

NodeMCU: el comienzo (I).

 agrportfolioeducativo.blogspot.com/2019/06/nodemcu-el-comienzo-i.html

Hace tiempo que tenía en mente introducirme en el mundo de los dispositivos IoT. De hecho, había comprado un NodeMCU, sin saber bien para qué servía.

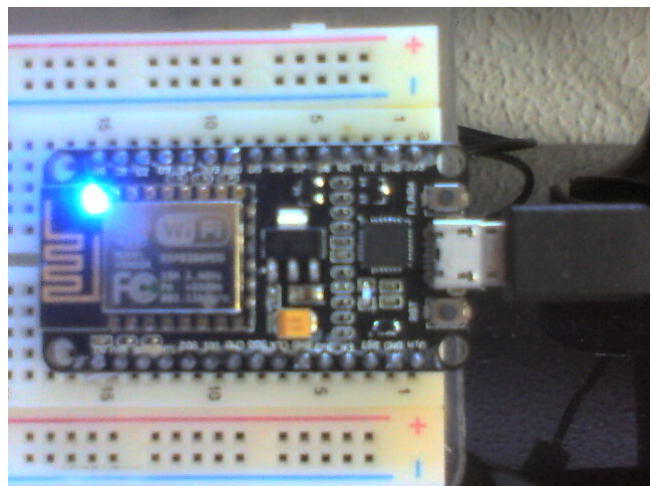
Indagando un poco, me doy cuenta de que es todo un mundo nuevo, para mí, sin explorar.

El primer paso ha consistido en conectarlo al ordenador, para poder cargarle programas. ¿Qué he hecho?

1.- Pincharlo en una placa ProtoBoard.

Mi dispositivo es un módulo NodeMCU v1.0/V2, según lo que aparece en el tutorial de esta página (<https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>) .

Pinchalo en la placa tal y como observas. Deja una línea de pines vacíos a izquierda y derecha para poder conectarle componentes.



Mi propio NodeMCU

2.- Conectarlo mediante un microUSB a mi PC con Ubuntu 18.04.2 LTS

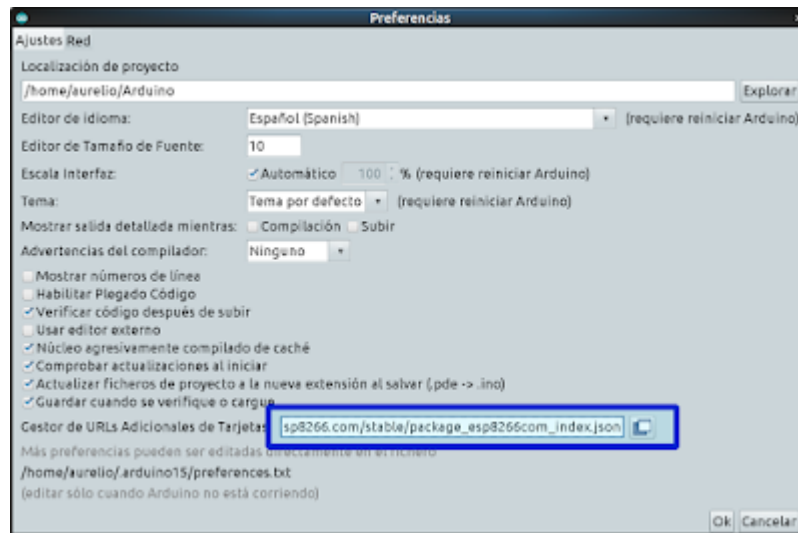
Necesitas el IDE Arduino. Para poder instalarlo descárgalo de esta página:

<https://www.arduino.cc/en/Main/Software>.

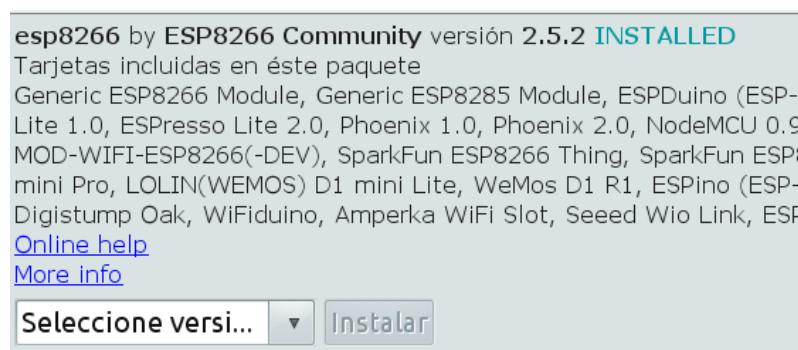
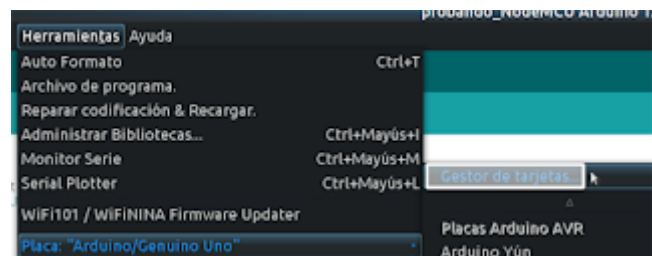
Puede que necesites información adicional, como agregar al usuario de Linux al grupo dialout: <https://www.todavianose.com/problemas-con-el-puerto-de-arduino-en-ubuntu-linux/>

Una vez tengas activo el IDE Arduino, necesitas una librería para ESP8266 (componente principal de NodeMCU).

- Ve a preferencias, e indica la dirección al gestor de tarjetas ESP8266:
http://arduino.esp8266.com/stable/package_esp8266com_index.json



- Dirígete al gestor de tarjetas (herramientas >> placas) y filtra por ESP8266 para instalar la última versión ESP8266. En mi caso ya está instalada. Si no la tienes, o la tienes desactualizada, instálala/actualízala.



- Una vez instalado todo, elige la placa "NodeMCU 1.0 (ESP-12E) Module". Yo le dejé el resto de parámetros por defecto, y tan sólo tengo que seleccionar el puerto. En mi caso el **/dev/ttyUSB0**.

3.-Probando un código...

Fácilmente podemos probar nuestro NodeMCU. Cargamos por ejemplo el siguiente código:

```
#define LED_BUILTIN 2

void setup(){
  // El LED integrado está conectado al pin 2.
  pinMode(LED_BUILTIN, OUTPUT);
}
```

```
void loop() {  
  // Encender el LED (Está conectado con una resistencia pull-up,  
  // por eso se enciende si el pin está a nivel bajo)  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000); // Esperar un segundo  
  // Apagar el LED  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000); // Esperar un segundo  
}
```

Y hará parpadear el pin del ESP8266 con una cadencia de un segundo.

Nota: código resaltado con <https://quickhighlighter.com/>