

Introducción a *MicroPython* y Raspberry Pi

Primera Sesión. (30 min)

Orden del día:

- Contexto IOT.
- ¿Qué es el Firmware?
- Hardware: Esp8266 y derivados.
- ¿Qué es MicroPython?
- ¿Micropython vs Arduino?
- Fundamentos MicroPython:
 1. Gpio
 2. Wifi
 3. Interrupciones
 4. Librerías overview



Internet of Things?

“IoT solutions can **record observations** in the form of data from one or more **sensors** and **make them available** for viewing by anyone anywhere **via internet.** ”

-Charles Bell (MicroPython for the internet of things)



Internet of Things?

“IoT and AI help strengthen the connection between the intelligent cloud and the physical world. Using data to create a “digital twin” (a virtual model of people, spaces and devices), **we can analyze how space is used and optimize it to better serve people’s needs** at every level from energy efficiency to employee satisfaction and productivity. ”

-Microsoft Blog



¿Qué es el Firmware?

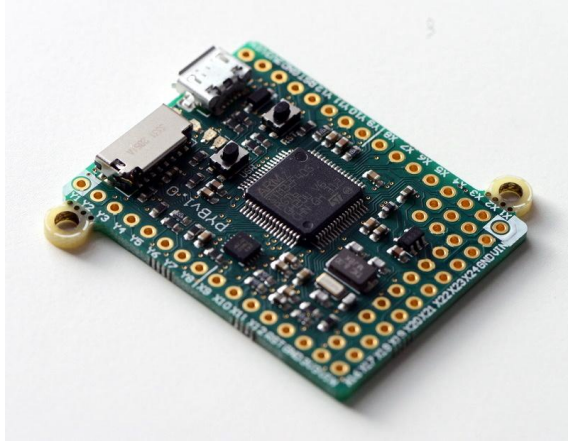
“El firmware es un programa que contiene una serie de instrucciones que permite al hardware (dispositivo físico) interactuar con el software (programa).

El firmware sirve de intérprete entre las órdenes recibidas y el funcionamiento del dispositivo.”

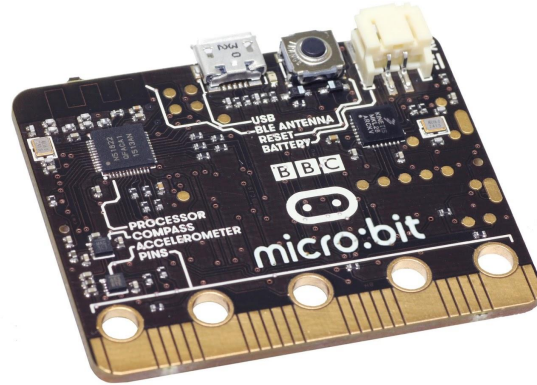
-SoftDoit



Hardware: Pyboard, Microbit, Lopy y muchas Más



Accelerometer, RTC, 4 LEDs, 2 switches, 30 GPIO



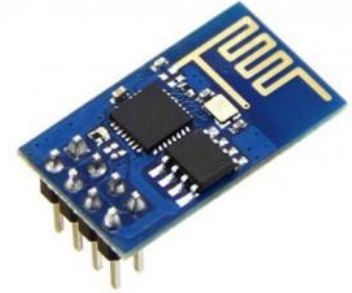
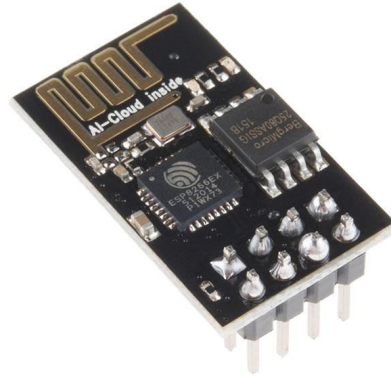
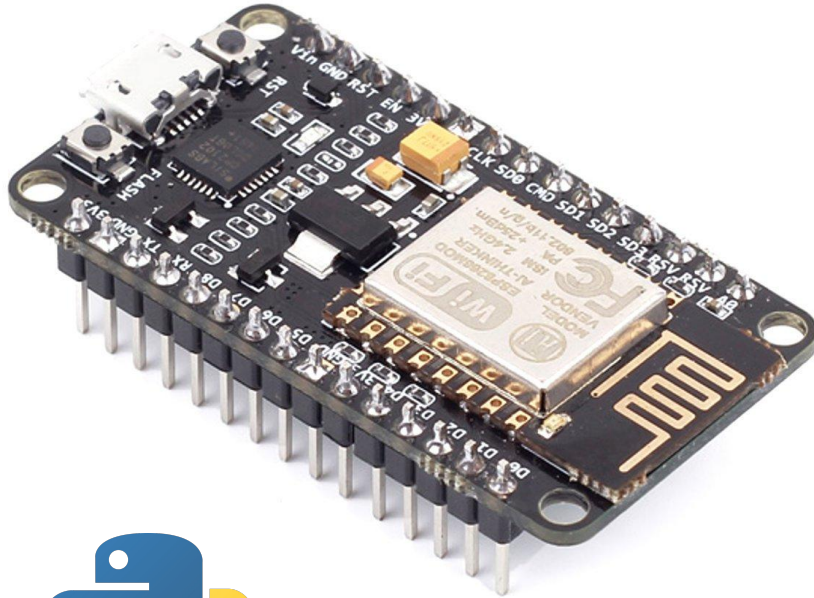
16kb RAM, 256kb flash, Cortex M0 @ 16 MHz



LoRa + Python

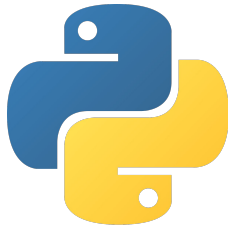
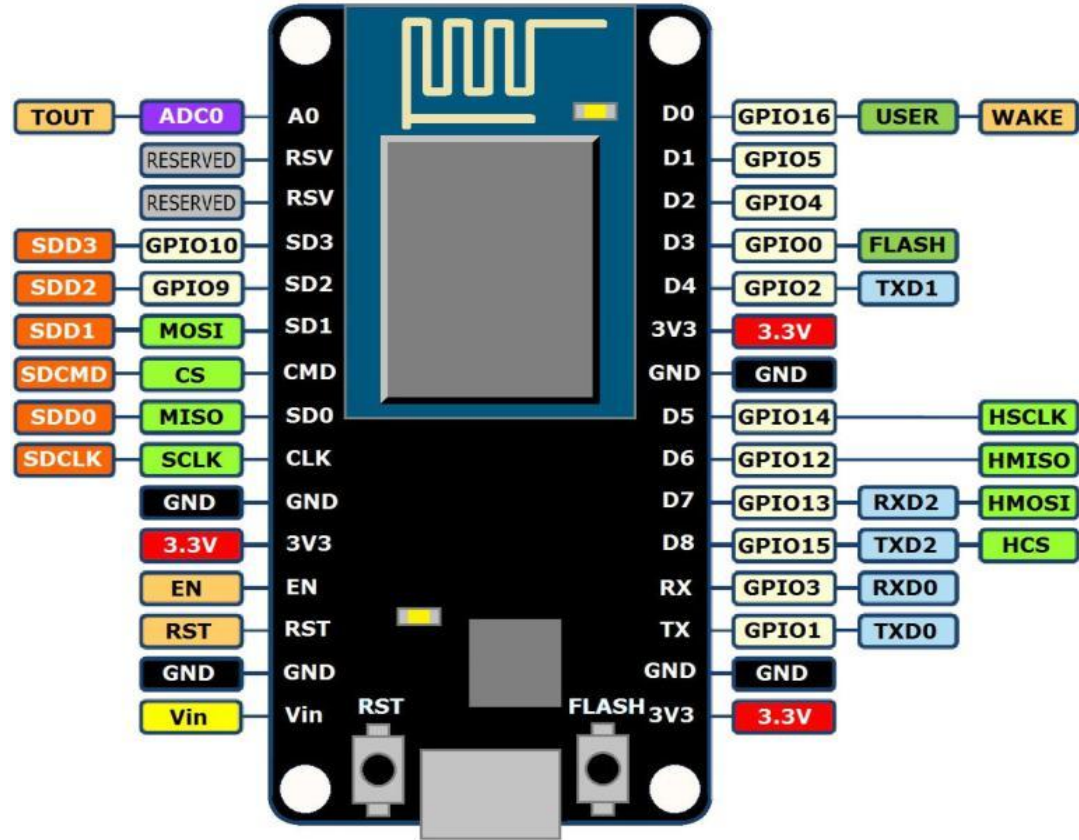


Hardware: Esp8266 (NodeMcu)



Hardware: Esp8266 (NodeMcu)

Mapa de Pines:



¿Qué es MicroPython?

- Nace en Kickstarter
- Damien George
- Simple Python



MicroPython

¿Qué es MicroPython?

“MicroPython is a tiny open source Python programming language interpreter that runs on small embedded development boards”

-Adafruit.



MicroPython

¿Qué es MicroPython?

“MicroPython is a lean and fast implementation of the Python 3 programming language that is optimised to run on a microcontroller”

-[Micropython.org](https://micropython.org)



MicroPython

¿MicroPython vs Arduino?



MicroPython

“(...) Arduino is an entire 'ecosystem' with the Arduino IDE (i.e. the desktop application you use to write and upload sketches), the Arduino programming language (based on C/C++), and Arduino hardware like the Arduino Uno R3 board.”

-Adafruit

A decorative graphic at the bottom right of the slide, consisting of several overlapping triangles in shades of pink and red, creating a modern, abstract design.

¿MicroPython vs Arduino?

“MicroPython language is interpreted instead of being compiled into code the CPU can run directly like with the Arduino programming language.”

-Adafruit



MicroPython

¿MicroPython vs Arduino?

“(…), there's less performance and sometimes more memory usage when interpreting code.”

-Adafruit



MicroPython



Fig 1. PyBoard, corriendo hola mundo (blink Led)



MicroPython

Fundamentos de uPython: Librerías (Módulos)

<https://github.com/micropython/micropython-lib>

“Currently you can access GPIO pins, connect to a WiFi network, and talk to the internet using a low-level socket-like interface with MicroPython on the ESP8266. Access to SPI, or other parts of hardware are not yet supported”



MicroPython

-Adafruit

Fundamentos de uPython: Librerías (Módulos)

BUT NOT ALL OF IT

```
>>> import functools
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: no module named 'functools'
>>> import this
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: no module named 'this'
```



MicroPython

Fundamentos de uPython: Requerimientos Físicos

- Pc (obviamente)
- Board de Desarrollo (esp32, esp826, etc)
- Cable Usb tipo C (Celulares)
- Cableado utp, leds, sensores, relays, etc
- Herramientas como alicates, pinzas, etc.
- Conocimiento en Electrónica (poco)



Fundamentos de uPython: Requerimientos SW

- Win. Driver: USB to UART (Cp210x)
- Firmware uPython: Para nodeMcu u otros.
- Flasher de Firmware (nodeMcu flasher, otros)
- IDE: uPycraft (win, linux), Jupyter
- Módulos adicionales.



MicroPython

Fundamentos de uPython: El lenguaje

```
sonic.py
import music
from microbit import *

def sonic(f=3000, fstep=32, fdepth=100, ton=10, toff=0, duration=2.0):
    try:
        ft = f
        music.pitch(ft)
        st = running_time()
        while running_time() < st+(duration*1000):
            sleep(ton)
            if toff != 0:
                music.stop()
                sleep(toff)
            fn = ft + fstep
            if fn >= f + fdepth or fn <= f - fdepth:
                fstep *= -1
            if fn != ft:
                ft = fn
                music.pitch(ft)
    finally:
        music.stop()

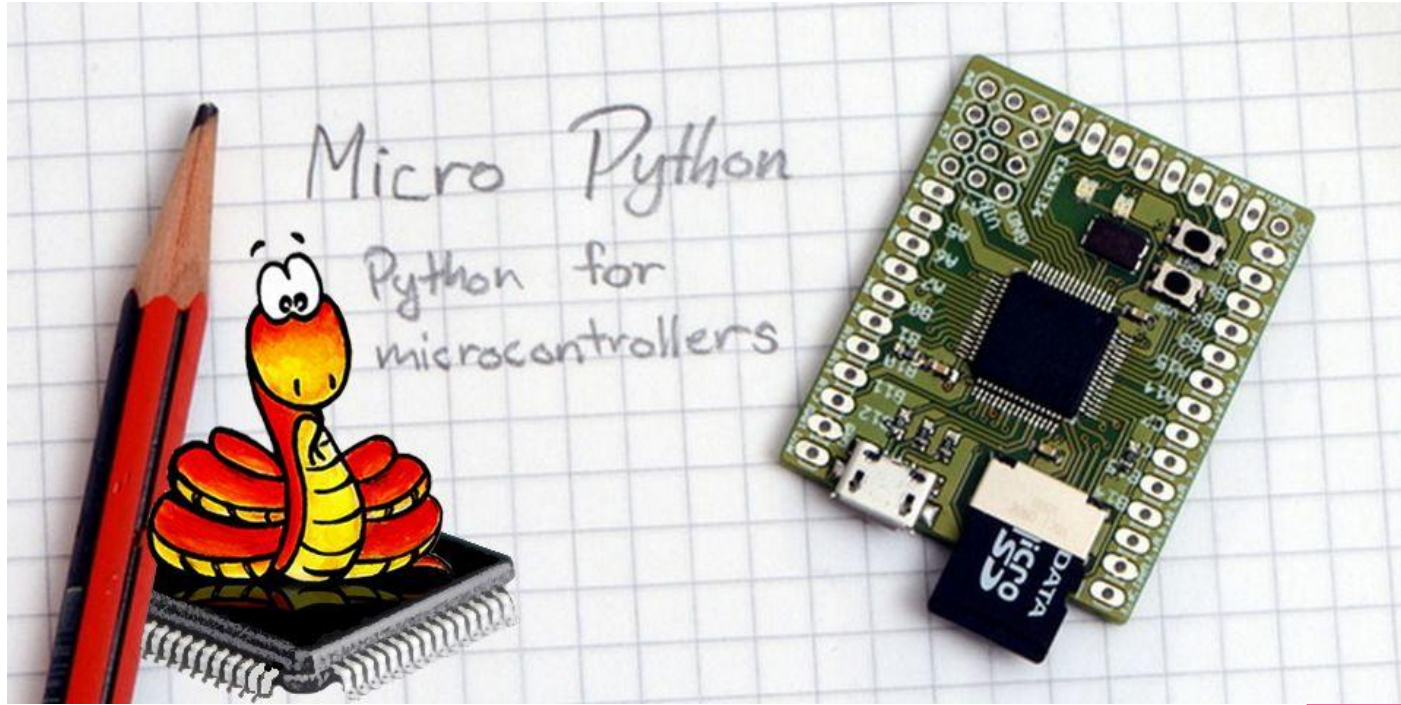
while True:
    if button_a.was_pressed():
        display.show(Image.SWORD)
        sonic(f=3000)
    elif button_b.was_pressed():
        display.show(Image.SWORD)
        sonic(f=4000)
    else:
        display.show(Image.DIAMOND)
```

- Funciones
- Clases
- Tipos de Variables
- Excepciones
- Y muchos más



MicroPython

¡Y a programar!



MicroPython

