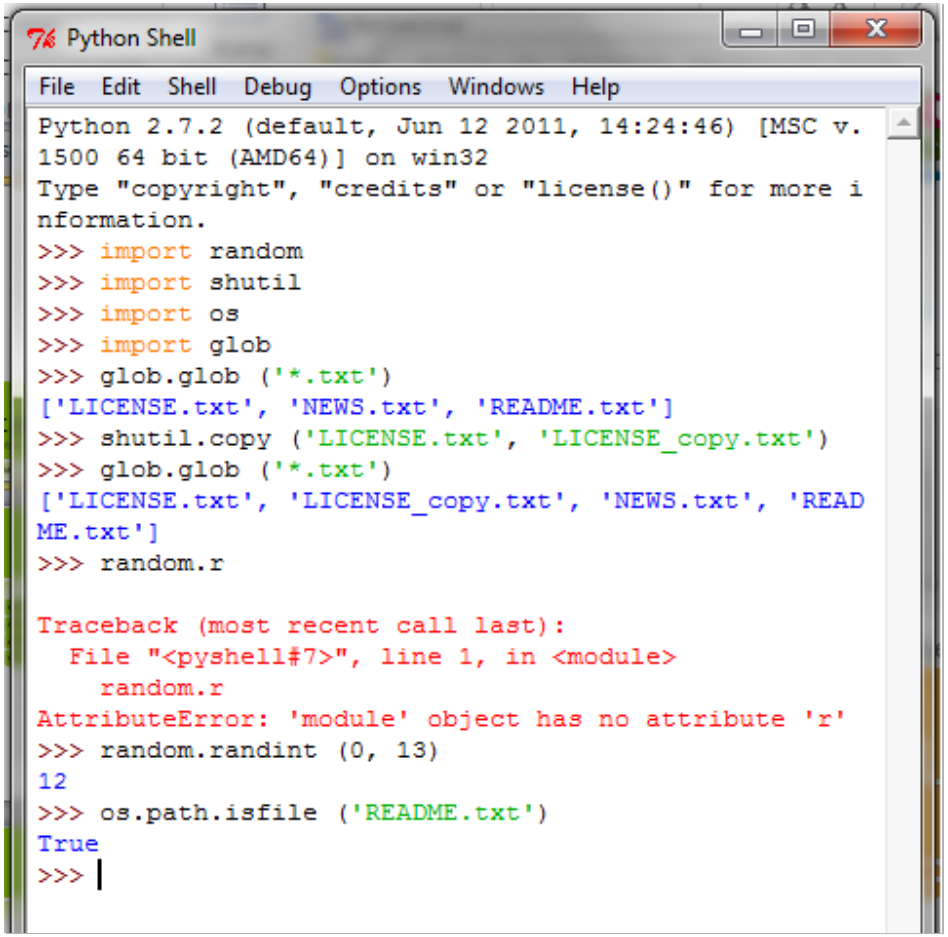


Python para IoT

Micro Python

Javier Becerra
javier@auva.es

ThinkTIC – Julio 2017

Contenidos

1. Micro Python
2. Micro Python en NodeMCU
3. MQTT

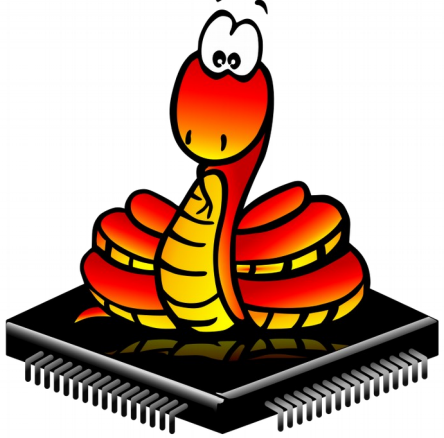
Micro Python

Micro Python

Creada en 2013 por el australiano Damien George, es una implementación de Python 3 específicamente diseñada para microcontroladores.

Incorpora:

- Módulos para acceder al hardware de bajo nivel
- Selección de módulos de la biblioteca estándar
- Implementación en C
- Intérprete de Python interactivo



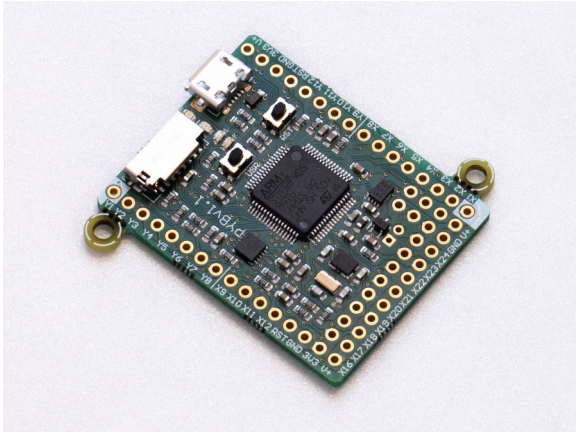
Micro Python

PyBoard

Creada por Damien George junto con Micro Python

Características:

- 1024KiB flash ROM y 192KiB RAM
- 24 GPIO, 3 ADC, 2 DAC
- Acelerómetro 3 ejes
- 168MHz Cortex M4 CPU



Micro Python

NodeMCU (ESP8266)

Basada en chip ESP8266:

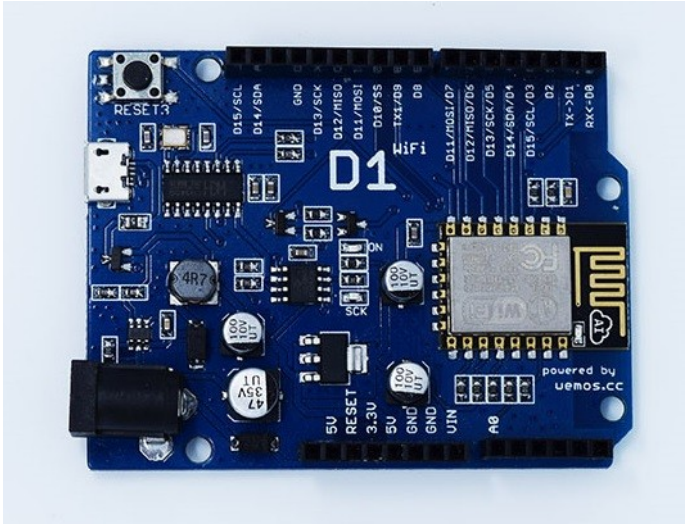
- 4096KiB flash ROM y 128KiB RAM
- Incorpora conversor para conector micro-usb
- WiFi 802.11n
- 16 puertos GPIO
- 1 entrada ADC de 10bits
- CPU 80MHz (160MHz max)
- PWM/I2C/IIC/1-Wire/SPI/SDIO

Micro Python

Wemos D1 (ESP8266) – Wemos D1 R2

Pinout compatible con Arduino

- 4096KiB flash ROM y 64+96KiB RAM
- Alimentación por micro-usb o clavija
- WiFi 802.11n
- 11 puertos GPIO
- 1 entrada ADC
- CPU 80MHz (160MHz max)
- PWM/I2C/SPI



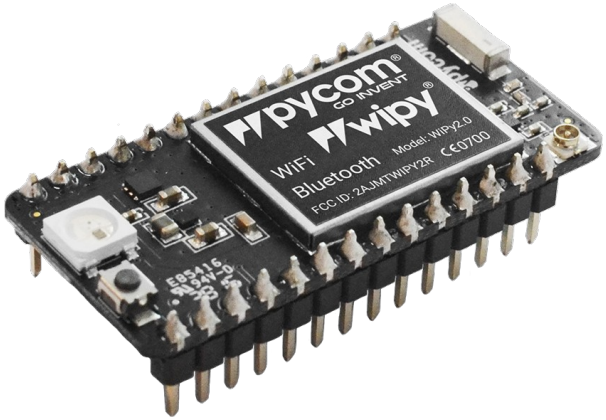
Micro Python

WiPy 2.0 (ESP32)

Basada en chip ESP32, sucesor del ESP8266:

- Conectividad WiFi, Bluetooth
- 4MB Flash, 512KB ram,
- 24 GPIO, 8x12bits ADC
- 2xUART, 2xSPI, I2C
- Versiones con conectividad

  Lora, Sigfox, LTE



Micro Python

Micro Python está basado en Python 3.4.
No es una implementación completa, y en ocasiones se realizan ciertos sacrificios de funcionalidad en aras de una mayor eficiencia.

El proyecto Micro Python Lib contiene módulos que completan la especificación original de Micro Python.

Micro Python en NodeMCU

NodeMCU

Las placas de desarrollo de NodeMCU vienen precargadas con un Firmware que permite programarlas en el lenguaje LUA.

Para usar Micro Python en ellas tenemos que grabar una nueva imagen de ROM.

NodeMCU

La documentación de Micro Python para ESP8266 contiene instrucciones para actualizar el firmware de la placa NodeMCU:

https://docs.micropython.org/en/latest/esp8266/esp8266/tutorial/intro.html

NodeMCU

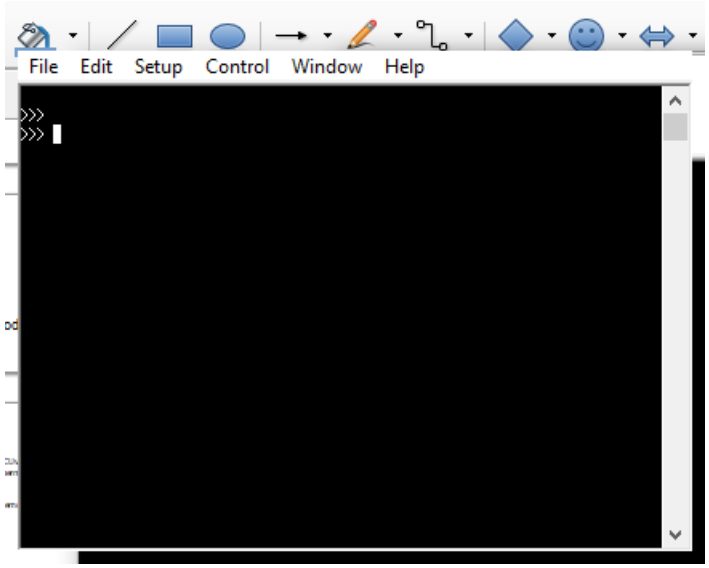
Una vez hayamos actualizado el firmware, tendremos que conectarnos para verificar que efectivamente Micro Python se ha instalado correctamente.

Para ello necesitaremos un terminal (por ejemplo Tera Term en Windows o), que nos permita seleccionar una conexión por puerto serie vía USB.
Una vez conectado, tenemos que asegurarnos de que la velocidad de conexión del puerto serie es 115200 baudios (menú Setup -> serial port)

NodeMCU

Si la instalación ha tenido éxito, podremos acceder a la interfaz REPL de Micro Python:

Read-Evaluate-Print-Loop

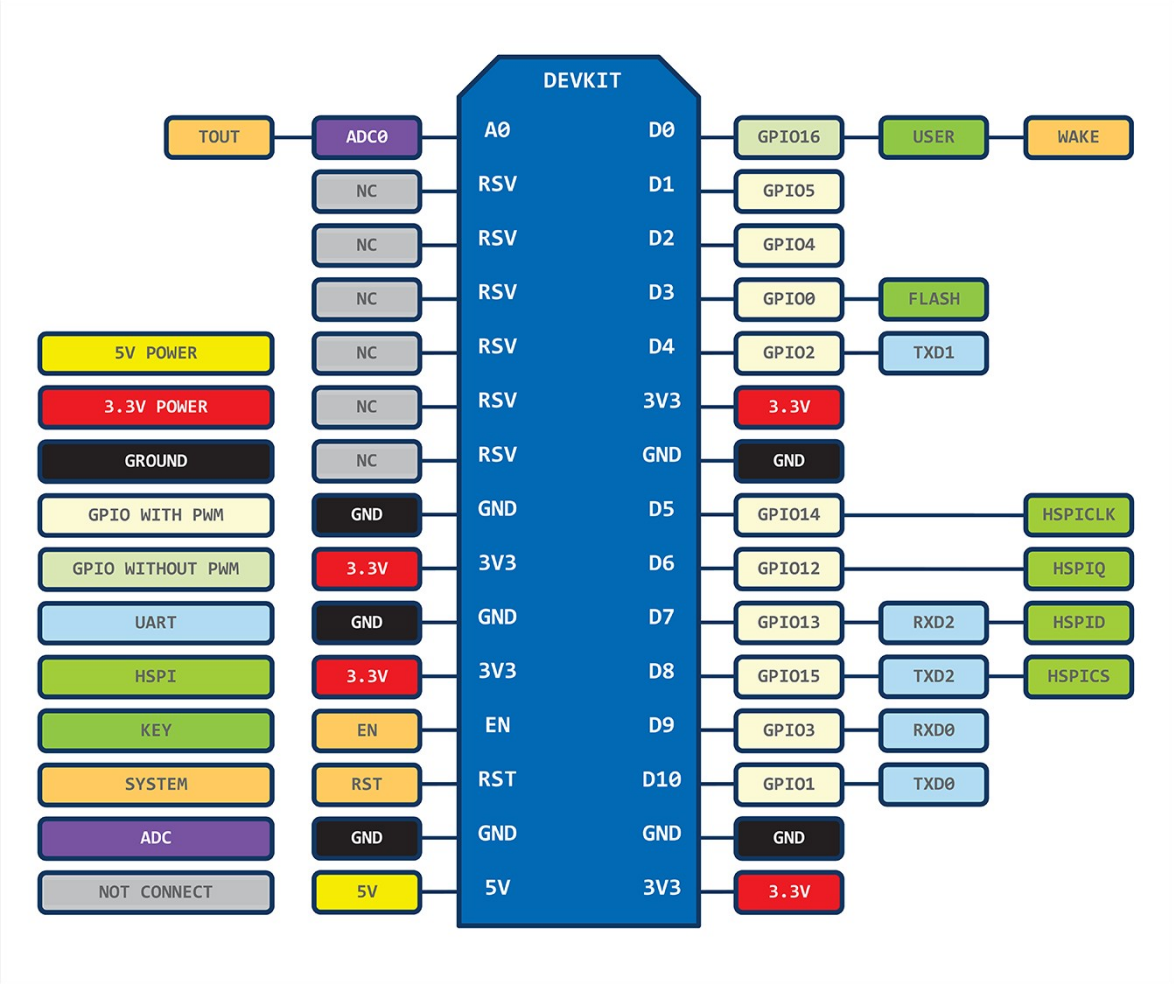


NodeMCU

```
# Comprobamos el funcionamiento correcto de la placa

from machine import Pin
from time import sleep

# GPIO16 (D0) está conectado internamente a un led
led = Pin(16, Pin.OUT)

while True:
    led.off()
    sleep(1)
    led.on()
    sleep(1)
```

NodeMCU

PinOut



NodeMCU

Nada más instalarlo, MicroPython activa su conexión wifi como AccessPoint.
El SSID de la red Wifi será MicroPython-xxxxxx, siendo xxxxxx los últimos digitos de la dirección MAC de nuestra placa.
¿Y cómo sabemos la MAC?

NodeMCU

```
import network
import ubinascii
mac = ubinascii.hexlify(network.WLAN().config('mac'),':').decode()
print(‘MAC: {}’.format(mac))
print(‘WiFi: MicroPython-{}’.format(‘’.join(mac.split(‘:’)[-3:]))
print(‘Clave: micropythoN’)
```

NodeMCU

Para activar la interfaz web de repl basta con importar el módulo webrepl_setup
Y reiniciar tras configurar el acceso vía web.

NodeMCU

Cómo os habéis portado bien, vamos a ver una película:

```
import socket
addr_info = socket.getaddrinfo("towel.blinkenlights.nl", 23)
addr = addr_info[0][-1]
s = socket.socket()
s.connect(addr)

while True:
    data = s.recv(500)
    print(str(data, 'utf8'), end='')
```

NodeMCU

En https://github.com/lvidarte/esp8266 encontraréis un tutorial completo con varios ejemplos más de como interactuar con nuestro esp8266 usando MicroPython.

MQTT

MQTT

MicroPython incorpora el módulo umqtt para enviar mensajes. Podemos comprobar la funcionalidad en la documentación de MicroPython.
En tenemos un ejemplo para leer una entrada y enviar su valor de forma periódica a un servidor remoto vía MQTT
https://home-assistant.io/blog/2016/08/31/esp8266-and-micropython-part2/