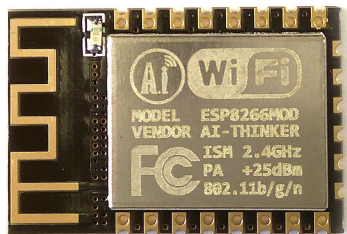
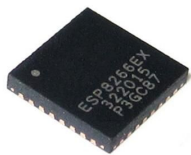
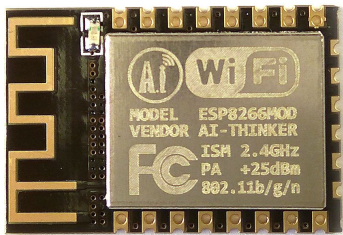
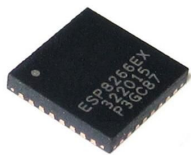


# ESP8266

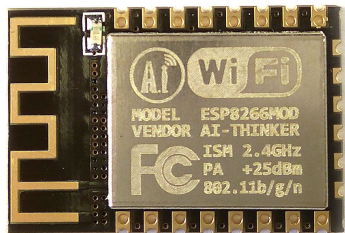
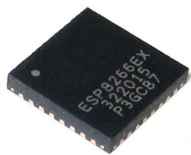




- ilość rdzeni: 1
- architektura: 32-bit
- taktowanie: 80 MHz (typowo) – 160 MHz (programowalne)
- pamięć: zewnętrzna
- pamięć flash: 512 KB – 4 MB (zależnie od wersji)
- poziomy logiczne: 3.6V DC (max)
- napięcie wejściowe: 2.5 – 3.6V DC



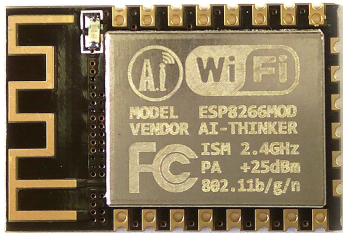
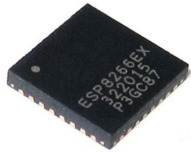
- zasilanie: GPIO pins  
(Vc & CH\_PD – both need to be connected)
- sieć: Wi-Fi (IEEE 802.11 b/g/n)  
(niedostępna w trybie uśpienia)
- antena - PCB
- standardy: FCC/CE/TELEC/SRRC
- zakres częstotliwości: 2.4G ~ 2.5G (2400 ~ 2483.5)



- security modes: WPA, WPA2
- szyfrowanie: WEP/TKIP/AES
- protokoły sieciowe: IPv4, TCP/UDP, DHCP
- Bluetooth: brak
- operating current - mode dependent:
  - tryby uśpienia: od 0.1 mA do 15 mA,
  - transmisja Rx/Tx (Wi-Fi) 80-170 mA,
  - maksymalnie 320 mA

# ESP8266

---

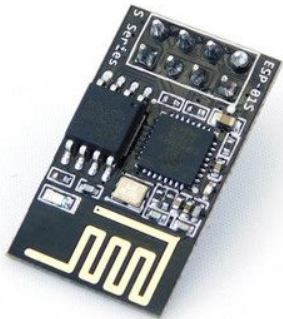


- wydajność prądowa I/O: 12 mA
- zakres temperatur pracy: -40 ~ 125 °C
- interfejsy: **Serial/UART**, SDIO, SPI

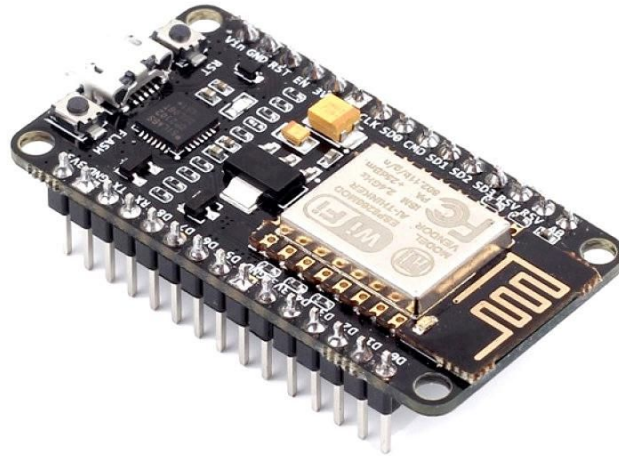


# ESP8266

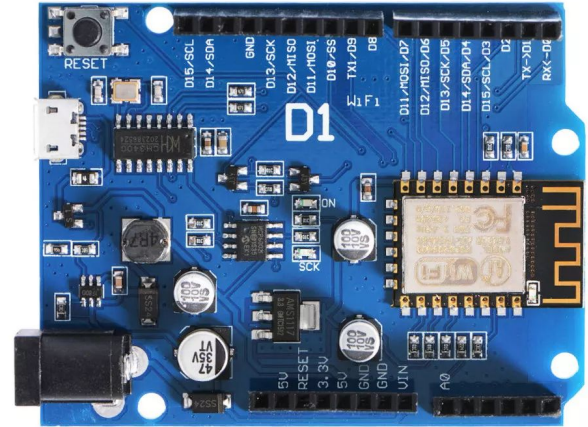
---



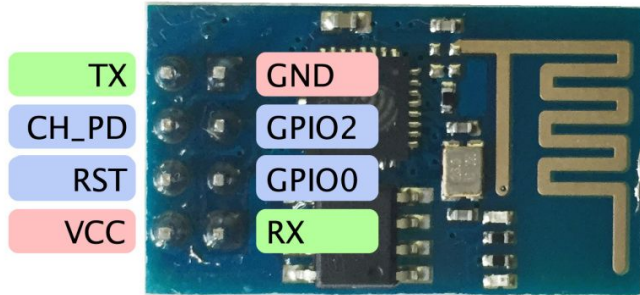
ESP8266-01



ESP8266-12  
NodeMCU v2



WeMos D1



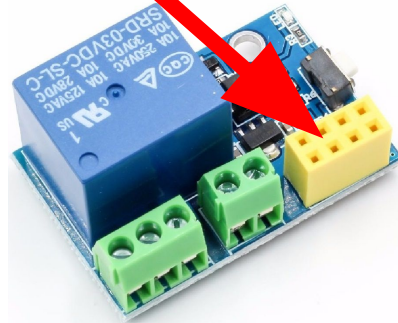
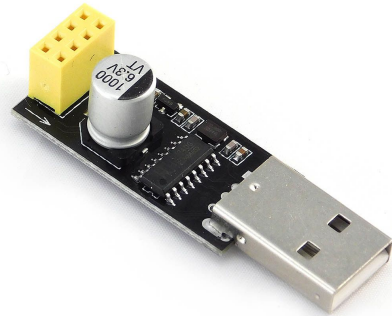
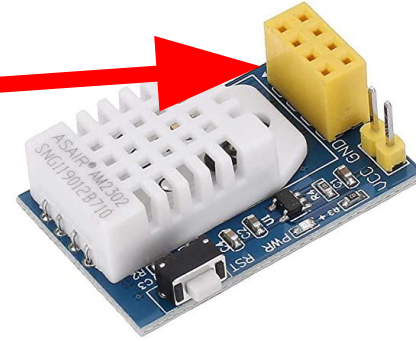
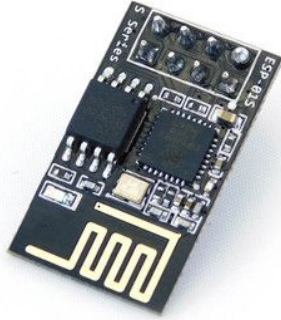
- 8 wyprowadzeń:
  - 2 x GPIO - we/wy cyfrowe (GPIO0 & 2)
  - UART (TX, RX)
- montaż: THT
  - raster: 2.54 mm
- wbudowana antena
- wymiary 24.8 x 16 mm



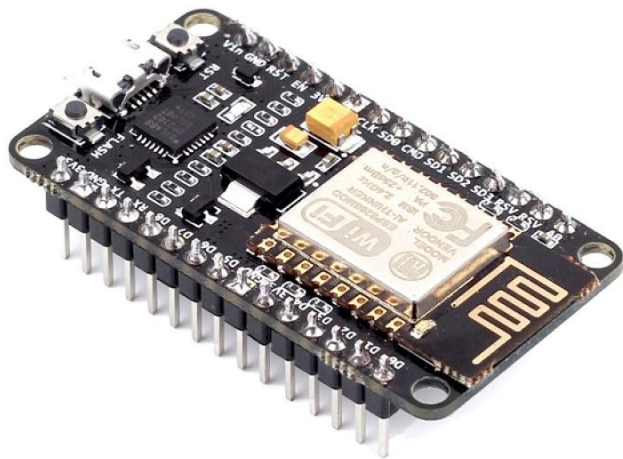
# ESP8266-01

---

ESP8266



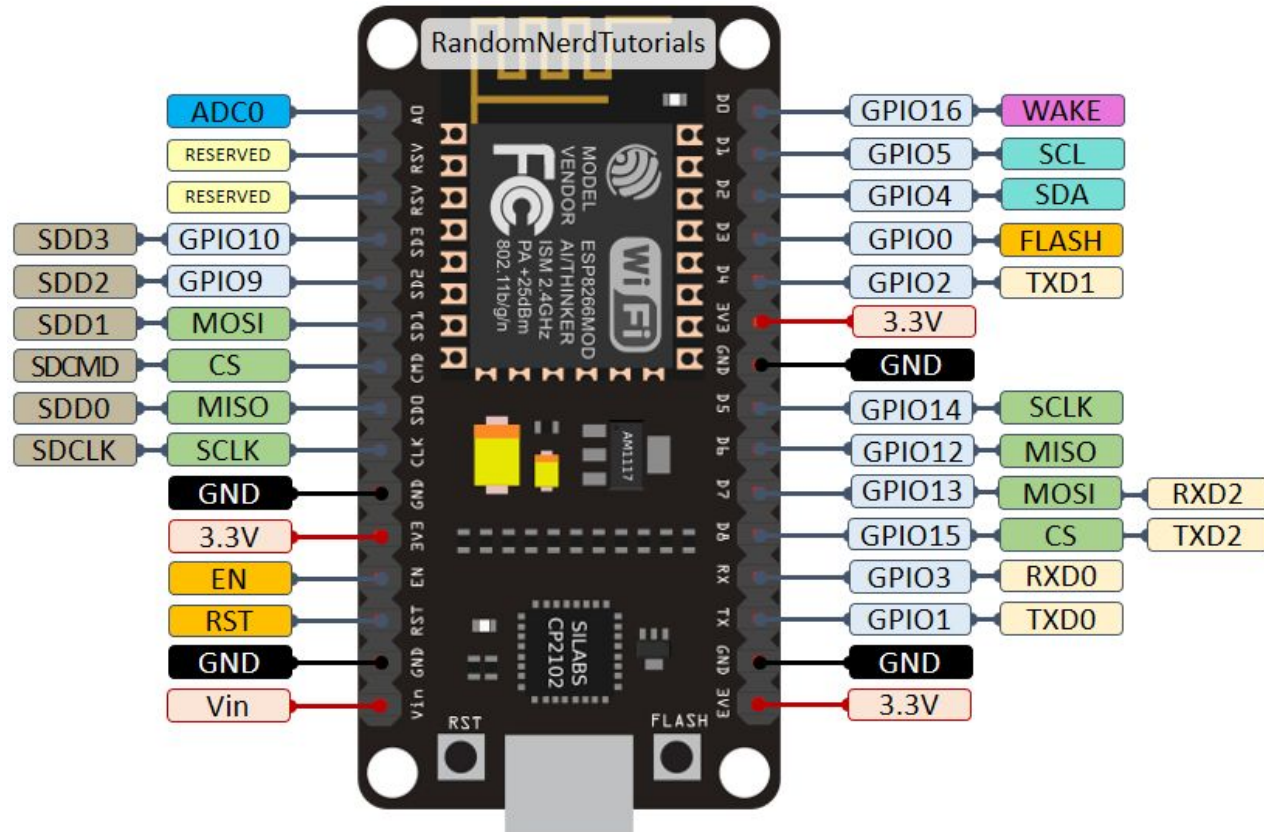




- 10 GPIO - każde może działać jako PWM, I2C lub 1-Wire
- pamięć flash: 4 MB
- interfejs USB-UART (CP2102)
- wbudowana antena
- raster wyprowadzeń: 2.54 mm
- wymiary: 49 x 25 mm

# NodeMCU v2

# ESP8266



# ESP8266

# ESP8266

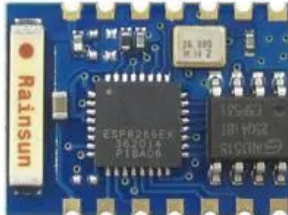
ESP-01



ESP-02



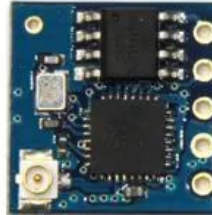
ESP-03



ESP-04



ESP-05



ESP-06



ESP-07



ESP-08



ESP-09



ESP-10



ESP-11



ESP-12



ESP-13



ESP-14



# ESP8266 modes

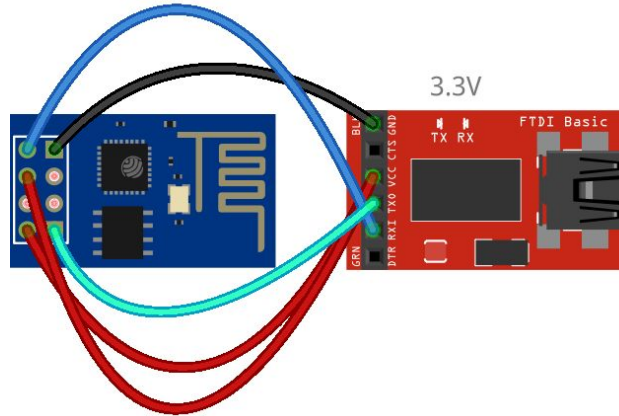
---

- AT commands

[www.espressif.com/sites/default/files/documentation/4a-esp8266\\_at\\_instruction\\_set\\_en.pdf](http://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf)

- UART/SPI  $\leftrightarrow$  802.11b/g/n
- SDK

# ESP8266 AT



Command	AT+CWMODE=<mode>
Response	OK
Parameters	<mode> 1 : station mode 2 : softAP mode 3 : softAP + station mode
Notes	This setting will be stored in the flash system parameter area. It won't be erased even when the power is off and restarted.

# ESP8266 SDKs

---



- [Arduino](#) — A C++-based firmware. With this core, the ESP8266 CPU and its Wi-Fi components can be programmed like any other Arduino device. [The ESP8266 Arduino Core is available through GitHub.](#)
- [ESP8266 BASIC](#) — An open-source [BASIC](#)-like interpreter specifically tailored for the [Internet of Things](#) (IoT). Self-hosting browser-based development environment.
- [ESP Easy](#) — Developed by [home automation](#) enthusiasts.

# ESP8266 SDKs

---



- [ESPHome](#) — ESPHome is a system to control your ESP8266/ESP32 by simple yet powerful configuration files and control them remotely through home automation systems.
- [Tasmota](#) - open-source firmware, very popular with home automation enthusiasts.
- [ESP-Open-RTOS](#) — Open-source FreeRTOS-based ESP8266 software framework.
- [ESP-Open-SDK](#) — Free and open (as much as possible) integrated SDK for ESP8266/ESP8285 chips.



# ESP8266 SDKs

---



- [Espruino](#) — An actively maintained JavaScript SDK and firmware, closely emulating [Node.js](#). Supports a few MCUs, including the ESP8266.
- [ESPurna](#) — Open-source ESP8285/ESP8266 firmware.
- [Forthright](#) — Port of Jones Forth to the ESP8266 microcontroller.
- [MicroPython](#) — A port of [MicroPython](#) (an implementation of Python for embedded devices) to the ESP8266 platform.
- [Moddable SDK](#) — includes JavaScript language and library support for the ESP8266

# ESP8266 SDKs

---



- [Mongoose OS](#) — An open-source operating system for connected products. Supports ESP8266 and ESP32. Develop in C or JavaScript.<sup>[13]</sup>
- [NodeMCU](#) — A [Lua](#)-based firmware.
- [PlatformIO](#) — A cross-platform IDE and unified debugger, which sits on top of Arduino code and libraries.
- [Punyforth](#) — Forth-inspired programming language for the ESP8266.
- [Sming](#) — An actively developed asynchronous C/C++ framework with superb performance and multiple network features.

# ESP8266 SDKs

---



- [uLisp](#) — A version of the [Lisp](#) programming language specifically designed to run on processors with a limited amount of RAM.
- [ZBasic for ESP8266](#) — A subset of Microsoft's widely-used Visual Basic 6, which has been adapted as a control language for the ZX microcontroller family and the ESP8266.
- [Zerynth](#) — IoT framework for programming ESP8266<sup>[14]</sup> and other microcontrollers in [Python](#).

# ESP8266 AP/STA

---

