

# ESP8266 IoT nodo

Hardware para el curso SpainLabsIoT2018. V1rA/v2Ra.

By Grafisoft – [www.spainlabs.com](http://www.spainlabs.com)

Version documentación: rev2.

## Introducción

En esta documentación se va a abordar lo relacionado al hardware que se ha diseñado para aplicar los diferentes puntos vistos en el curso SpainLabsIoT2018.

## Hardware

Para disponer de una plataforma hardware sobre la que aplicar lo que se ha ido publicando, se ha diseñado el siguiente nodo. Se ha basado el diseño en NodeMCU Lolin, quedándonos con lo que necesitábamos y añadiendo alguna cosilla más. El objetivo era disponer de un hardware económico sin perder de vista el uso que iba a tener.

El hardware es Open Source. Se dispone de los Gerbers para poder fabricar más PCBs.

El proyecto está desarrollado en la comunidad SpainLabs, concretamente en:

<https://www.spainlabs.com/foros/tema-SpainLabsIoT2018-Caso-real-Nodos-ESP8266>

**GitHub:** <https://github.com/grafisoft/SpainLabsIoT2018>

### Características

Alimentación batería: LiPo 3,7V (1S).

Alimentación Max en el conector de batería: 5V.

Alimentación desde el conector Vext: 7V. Vext mínima recomendada 5v.

- Panel solar recomendado: 6-7v 2-3W.
- Carga de batería con dos opciones, a 100mA y 450mA (seleccionable mediante una resistencia).

Alimentación del módulo ESP8266 a 3v3 y 200mA de I<sub>max</sub>.

Consumo: 150mA en transmisión Wifi.

- DeepSleep: sobre 50uA.

Botón de reset.

Jumper para habilitar/deshabilitar Deepsleep. En alguna de las versiones del ESP, este jumper en el momento de programar no puede estar colocado.

Compatible con ESP8266 especificaciones 07/12/12E

- Posibilidad de tener todos los pines accesibles.
- Zócalos para programar y posibilidad de poder hacer shields para integrar nuevos sensores.
- El módulo ESP8266 puede ser soldado directamente sobre la PCB o pueden añadirse zócalos de 2.00mm para poder extraerlo.

Conectores para batería y panel solar con conector JST. Posibilidad de añadir holder para las baterías tipo 18650. 1 celda o batería o 2 celdas o baterías (en paralelo).

Hardware escalable. No es necesario soldar todos los componentes para su correcto funcionamiento.

Opcionales:

- Etapa de carga.
- Etapa para la programación (se dispone de una etapa de transistores para poder programar el modulo ESP. No es necesaria para tener comunicación UART y/o el funcionamiento normal).

Puerto I2C duplicado. Se han sacado los pines necesarios de I2C para poder conectar directamente un sensor (3v3, GND, SDA, SCL).

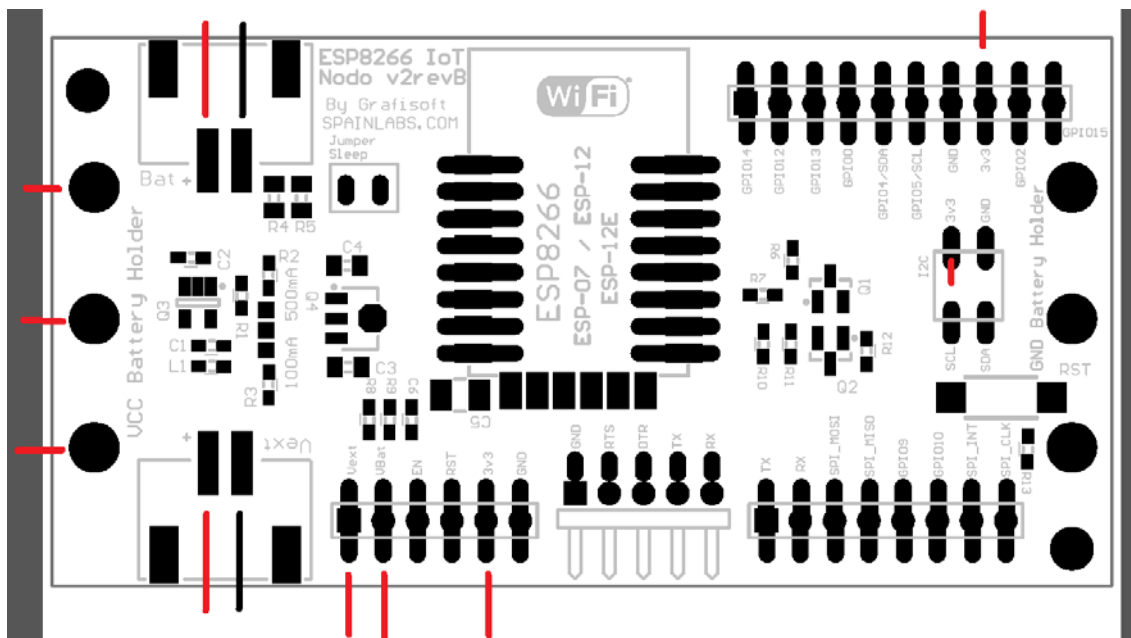
Control del nivel de batería. El puerto ADC del módulo ESP se ha usado para medir la tensión de la batería. Existen ICs ADC por I2C en caso de necesitarlos.

Dimensiones: 79mm x 41mm.

**Nota1:** Es necesario alimentar la PCB para realizar la programación del módulo ESP. No se toma la alimentación del módulo USB-Serie por no ser suficiente.

## Esquema de conexiones

A continuación vamos a ver el pinout de la PCB. Hemos respetado el nombre de cada pin respecto al nombre que tiene en el módulo ESP8266, por tanto tiene una equivalencia directa.



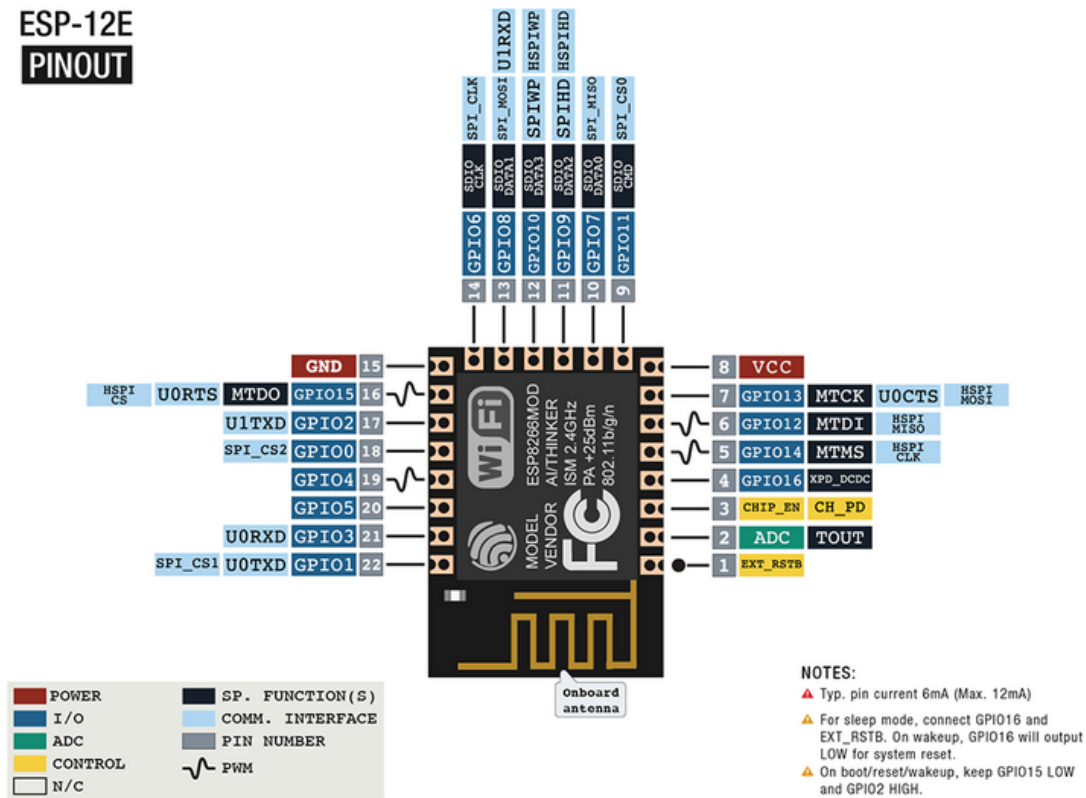
Se han marcado en rojo todas las alimentaciones positivas. En negro las negativas. Todos los pines GND son las conexiones negativas.

Los pines con serigrafía 3v3 corresponden a que en ese pin disponemos de alimentación 3v3. No es para meter de forma externa una alimentación 3v3.

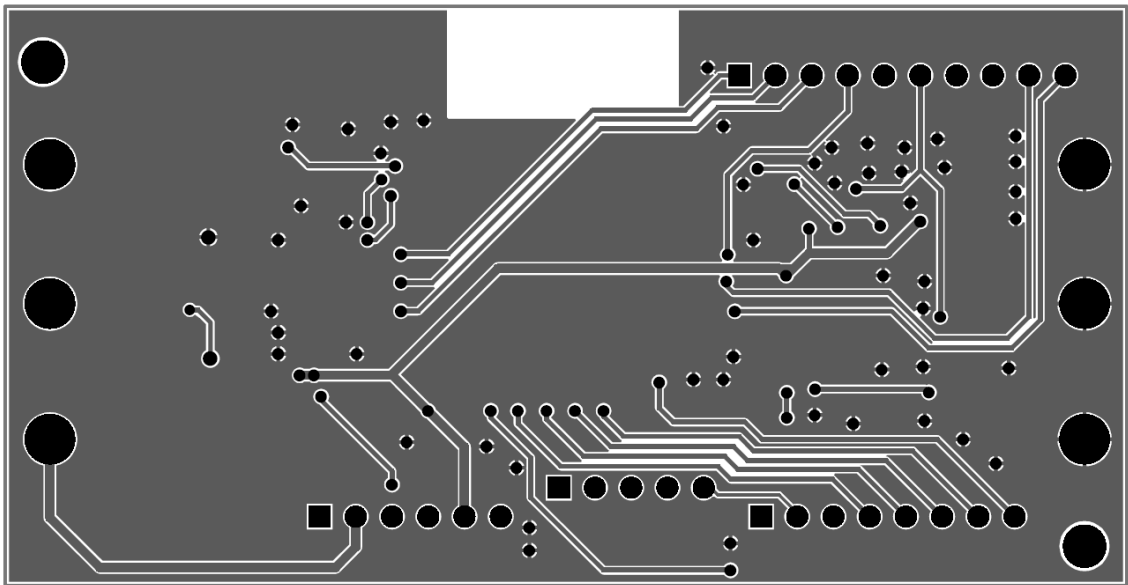
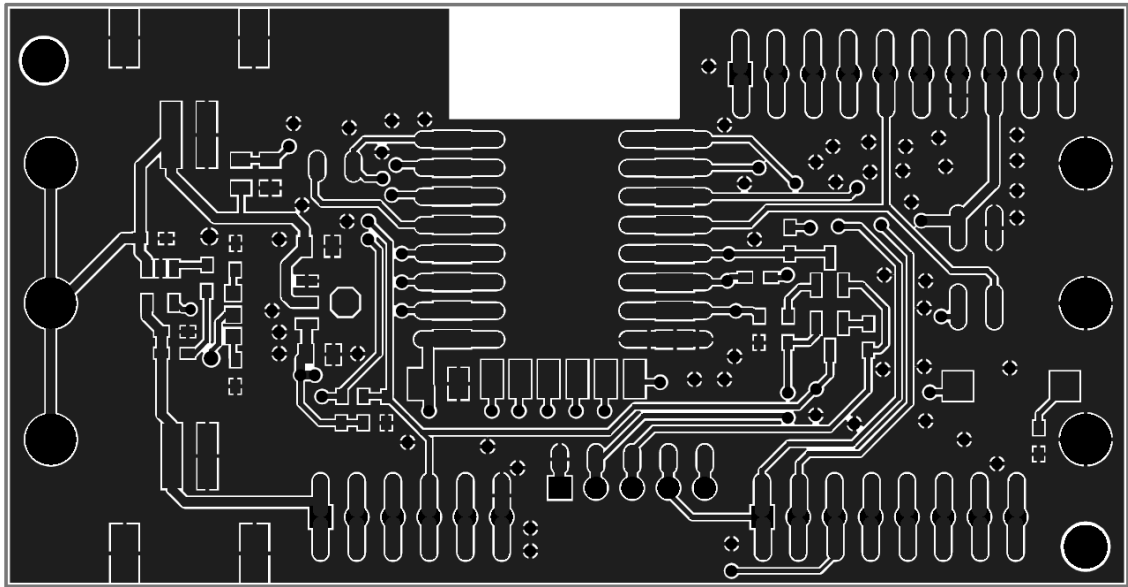
Los pines Vext y Vbat son los mismos que se dispone en los conectores JST. Se han duplicado por si se quieren controlar desde una shield que se pinche encima. Estos pines sí que son para alimentar la PCB.

Si nos vamos al módulo ESP8266 más completo, este sería la especificación 12E. Esta versión tiene todos sus pines accesibles. Las versiones 07 y 12 tienen los mismos pines a excepción de los pines de la parte superior.

## ESP-12E PINOUT



Vistas Top y Bottom.



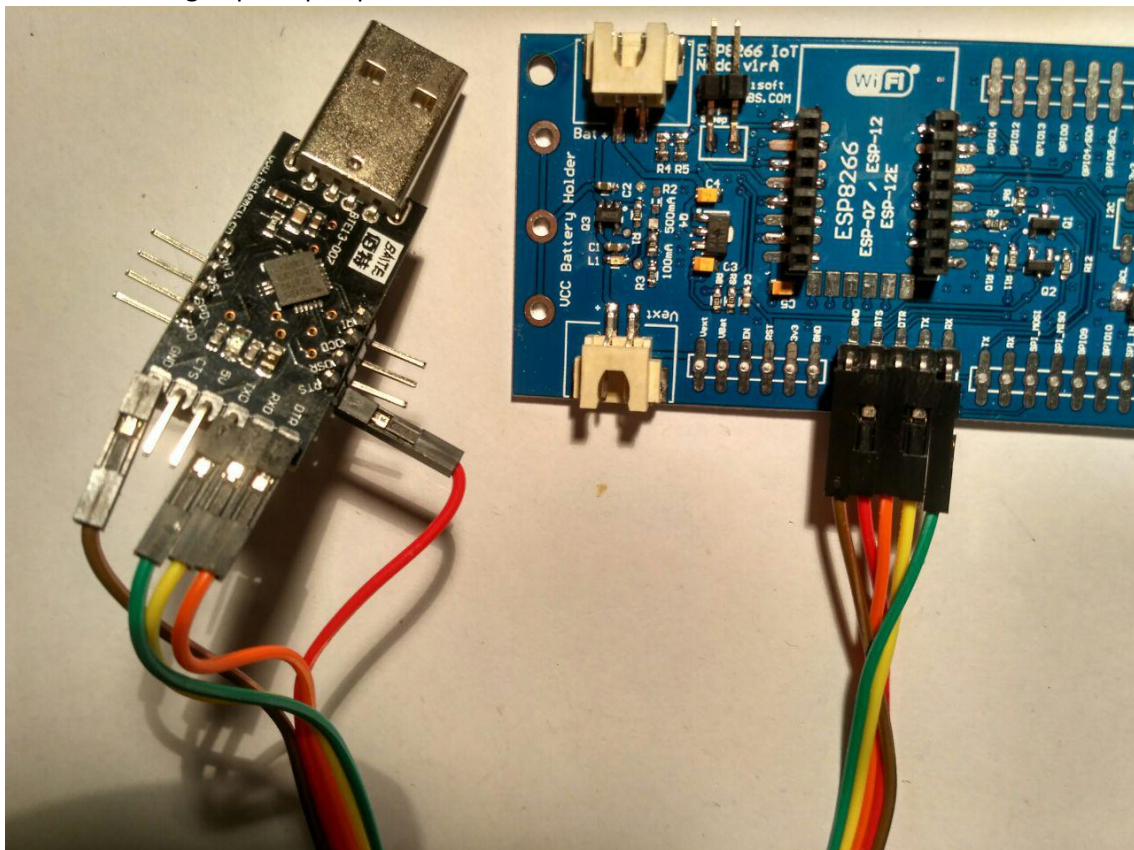
## Conexionado para programación mediante USB-Serie

Para realizar la programación del módulo ESP8266 en cualquiera de las 3 versiones que es compatible la PCB, podremos usar un conversor USB-Serie. Lo único que debemos tener en cuenta es que tenga accesibles ciertos pines el conversor.

Vamos a necesitar los siguientes pines para poder hacer la programación: GND, TXD, RXD, DTR, RTS.

**Nota:** Recordar que en el momento de conectar los pines RX y TX, debemos cruzarlos. Es decir, el RX del conversor USB-Serie tendrá que ir al TX de la PCB y lo mismo con el RX del conversor, que tendrá que ir al TX de la PCB.

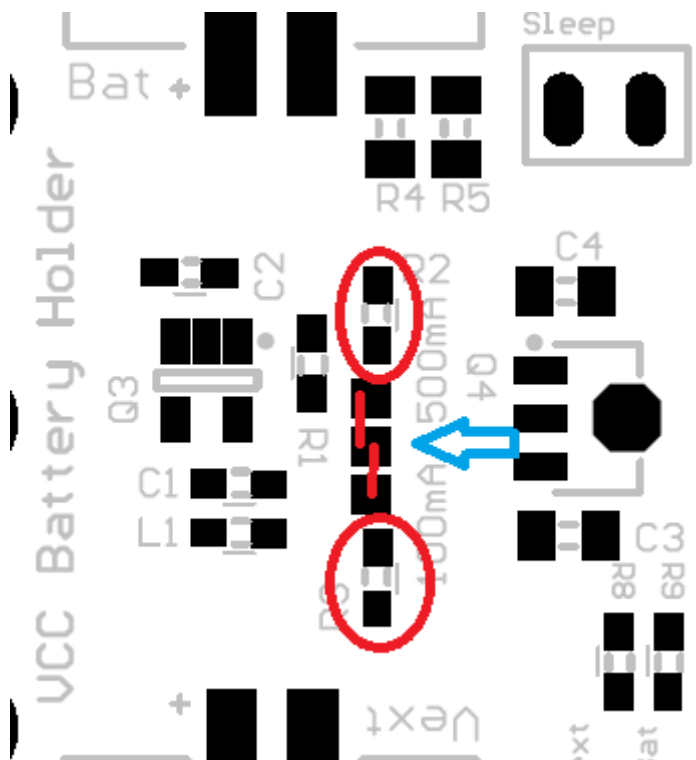
Añado una imagen para que quede más claro con los cables de diferente color.





## Ajustar corriente para la carga de la(s) batería(s)

En la PCB disponemos de un sitio para elegir que corriente de carga queremos usar. Esta elección es manual, mediante un puente donde con una resistencia de 0 ohm 0603/0805 o un puente de estaño, elegiremos que resistencia preconfigurada queremos usar para establecer la corriente de carga en la batería.



Las posiciones de las resistencias R2 y R3 son las que preconfiguran 2 valores distintos de carga. Siendo R2 el que con una resistencia de 2k2 ohm configura casi el máximo de corriente que puede cargar el IC encargado, correspondiente a unos 450mA. Mientras tanto, para un valor de 100mA usaremos un valor de 10k ohm. Si se quieren valores intermedios, es una simple regla de tres. Por ejemplo, para una carga de 250mA tendríamos que usar una resistencia de aproximadamente 5k6 ohmios.

En la imagen adjunta, se pueden apreciar 3 cuadros, estos se usan para seleccionar que valor de resistencia usar. Uniendo el pad centra con uno de los extremos, elegiremos una configuración u otra.