

Módulo.3: NodeMCU. Características generales



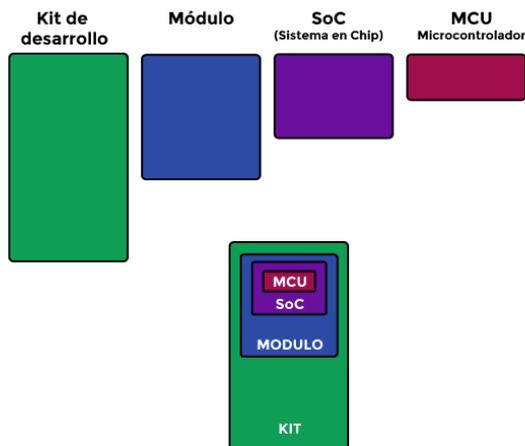
ÍNDICE

- 3 NodeMCU. Características generales2
- 3.1 Introducción.2
- 3.2 Kits de desarrollo basados en ESP82664
- 3.3 Características generales de las placas NodeMCU6
- 3.5 Modos de funcionamiento de módulo Wifi10
- 3.6 Información adicional y enlaces de interés10
- 3.7 Créditos y licencias11

3 NodeMCU. Características generales

3.1 Introducción.

A lo largo de los últimos años han ido apareciendo en el mercado numerosas placas de desarrollo de bajo coste basadas en microcontroladores (MCU). Las placas de desarrollo más populares son las placas Arduino en sus distintas versiones (UNO, Nano, Mega, Mkr, etc).



Fuente: <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>

Lo primero que conviene aclarar es la diferencia entre microcontrolador (MCU) y placa de desarrollo. Un microcontrolador es un pequeño circuito integrado que actúa como 'cerebro' y se encarga de ejecutar todas las instrucciones, controlar los puertos de E/S, la lectura y escritura en memoria, las comunicaciones, etc. Pero este MCU necesita para funcionar de otros circuitos integrados (chips), y de un soporte, es decir, otra placa donde alimentarse y conectarse con el mundo exterior: a esta placa la denominamos placa de desarrollo o kit de desarrollo.

La arquitectura hardware de las placas Arduino (el esquema de conexiones) y su software de programación (IDE de Arduino) son de código abierto. Esto significa que los esquemas son conocidos y eso ha dado pie a que proliferen muchos fabricantes que hagan sus propias versiones de placas de desarrollo. Así podemos encontrar kits de desarrollo basadas en MCU de empresas como Adafruit, Sparkfun o LoLin-Wemos.

Comprar placas de estos fabricantes nos da acceso a todos los recursos de sus páginas web donde se pueden encontrar librerías, ejemplos y mucha documentación de su comunidad de usuarios.

Otra alternativa más económica es comprar placas de desarrollo de fabricantes chinos que, fabrican las placas copiando los esquemas originales de terceros. Estas placas, son perfectamente válidas, pero hay que tener cuidado ya que, a veces, contienen pequeñas modificaciones de hardware (chips de comunicaciones CH40G en lugar del FTDI por ejemplo). Esto hace que haya que instalar drivers adicionales.

¿Qué nos aporta la placa NodeMCU? La familia de placas Arduino, UNO, Nano y Mega originales carecen de un chip de comunicaciones inalámbricas. Esto hace obligatorio comprar un escudo WiFi (Wi-Fi shield) si queremos tener comunicaciones Wi-Fi o acceso a Internet en nuestros proyectos. Encareciendo el precio del proyecto y añadiendo una placa adicional.



Para ocupar este hueco en el mercado la empresa china Espressif (<https://www.espressif.com/>), líder mundial en fabricación de Chips y módulos de comunicación Wi-Fi, desarrolló un 'Chip' que contenía un microcontrolador de 32bits (Tensilica L106), 50kb de memoria RAM, una entrada analógica de 10bits, 17 pines de entradas/salidas digitales y un módulo WiFi de 2.4Ghz. Había nacido el SoC (System on Chip) **ESP8266**.

Para poder utilizar este SoC en nuestros proyectos hay que comprar un módulo que lo integre y nos proporciones los pines necesarios para alimentarlo, programarlo y poder así usar sus periféricos. Estos son algunos de los más populares.

ESP01			ESP02	El módulo ESP01 es de los más veteranos y todavía sigue siendo una opción válida para pequeños proyectos donde necesitemos conexión a Internet o crear un pequeño servidor web y no queramos gastar mucho dinero.
ESP03			ESP04	
ESP05			ESP06	El inconveniente es que hay que configurarlo con comandos AT a través de un adaptador USB y alimentarlo a 3.3V. También se puede utilizar para configurarlo una placa Arduino UNO, aunque se desaconseja alimentarlo desde su salida de 3.3V.
ESP07			ESP08	
ESP09			ESP10	
ESP11			ESP12	



En los últimos años se ha popularizado mucho el uso del módulo ESP12 y el ESP12E, que son la base del kit de desarrollo NodeMCU.



El nombre NodeMCU se refiere al firmware de código abierto (Open Source Firmware) que tiene el módulo y también a las placas de desarrollo que incorporan el SoC ESP8266.

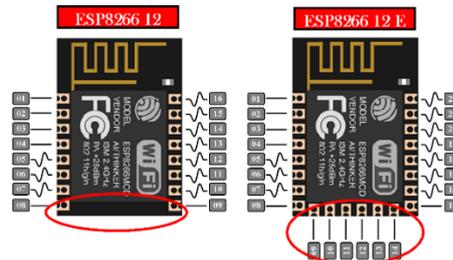
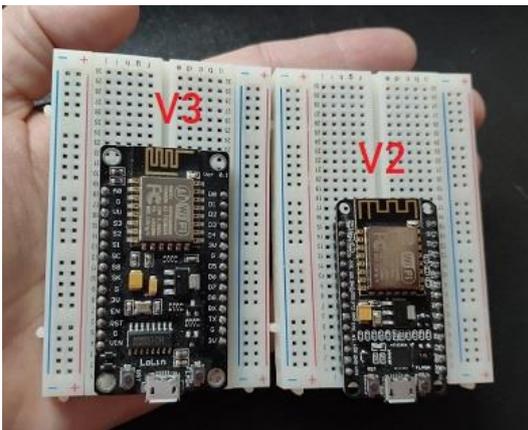
El ESP8266 fue lanzado al mercado en 2013 y son muchas las placas que usan este SoC al ser de código abierto. Suelen ser compatibles entre sí, comparten librerías y se programan con los mismos lenguajes de programación.

3.2 Kits de desarrollo basados en ESP8266

Placas NodeMCU y sus distintas versiones:

NodeMCU v0.9/V1/1ª Gen.	NodeMCU v1.0/V2/2ª Gen.	NodeMCU V1.0/V3/3ª Gen.
		

La V1 es la primera que salió al mercado, utiliza el módulo ESP12 original y actualmente es difícil de encontrar.

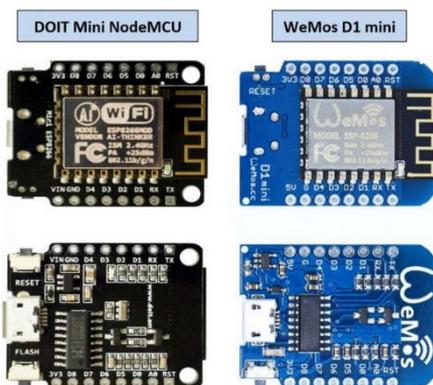


La versión V2 utiliza el módulo ESP12E (chip mejorado con 6 pines adicionales, cuatro dedicados a la comunicación SPI y dos nuevos GPIO).

Esta placa tiene la ventaja de encajar perfectamente en una protoboard y dejar libres una fila de pines para hacer nuestras conexiones. Se puede encontrar en tiendas on-line como Aliexpress o Amazon fabricadas por AMICA con chip USB-Serie CP2102. El precio oscila entre los 3€ y los 8€.

La versión V3 es, prácticamente la misma placa que la V2, ya que comparten el mismo módulo ESP12E pero es más ancha que la, la que más se vende en tiendas, la fabrica la empresa china LoLin-Wemos (<https://docs.wemos.cc/en/latest/>) y utiliza conversores USB-Serie del tipo CH340G. El precio es, prácticamente, el mismo que para la versión V2.

Placas Mini NodeMCU y Wemos D1.



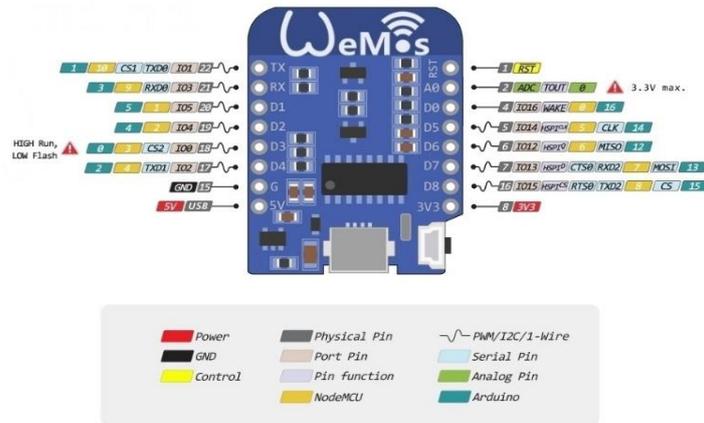
Para proyectos donde no dispongamos de mucho espacio una opción interesante son los kits de desarrollo **Mini-NodeMCU**, fabricado por la empresa DOIT, o la **Wemos-D1** de la empresa LoLin-Wemos en alguna de sus versiones (R2 o mini). Esta pequeña placa ocupa muy poco espacio, es compatible con los programas realizados para NodeMCU y se fabrican gran cantidad de escudos (shields) compatibles.

Web: https://wiki.wemos.cc/products:d1:d1_mini

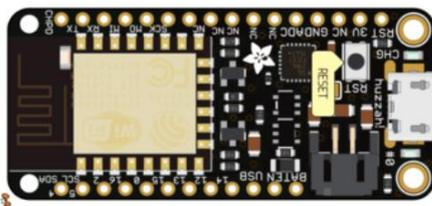
Esta pequeña placa es la preferida de los *makers* para realizar proyectos de IOT. Con los mismos 11 pines GPIO de propósito general que las NodeMCU dispone de *shields* para todo tipo de sensores y actuadores (Ildr, relés, dht11, Oled, motor, etc)

De forma muy resumida estas son algunas de las principales características:

- Velocidad: 80MHz/160MHz
- Flash: 4M bytes
- Tensión funcionamiento: 3.3V
- Entradas y salidas digitales: 11, todos (salvo el D0) con PWM, interrupciones, e I2C
- Entradas analógicas: 1 (Max. 3.2V)
- Conector Micro-USB



Placa Adafruit feather Huzzah



Esta popular placa desarrollada por la empresa Adafruit tiene la peculiaridad de incorporar un conector para una batería LiPo. El fabricante proporciona mucha información sobre esta placa en su web:

<https://learn.adafruit.com/adafruit-feather-huzzah-esp8266/overview>

Placa Sparkfun ESP8266 Thing

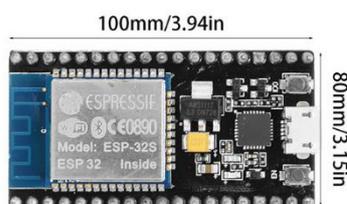
El fabricante Sparkfun ha desarrollado su placa de desarrollo añadiendo un interruptor (en el modelo Dev-kit) y también un conector de alimentación para una batería LiPo de 3.7V. La



empresa también desarrolla varios escudos (*shields*) para esta placa. Se puede encontrar mucha información sobre la placa en su web:

<https://www.sparkfun.com/products/13907>

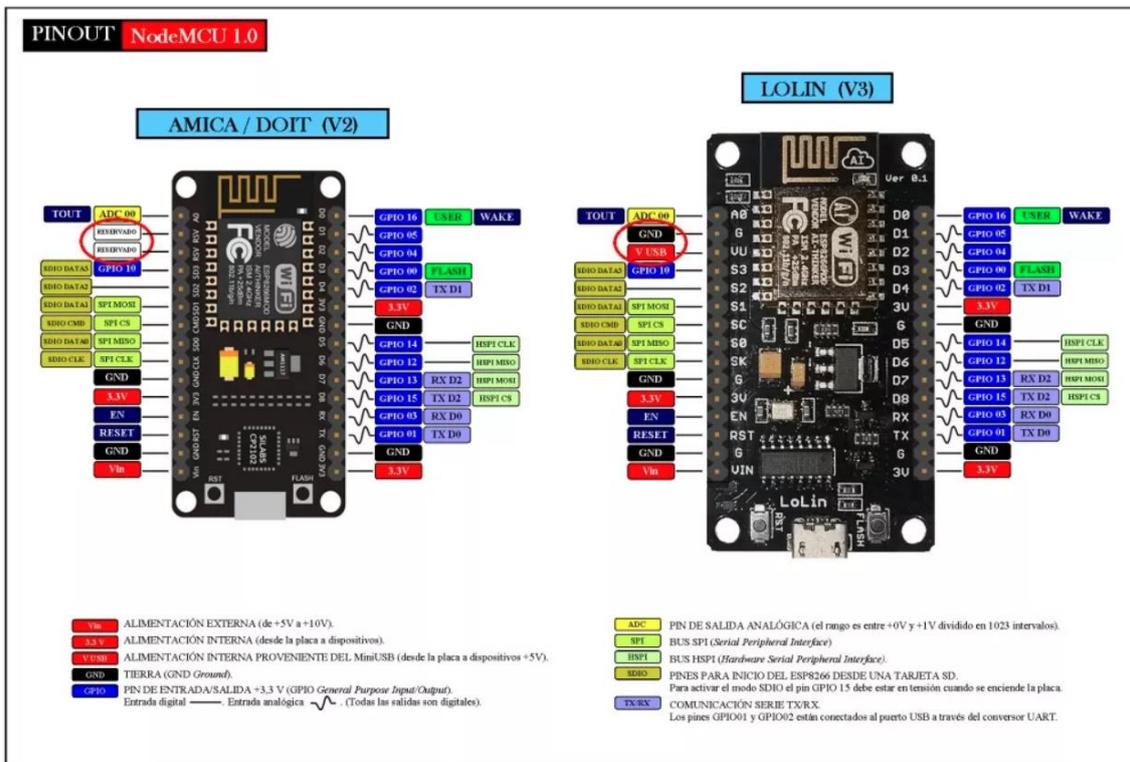
El nuevo SoC ESP32



Recientemente la empresa Espressif Systems, fabricante del ESP12x ha lanzado una actualización del mismo, el modelo ESP32 que incorpora, entre otras mejoras: más memoria, un procesador más rápido y un menor consumo. Se pueden encontrar kits de desarrollo de la propia Espressif o de otros fabricantes. Todos los ejemplos del curso han sido probados con las versiones de la placa NodeMCU V2/V3 y Wemos-D1 mini pero funcionarán igualmente en la versión ESP32 con pequeños cambios.

Toda la información sobre este nuevo SoC ESP32 se puede encontrar en la página oficial del fabricante: <https://www.espressif.com/en/products/hardware/esp32/overview>

3.3 Características generales de las placas NodeMCU



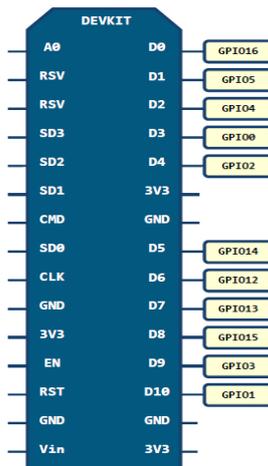
Como se puede observar, los cambios son mínimos entre ambas versiones. Solamente se han añadido dos pines adicionales en la rev3, uno de GND y otro de VUSB(5V). El SoC es el mismo (ESP12E) por lo tanto los programas realizados en la rev2 funcionarán perfectamente en la rev3. Hay que recordar que los chips USB-Serie son diferentes, por lo tanto habrá que instalar drivers distintos. Una vez instalado el driver y reconocida la placa, el ordenador le asignará un puerto COMx (en sistemas Windows). La conexión de datos y alimentación con el ordenador se hace a través de un conector micro-usb.

Las características de la placa NodeMCU son las siguientes:

- 1 pin analógico (A0)
- 11 pines digitales (D0 a D10)
- 1 pin de 5V en V2 y 2 pines en V3
- 3 pines de 3.3V
- 4 pines de GND en V2 y 5 pines en V3

El pin analógico A0. Se trata de una entrada analógica asociada a un ADC de 10 bits. En esta entrada podemos conectar un potenciómetro o cualquier sensor que nos proporcione una tensión de entre 0-3.3V. El rango de valores que obtendremos al hacer una lectura del ADC será de 0 (para 0 Voltios) a 1023 (para 3.3 Voltios). Muy importante no superar NUNCA la tensión máxima de 3.3V en esta entrada.

Los pines digitales (D0 a D10). Estos pines se pueden utilizar como pines de E/S digital de propósito general teniendo en cuenta que la lógica funciona a 3.3V.



Es decir, si un pin está configurado como salida, ponerlo a nivel Alto (HIGH) implica tener en el pin una tensión de 3.3V y ponerlo a nivel Bajo (LOW) implica tener en el pin una tensión de 0.0V.

Por el contrario, si un pin se configura como entrada, debemos asegurarnos de que las tensiones que recibe varían entre 0V (LOW) y 3.3V (HIGH).

El ESP12E no tiene memoria Flash para poder guardar los programas, por lo tanto, varios de sus pines GPIO se dedican al interface con la memoria Flash y no los podemos utilizar por eso no vienen sergrafados en la placa.

Los pines D9 y D10, que son los pines de Rx y Tx de la UART interna se utilizan para cargar los programas al módulo (pin Rx). Por lo tanto no resulta conveniente utilizarlos en nuestros proyectos. Si es imperativo utilizarlos para algún propósito (encender un led o leer el estado de un pulsador) habrá que habilitar algún medio para poder desconectar estos pines durante el proceso de carga del programa.

Además, es importante saber que algunos pines digitales pueden ser utilizados como salidas, pero **no como entradas**. Estos pines son D3, D4 y D8. La placa lee estos pines en el arranque y configura su modo de funcionamiento según la siguiente tabla.

	D3 (GPIO0)	D4 (GPIO2)	D8 (GPIO15)
Modo UART (Carga de programa)	LOW	HIGH	LOW
Modo FLASH (Ejecución de programa)	HIGH	HIGH	LOW
Modo arranque de la tarjeta SD	LOW	LOW	HIGH

Los pines de alimentación y Gnd. A través de Estos pines podemos alimentar sensores o generar un nivel alto/bajo para nuestros pulsadores. En la rev2 no disponemos de ningún pin que ponga 5V pero en el pin Vin disponemos de esos 5V, muy útiles para alimentar sensores que funcionen con esta tensión. NUNCA usar esa tensión para alimentar las entradas digitales del módulo.

LEDs de la placa. Dentro de la propia placa tenemos dos leds integrados (D0) y (D4). Ambos se pueden controlar desde el programa, igual que el led del pin 13 de la placa Arduino. Tan solo recordad que estos leds funcionan con lógica negativa, es decir, se encienden con un nivel bajo y se apagan con un nivel alto. El led en D4 parpadea cuando la placa está recibiendo un nuevo programa.

Los pulsadores RST y FLASH. El pulsador de reset (RST) se encuentra a la izquierda del conector micro-usb de la placa y resetea la placa si se pulsa. Es decir, se interrumpe la ejecución del programa y se vuelve a cargar desde el principio.

El botón FLASH está a nivel alto en reposo y se utiliza para poder hacer una actualización del “firmware” de la placa cuando se pulsa (nivel bajo).

Probablemente nunca lleguemos a pulsar este botón ya que, al igual que pasa con Arduino, la propia placa pone este pin a nivel bajo cuando actualiza el programa en la memoria Flash interna.

Resumen de pines:

Pin	GPIO	Input	Output	Comentarios
D0	GPIO16	No interrupciones	No PWM No I2C	HIGH durante boot Resistencia Pull-Down Conectar a RST para Wake-Up
D1	GPIO5	OK	OK	SCL (I2C) (frecuentemente)
D2	GPIO4	OK	OK	SDA (I2c) (frecuentemente)
D3	GPIO0	Pulled Up	OK	Boot falla si pulled LOW Conectado a botón FLASH
D4	GPIO2	Pulled Up	OK	HIGH durante boot Boot falla si pulled LOW Built-in LED TX1

Fuente: <https://www.luisllamas.es/esp8266-nodemcu/>

D5	GPIO14	OK	OK	SLCK (SPI)
D6	GPIO12	OK	OK	MISO (SPI)
D7	GPIO13	OK	OK	MOSI (SPI)
D8	GPIO15	Pulled GND	OK	CS (SPI) LOW durante boot Boot falla si pulled HIGH No tiene Pull-Up
RX	GPIO3	OK	RX	HIGH durante boot No usable si se usa UART
TX	GPIO1	TX	OK	HIGH durante boot Boot falla si pulled LOW Debug output en boot No usable si se usa UART
A0	ADC0	Analog Input	NO	

3.4 Entornos de programación

Todas las placas basadas en ESP8266 se pueden programar con alguno de estos lenguajes:

LUA. https://www.lua.org/about.html	
MicroPython. https://micropython.org/	
Arduino IDE. https://www.arduino.cc/	
ArduinoBlocks http://www.arduinoblocks.com/	

LUA es un lenguaje de programación multiparadigma, imperativo, estructurado y bastante ligero, que fue diseñado como un lenguaje interpretado con una semántica extensible. Fue popular hace algunos años por su reducido tamaño pero hoy en día no se utiliza demasiado.

MicroPython, como su nombre indica es una versión reducida del conocido lenguaje Python optimizado para poder ejecutarse en microcontroladores.

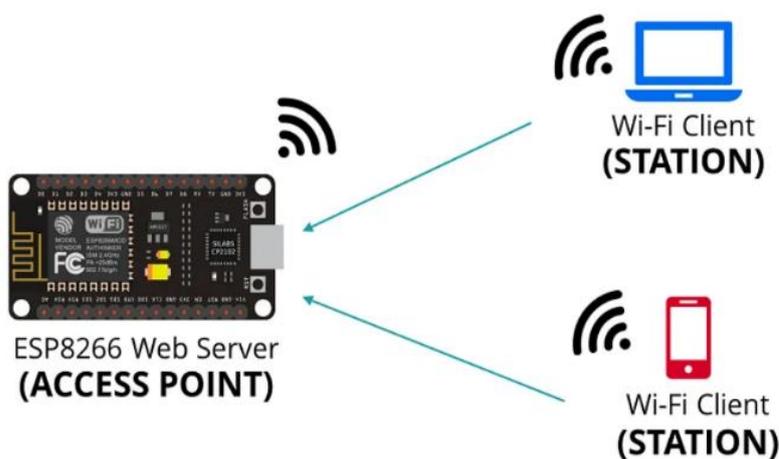
Arduino IDE, es la opción más utilizada para los que ya tienen experiencia en programar placas Arduino. Sólo hay que añadir nuestra placa de desarrollo desde el gestor de placas y trabajar con ella como si se tratara de un Arduino.

En este curso nos centraremos en la programación visual de la placa NodeMCU usando ArduinoBlocks. Se trata de un entorno visual de programación, fácil de usar para principiantes, pero con bloques potentes para manejar de forma sencilla, multitud de sensores y actuadores.

En el siguiente módulo del curso veremos las posibilidades del entorno, especialmente, en todo lo que tiene que ver con comunicaciones inalámbricas y aplicaciones de IOT.

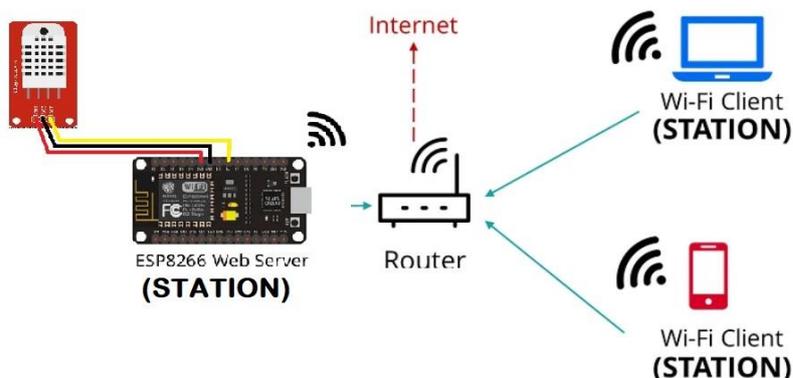
3.5 Modos de funcionamiento de módulo Wifi

Modo Access Point (Access Point). En este modo la placa genera una red Wi-Fi propia con una



SSID y una puerta de enlace predeterminada. Se puede programar un pequeño servidor web donde nos podremos conectar desde cualquier dispositivo para ver el estado de nuestros sensores o manejar dispositivos a distancia. Siempre dentro del alcance de la Wi-Fi del dispositivo.

Modo Estación (Station). En este modo la placa se conecta a la Wi-Fi de nuestro router y



podemos tener acceso a nuestros sensores y actuadores desde Internet y/o publicar nuestros datos a través de algún protocolo como MQTT. Todas las prácticas del curso las haremos con el módulo configurado de esta manera.

3.6 Información adicional y enlaces de interés

<https://www.espressif.com/en/home>

<https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>

<https://www.luisllamas.es/esp8266-nodemcu/>

<http://mrobotics.es/blog/category/nodemcu/>

<http://www.arduinoblocks.com/>

<http://arduinoblocks.didactronica.com/>

3.7 Créditos y licencias

Autor: Domingo Llorente Escobedo. Publicado por el Centro Regional de Formación del Profesorado de Castilla-La Mancha.

<http://centroformacionprofesorado.castillalamancha.es/comunidad/crpf>

Bajo licencia Creative Commons 4.0 con reconocimiento – No Comercial – Compartir Igual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Empleando imágenes y esquemas desarrollados con Fritzing bajo licencias GPLv3 y Creative Commons BY-SA en <http://fritzing.org/home/>