



Taller Introducción a loT: Planta Twittera

Jaime Laborda Macario

Miembro de MakersUPV

es.libre.university



¿Quién somos?

- Comunidad de estudiantes
- DIY, Learning by Doing
- + 2000 Makers
- +200 Asociados
- Proyectos
- Talleres prácticos
- Organización de eventos: ORC, FCVLC...
- Salir de las aulas







¿Tol 29 au Signature de la contraction de la con







¿Qué e

Internet de las c un concepto q cotidianos con conexión de Int



T)12 es

DS

as es la



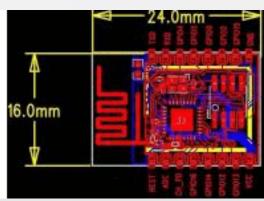




Conociendo al ESP8266

- 32-bit RISC CPU a 80 MHz
- 64 KiB de RAM para instrucciones KiB de RAM para
- IEEE 802.11 k
- 16 pines GP.
- SPI e I2C
- UART
- Convertidor Analógico-Digital (ADde 10 bit









Conociendo al NodeMCU Development Kit



- Proyecto Open-Source para el desarrollo de un modelo sencillo de integrar la loT (Internet of Things)
- Conexión directa a USB
- Regulador alimentación 3.3V
- Gran comunidad
- Programación con Arduino



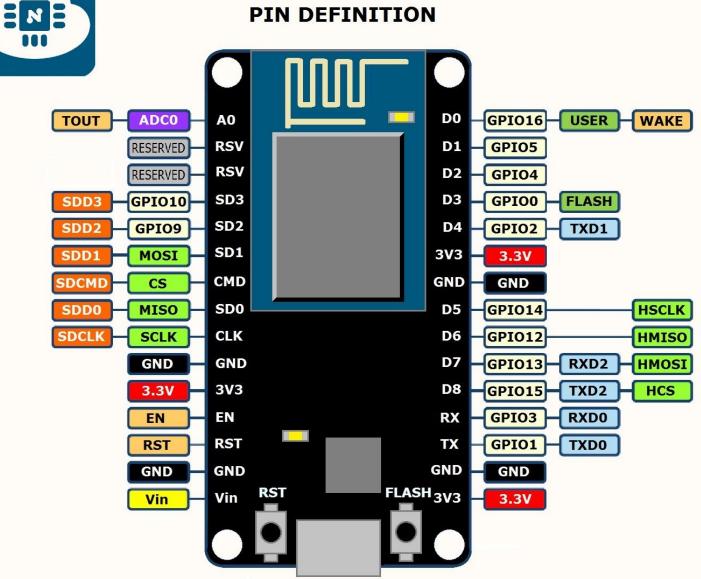




NodeMCU ESP-12 development kit V1.0



PIN DEFINITION





LANSPARÈNCIA)E DADES

iManos a la obra!







2. Programación en Arduino

- Diversas formas de programarlo
 - Sin firmware: Comandos AT
 - Lua
 - Micro-Python
 - Arduino
- Facilidad: Entorno ya familiar
- Librería WiFi compatible







Configuración de Arduino IDE

- Las instrucciones detalladas se encuentran en <u>es.libre.university</u>
- Añadir nuevo repositorio: Archivo → Preferencias
- En gestor de URL añadimos:
 http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Añadir la placa: Herramientas → Placa → Gestor de placas
- Buscamos: "esp8266" e instalamos.





¡Hola mundo!

- Comprobaremos que hemos instalado la placa correctamente
- Comprobaremos que podemos subir sketches desde Arduino
- Abrimos el sketch de ejemplo Blink: Archivos -> Ejemplos -> ESP8266 ->
 Blink
- En Herramientas -> Placa seleccionamos NodeMCU 1.0 (ESP-12E Module)
- Asegurarse de tener el puerto COM correcto



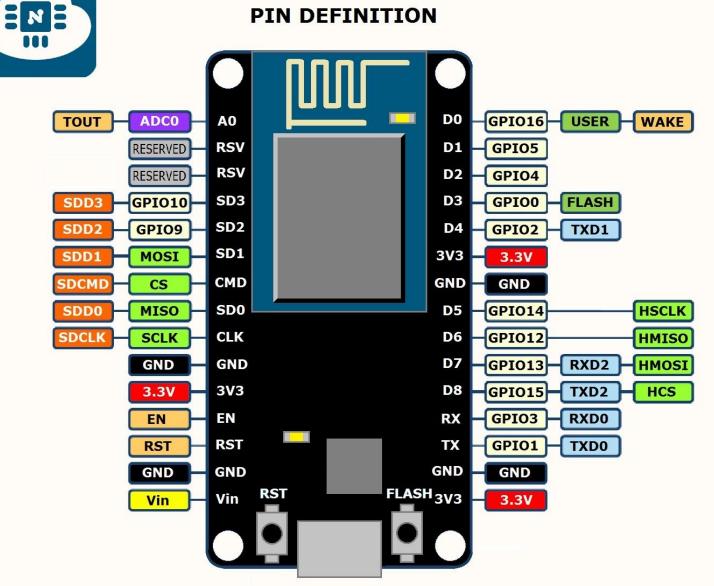


Ejer

- Can
- Can
- Observed

NodeMCU ESP-12 development kit V1.0

PIN DEFINITION



LANSPARÈNCIA)E DADES

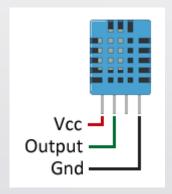
Lectura de sensores

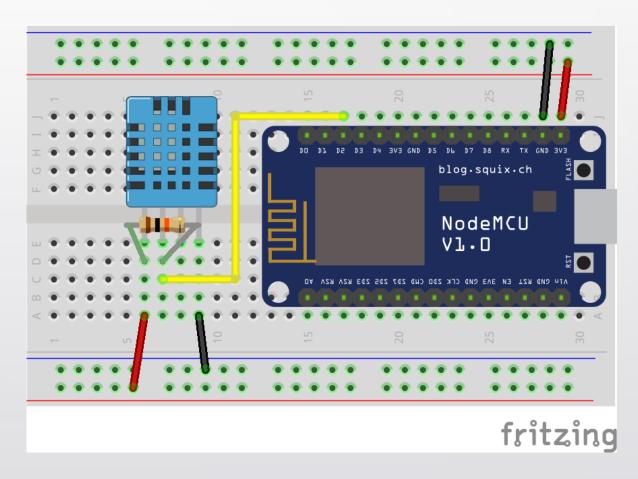




DHT11

- Sensor Temperatura y Humedad relativa
- Usaremos la librería SimpleDHT
- Programa > Incluir librería > Gestor de librerías
 - Buscar SimpleDHT
- Vcc (+) → 3.3V
- GND (-) → GND
- Out → D4
- Cargar ejemplo



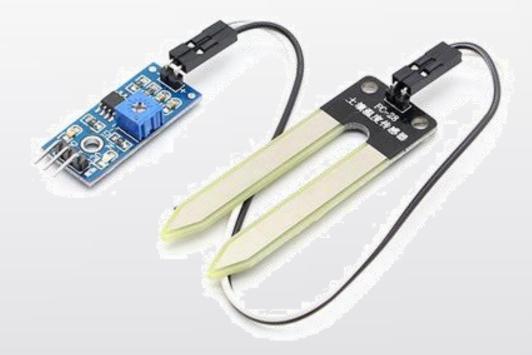






Sensor humedad de suelo

- Sensor Analógico
- Vcc → 3.3V (¡¡OJO, NO CONECTAR A 5V!!)
- GND → GND (0V)
- $A0 \rightarrow A0$
 - Fácil, no?!:D
- Ejercicio: Cargamos el sketch de ejemplo y sumergimos la sonda en un vaso con agua. ¿Qué ocurre?





https://github.com/jaimelaborda/Planta-Twittera



Servidor web: Control de un LED

- Controlar el LED interno de la NodeMCU mediante una página web HTTP
- Mediante dos botones realizaremos peticiones al servidor para que encienda o apague el LED.
- Es el ejemplo más sencillo que se puede realizar después del Blink y con él aprenderemos a conectarnos a una red Wifi, a crear un servidor HTTP, y a responder y diferenciar entre las distintas peticiones que nos hará un cliente.





Conexión a red WiFi

- Usaremos la librería es la <u>ESP8266Wifi.h</u>, muy parecida a la nativa de Arduino WiFi.h.
- WiFi.begin("network-name", "pass-to-network"): Nos permite iniciar una conexión.
 Acepta como parámetros la SSID y el PASS del WiFi.
- WiFi.mode(m): Establece el modo de operación del WiFi (WIFI_AP, WIFI_STA, WIFI_AP_STA o WIFI_OFF)
- **WiFi.status()**: Devolverá el parámetro *WL_CONNECTED* si estamos conectados correctamente.
- WiFi.locallP(): Nos devuelve la IP a la que estamos conectados. (Solo para STA)





```
#include <ESP8266WiFi.h>
void setup()
                                  Disponible en:
                                  https://github.com/jaimelaborda/Planta-Twittera
 Serial.begin(115200);
 Serial.println();
 WiFi.begin("network-name", "pass-to-network");
 Serial.print("Connecting");
 while (WiFi.status() != WL_CONNECTED)
    delay(500);
    Serial.print(".");
  Serial.println();
 Serial.print("Connected, IP address: ");
 Serial.println(WiFi.localIP());
```





Credenciales WiFi

• SSID: TechFest2

• PASS: techfest2017







Control de un LED por WiFi

 Haremos que el NodeMCU se convierta en un pequeño servidor web listo para atender a peticiones HTTP de clientes que se conecten a el.

Código disponible en: https://github.com/jaimelaborda/Planta-Twittera/tree/master/ejemplos/esp8266_led_server







Ejercicio 2

- **Ejercicio 1:** Se propone como ejercicio modificar el código para que funcione de manera correcta, es decir, de manera lógica.
- Ejercicio 2: Conecta un LED externo con una resistencia de 160-330 Ohms a otro pin y modifica el código para encender/apagar desde el navegador dicho LED.
- **Ejercicio 3:** Añade 2 LEDs y modifica el código para que el servidor muestre al usuario 4 botones y poder encender/apagar independientemente uno u otro.





ThingSpeak



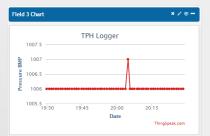




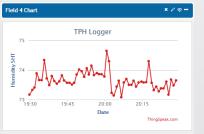
¿Qué es?

- Conjunto de servidor web, más una base de datos y una API que nos permite almacenar y transmitir datos usando el protocolo HTTP
- Sencillez: nos permite desarrollar una aplicación visual para mostrar nuestros datos en pocos minutos y así centrarnos en la parte Hardware.













Creación de una cuenta

- Creamos una cuenta
- Creamos un canal
- Obtenemos la API Key

https://thingspeak.com/





¿Qué es una API?

Conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos.

API REST:

- GET: Pedir al servidor un servicio
- POST: Actualizar una variable creando un nuevo registro
- PUT: Modificar un valor de un registro
- DELETE: Borrar un valor de un registro





ThingTweet

- También de ThingSpeak: Apps → ThingTweet.
- Nos permite comunicarnos con la API de Tweeter y poner tweets en pocos minutos
- Asociamos nuestra cuenta de Twitter
- Obtenemos la API Key
- Ejercicio: Mandar un Tweet desde el ESP8266 mencionando a @techfestUPV





Planta Twittera





