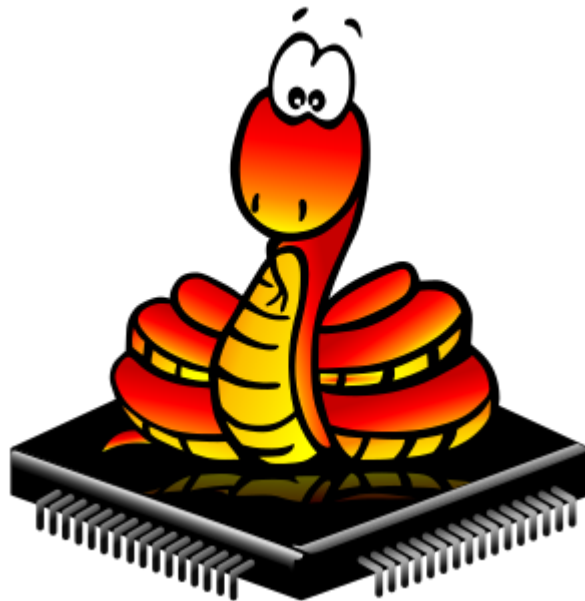


NodeMCU 13: intentando el uso de micropython en un v1.0 Lolin

agrportfolioeducativo.blogspot.com/2020/05/nodemcu-13-intentando-el-uso-de.html



Ya que no puedo terminar las placas, y ahora mismo con el estado de alarma no sé muy cómo continuar sin brocas adecuadas (no me atrevo a pedir las a China), continuaré practicando algo más con los NodeMCU.

En este caso creo que intentaré usar micropython como lenguaje.

En primer lugar, estoy usando los apuntes de la página:

<https://randomnerdtutorials.com/getting-started-micropython-esp32-esp8266/>

=====

Lo primero es cargarle un firmware apropiado al NodeMCU. Tras unos intentos infructuosos, lo he conseguido. La lista de firmwares viene en la página

<https://micropython.org/download/esp8266/>

1. Me he descargado el primero estable,
<https://micropython.org/resources/firmware/esp8266-20191220-v1.12.bin>
2. He instalado con **pip install esptool** la herramienta esptool.py (version 2.8), tal como se describe en la web: <https://github.com/espressif/esptool> **Nota importante:** no me ha funcionado la herramienta esptool directamente instalada desde comandos con apt (parece antigua).
3. Comando que borra la memoria del NodeMCU: **esptool.py --port /dev/ttyUSB0 erase_flash**
4. Comando que flashea el NodeMCU: **esptool.py --chip esp8266 --port /dev/ttyUSB0 write_flash --flash_mode dio --flash_size detect 0x0 ~/Escritorio/NodeMCU_micropython/esp8266-20191220-v1.12.bin**

= = = = =

Instalo el programa thonny, que es un IDE para Python, desde el terminal con la orden **pip3 install thonny**. Hay que ejecutarlo desde un terminal. Mediante ese procedimiento se descarga la última versión del programa.

En opciones, intérprete, elegir MycroPython (generic) y el puerto al que está conectado.

En shell, teclear help() para ver la salida y comprobar que todo funciona.

Y, por fin, escribir el primer programa en micropython.

```
from machine import Pin
```

```
from time import sleep
```

```
led = Pin(2,Pin.OUT)
```

```
while True:
```

```
    led.value(not led.value())
```

```
    sleep(0.5)
```

[view raw NodeMCU_MP_01_Blink.py](#) hosted with ♥ by [GitHub](#)

Primeros pasos

Bueno, parece ser que los programas no se cargan directamente en memoria. Hay un programa llamado boot.py que arranca con el dispositivo.

Hay que hacer dos cosas:

1. En el programa **boot.py** incorporar la línea **import [nombre programa].py** que queramos ejecutar.
2. Guardar el programa **[nombre programa].py** al dispositivo. Click derecho en el nombre del programa y elegir "**subir a**".