

# ESP8266 & MicroPython

Karolina Włodarska

Adrian Nowak

Patryk Jarosz

1. Celem ćwiczenia jest pobieranie czasu z serwera NTP i udostępnianie go przez UART. Wykorzystywane będą ESP8266 oraz MicroPython.

2. Pinout płytki wygląda następująco:

```
+-----+
| 1 2   |
| 3 4   |
| 5 6   |
| 7 8   |
+-----+
```

1 - TX0 (debugowy)

2 - GND

3 - CH\_PD (zwarne do VCC; moduł zawsze włączony)

4 - TX1 (nadawanie danych do urządzenia docelowego)

5 - RST (przycisk do masy; resetowanie urządzenia)

6 - GPIO0 (pullup + zworka do GND; domyślnie start z flasha, gdy zwarta to bootloader)

7 - VCC

8 - RX0 (debugowy)

3. Instalacja MicroPythona

- a. Otwórz wiersz poleceń i przejdź do lokalizacji katalogu "esptool-master"
- b. Podepnij ESP8266 i sprawdź w menedżerze urządzeń na którym COM porcie jest Twoje urządzenie
- c. Wróć do wiersza poleceń, wpisz komendę:  
`>>esptool.py --port COMXX erase_flash`  
gdzie XX to numer portu, do którego podpięte jest ESP
- d. Wciśnij reset na płytce
- e. Wpisz komendę w wierszu poleceń:  
`>>esptool.py --port COMXX --baud 115200 write_flash --flash_size=detect 0 binarka_stara.bin`
- f. Wciśnij reset na płytce
- g. Po poprawnie wgraniu sofci uruchom terminal, ustaw odpowiedni numer portu oraz prędkość na 115200
- h. Wciśnij reset na płytce
- i. Uruchomił się MicroPython

#### 4. Instalacja webREPL

- a. W terminalu wpisz następujące komendy:  

```
>>import webrepl_setup  
>>E  
>>qwerty  
>>y
```

gdzie "qwerty" to hasło, można ustawić inne

#### 5. Podłączenie do sieci

- a. Sprawdź nazwę sieci i hasło, do której chcesz się zalogować swoim ESP
- b. W terminalu wpisz następujące komendy:  

```
>>import network  
>>sta = network.WLAN(network.STA_IF)  
>>sta.active(True)  
>>sta.scan()  
>>sta.connect("Nazwa_sieci", "Haslo_sieci")  
>>sta.isconnected()  
>>sta.ifconfig()
```
- c. Po udanym połączeniu terminal zwróci IP. Zapisz je, ponieważ przyda się ono w następnych krokach

6. W pliku boot.py znajdującym się w główny folderze z programami uzupełnij liniijkę 18 o poprawne dane logowania: nazwę sieci i hasło:  
`sta_if.connect('Nazwa_sieci','Haslo_sieci')`

#### 7. webREPL

- a. Wejdź w folder "webrepl-master" i uruchom plik "webrepl.html"
- b. W lewym górnym rogu REPL popraw w konfiguracji TYLKO adres IP (zamień je na to, które uzyskano w punkcie 5c)
- c. Kliknij Connect
- d. Podaj hasło, które skonfigurowane było w punkcie 4a
- e. W sekcji "Send a file" kliknij "Wybierz plik"
- f. Wybierz plik "boot.py" i kliknij "Send to device"
- g. Powtórz punkty e-f, tym razem wybierz plik "main.py"
- h. Kliknij "Disconnect"

W tym kroku należy wybrać pliki boot.py i main.py - ich funkcjonalność została opisana w punktach 8 i 9.

8. Plik boot.py przedstawiony na listingu poniżej uruchamia WEB REPL po starcie systemu i łączy się do wybranej sieci wi-fi (linijka 18 - w miejsce "SSID" i "password" należy wpisać odpowiednio nazwę i hasło do sieci, do której chcemy się łączyć). Jeżeli nie połączymy się w 10s to dostaniemy komunikat zwrotny "Błąd połączenia"

```

1  # This file is executed on every boot (including wake-boot from deepsleep)
2
3  import esp
4  esp.osdebug(None) #turn off vendor O/S debugging messages
5
6  #domyslnie
7  import gc
8  import webrepl
9  webrepl.start()
10 gc.collect()
11
12 #wylaczenie trybu AP i polaczenie z istniejaca siecia
13 print('\n\nLaczenie z siecia WiFi...')
14 import network, time, machine
15 network.WLAN(network.AP_IF).active(False)
16 sta_if = network.WLAN(network.STA_IF)
17 sta_if.active(True)
18 sta_if.connect('SSID','password')
19 count = 0
20 while not sta_if.isconnected():
21     time.sleep_ms(1)
22     count += 1
23     if count==10000:
24         print('Blad polaczenia.')
25         break
26 print('Konfiguracja: ', sta_if.ifconfig(), '\n')

```

9. Główna część pliku main.py została przedstawiona na listingu poniżej. Skrypt aktywuje dwa timery tim1 i tim2. Tim1 jest odpowiedzialny za pobieranie i synchronizację czasu poprzez protokół ntp z sieci, natomiast tim2 co sekundę wysyła komunikat o aktualnym czasie.

```

65 #####
66 print('main.py')
67
68 #timer do aktualizacji RTC z NTP - raz na sekunde
69 tim1 = machine.Timer(-1)
70 tim1.init(period=1000, mode=machine.Timer.PERIODIC, callback=updateTime_timer)
71 tim1_presc = 0
72 updateTime() #pierwsze uruchomienie, zeby nie czekac
73
74 #uart i timer do komunikacji z urzadzeniem zewnetrznym - raz na sekunde
75 uart = machine.UART(1, BAUDRATE, bits=8, parity=None, stop=1)
76 tim2 = machine.Timer(-1)
77 tim2.init(period=1000, mode=machine.Timer.PERIODIC, callback=sendTime_timer)
78 tim2_presc = 0
79 sendTime() #j/w

```

Apendix.

[1] - ogólna prezentacja

[https://docs.google.com/presentation/d/1I96YuwVjh\\_mJ1Mfw1itwwkOrpoRTR7DH3KRzApD\\_EALA/edit?usp=sharing](https://docs.google.com/presentation/d/1I96YuwVjh_mJ1Mfw1itwwkOrpoRTR7DH3KRzApD_EALA/edit?usp=sharing)