

# Moduł ESP8266 + MicroPython

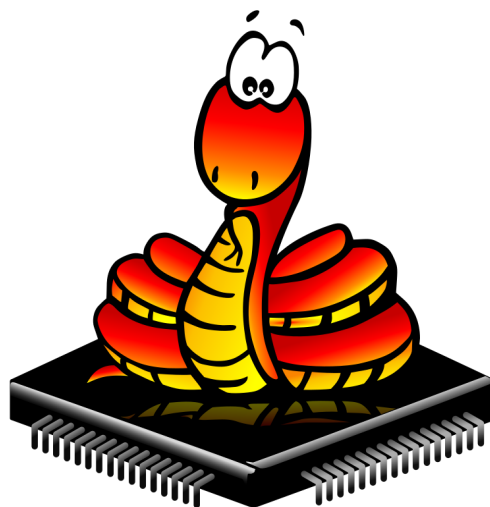
## 1 O module

Moduł zawiera mikrokontroler ESP8266, 4MB pamięci Flash, układ CH340G odpowiedzialny za komunikację UART poprzez USB, stabilizator 3.3V umożliwiający zasilanie z portu USB.

## 2 O ESP8266

ESP jest mikrokontrolerem zawierającym wbudowany układ WiFi oraz pełny stos TCP/IP dedykowanym do zastosowań IoT. Układ posiada 1 kanałowy 10 bitowy przetwornik analogowo-cyfrowego, interfejs SPI oraz 13 pinów GPIO dostępnych dla użytkownika, umożliwiających m.in. realizację PWM, komunikację z użyciem UART, a programowo także m.in. I2C i 1Wire.

*UWAGA: Wejścia układu działają z logiką 3.3V. Podanie wyższego napięcia może doprowadzić do uszkodzenia układu.*



## 3 O MicroPythonie

MicroPython jest implementacją Pythona napisaną w C z przeznaczeniem do uruchamiania bezpośrednio na mikrokontrolerach i udostępnianą na zasadach licencji typu MIT.

## 4 Podłączenie z układem

W pamięci flash układu zainstalowany jest MicroPython w wersji 1.9.4, wystawia on interaktywny interpreter na porcie szeregowym udostępnianym poprzez USB. Port działa z prędkością 115200 bodów, w trybie bez parzystości, 8 bitów danych, 1 bitu stopu.

### 4.1 Linux

W przypadku systemów Linux po podłączeniu układu zostanie on automatycznie wykryty jako port szeregowy i zostanie utworzone odpowiednie urządzenie - najczęściej `/dev/ttyUSB0`, ale w zależności od konfiguracji danego systemu może mieć inną nazwę. Celem nawiązania z nim komunikacji należy uruchomić emulator terminalu portu szeregowego np.: `sudo picocom -b 115200 /dev/ttyUSB0`

### 4.2 Windows

W przypadku systemów Windows konieczne może być zainstalowanie sterowników (<https://sparks.gogo.co.nz/ch340.html>). W roli emulatora portu szeregowego można użyć putty (<https://www.putty.org/>), należy wybrać typ połączenia serial, wskazać właściwy port szeregowy i ustawić odpowiednią prędkość (115200).

## 5 Przykładowy kod

Poniżej zamieszczamy przykład kodu który:

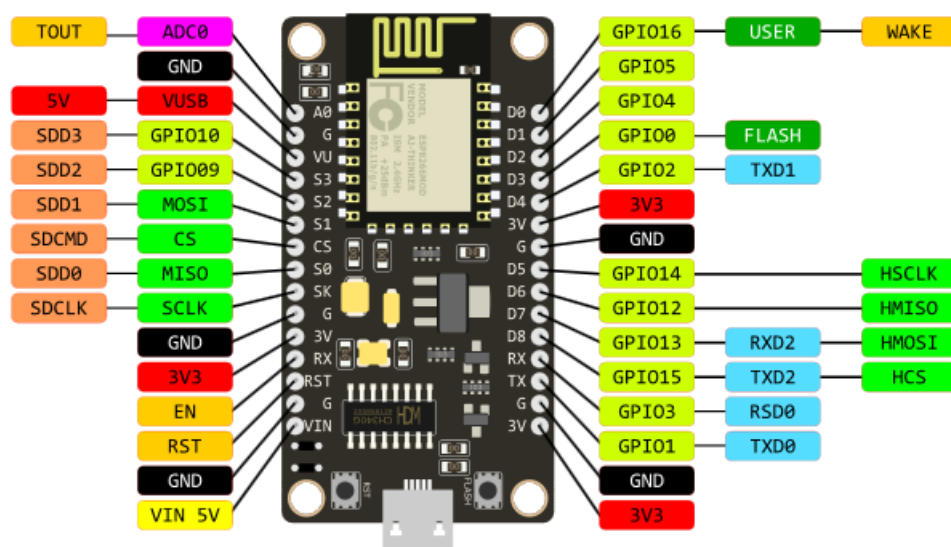
- konfiguruje pin 16 jako wyjście i pobiera obiekt go reprezentujący
- w nieskończonej pętli zmienia jego stan na przeciwny odczekując pomiędzy zmianami 1s

Efektom będzie miganie umieszczonej na module diody świecącej podłączonej właśnie do pinu 16.

```
import machine
p = machine.Pin(16, machine.Pin.OUT)
import time
while True:
    p.value(int(a))
    a = not a
    time.sleep(1)
```

Działanie programu można przerwać przy pomocy kombinacji *Ctrl+C*.

## 6 Układ wyprowadzeń modułu



## 7 Informacje dodatkowe

Dokumentacja MicroPython 1.9.4 dla ESP8266: <https://docs.micropython.org/en/latest/esp8266/index.html>

Aktualizacja MicroPythona lub zastąpienie go innym (także własnym) programem możliwe przy pomocy narzędzia *esptool*: <https://github.com/espressif/esptool>.

*Uwaga: płytka wymaga opcji `-fm dio` przy poleceniu `write_flash`.*

```
sudo ./esptool.py --port /dev/ttyUSB0 erase_flash
sudo ./esptool.py --port /dev/ttyUSB0 write_flash -fm dio \
    --flash_size=detect 0 image.bin
```