
loop.create_task(web_serv_run())
loop.run_forever()
```

# Oprogramowanie

## Back-end – Funkcja await

```
async def motor_step_right(d1, d2, d3, d4):  
    d4.off()  
    d3.on()  
    await asyncio.sleep(motor_sleep_time)  
    d3.off()  
    d2.on()  
    await asyncio.sleep(motor_sleep_time)  
    d2.off()  
    d1.on()  
    await asyncio.sleep(motor_sleep_time)  
    d1.off()  
    d4.on()  
    await asyncio.sleep(motor_sleep_time)
```

# Efekt końcowy – front – end strony WWW

## Panel kontrolny

### Zasilanie

Wyłącz  Włącz

### Kierunek

LEWO

PRAWO

### Prędkość

Wartość: 68

## Preset

### Wgraj plik:

PLIK \*.AJAG

### Wprowadź nastawy manualnie:

przykład: silniki(kierunek, liczba kroków, prędkość)

SUBMIT

# Panel kontrolny

## Zasilanie

Wyłącz ☒ Włącz

## Kierunek

LEWO

PRAWO

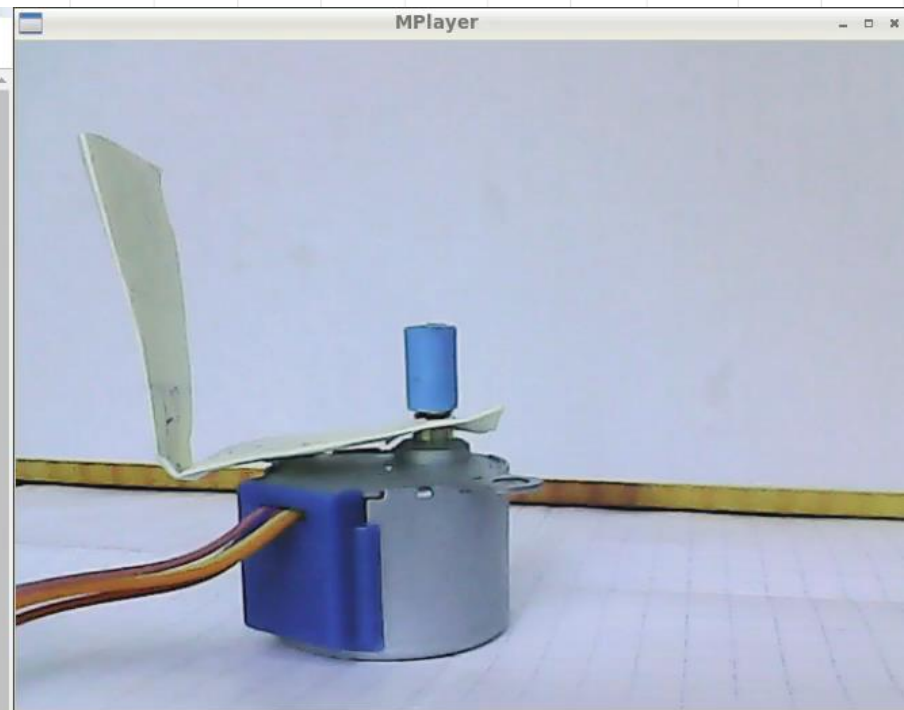
## Prędkość

95

## Preset

Wgraj plik:

PLIK \*.AJAG



```
miniterm.py /home/padawan
Plik Edycja Karty Pomoc
(X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103
Safari/537.36 OPR/60.0.3255.109\r\nContent-Type: text/plain;charset=UTF-8\r\nAc
cept: */*\r\nReferer: http://192.168.4.1:8080/\r\nAccept-Encoding: gzip, deflate
\r\nAccept-Language: pl-PL,pl;q=0.9,en-US;q=0.8,en;q=0.7\r\n\r\n{"power":false}'
Ilość odebranych bajtów to: 453
Jest JSON!
Twój JSON: b'{"power":false}'
Silnik włączony: False
Zamykamy już to połączenie!
Mamy połączenie od: {} ('192.168.4.2', 56144)
Ile: 452
b'POST /tu_slij HTTP/1.1\r\nHost: 192.168.4.1:8080\r\nConnection: keep-alive\r\n
Content-Length: 14\r\nOrigin: http://192.168.4.1:8080\r\nUser-Agent: Mozilla/5.0
(X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103
Safari/537.36 OPR/60.0.3255.109\r\nContent-Type: text/plain;charset=UTF-8\r\nAc
cept: */*\r\nReferer: http://192.168.4.1:8080/\r\nAccept-Encoding: gzip, deflate
\r\nAccept-Language: pl-PL,pl;q=0.9,en-US;q=0.8,en;q=0.7\r\n\r\n{"power":true}'
Ilość odebranych bajtów to: 452
Jest JSON!
Twój JSON: b'{"power":true}'
Silnik włączony: True
Zamykamy już to połączenie!
```

# Bibliografia

<https://docs.python.org/3/library/asyncio-task.html>

<https://github.com/peterhinch/micropython-async/blob/master/TUTORIAL.md>

<https://www.geeksforgeeks.org/coroutine-in-python/>

<https://botland.com.pl/pl/moduly-wifi/8241-modul-wifi-esp8266-nodemcu-v3.html>

<https://botland.com.pl/pl/silniki-krokowe/3480-silnik-krokowy-z-przekladnia-28byj-48-5v-01a-003nm-ze-sterownikiem-uln2003-5903351241458.html>