

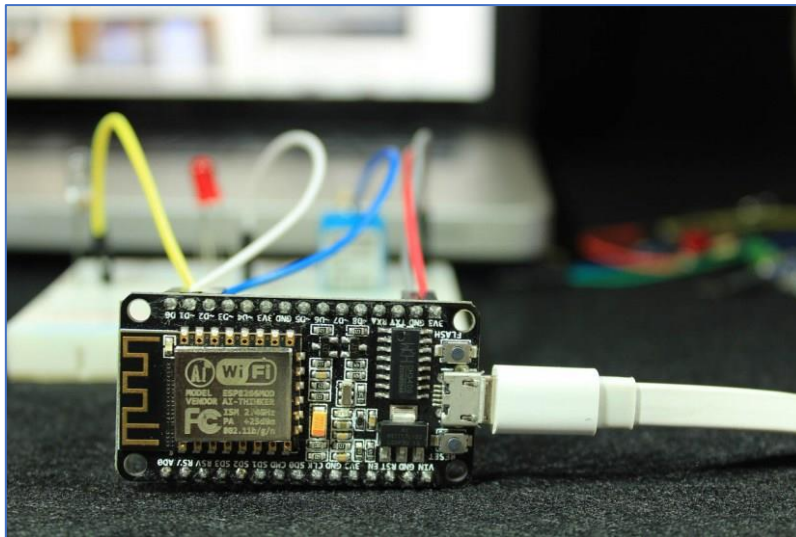
CARRERA: TECNICATURA SUPERIOR EN TELECOMUNICACIONES

MATERIA: ELECTRONICA MICROCONTROLADA

DOCENTES:

- JORGE E. MORALES, INGENIERO ELECTRICISTA ELECTRÓNICO (UNIVERSIDAD NACIONAL DE CÓRDOBA)
- C. GONZALO VERA, TECNICO SUPERIOR EN MECATRONICA (U.T.N.)

ES8266



CORDOBA, 29 DE JULIO DEL 2022.-

GRUPO N°2:

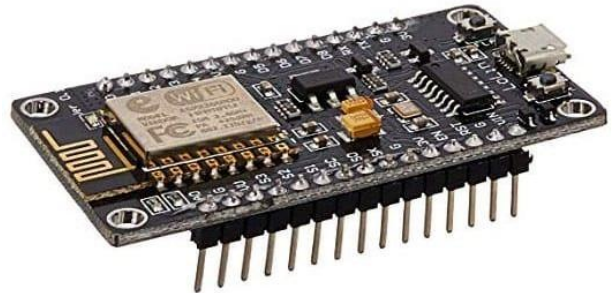
- DARIO ARRIOLA
- MARCOS JULIAN FINES
- DANIEL RODRIGUEZ
- NATALIA GALLIANI
- JEREMIAS CASTRO
- CARLA ARGENTINA WAYAR

I. APLICACIONES

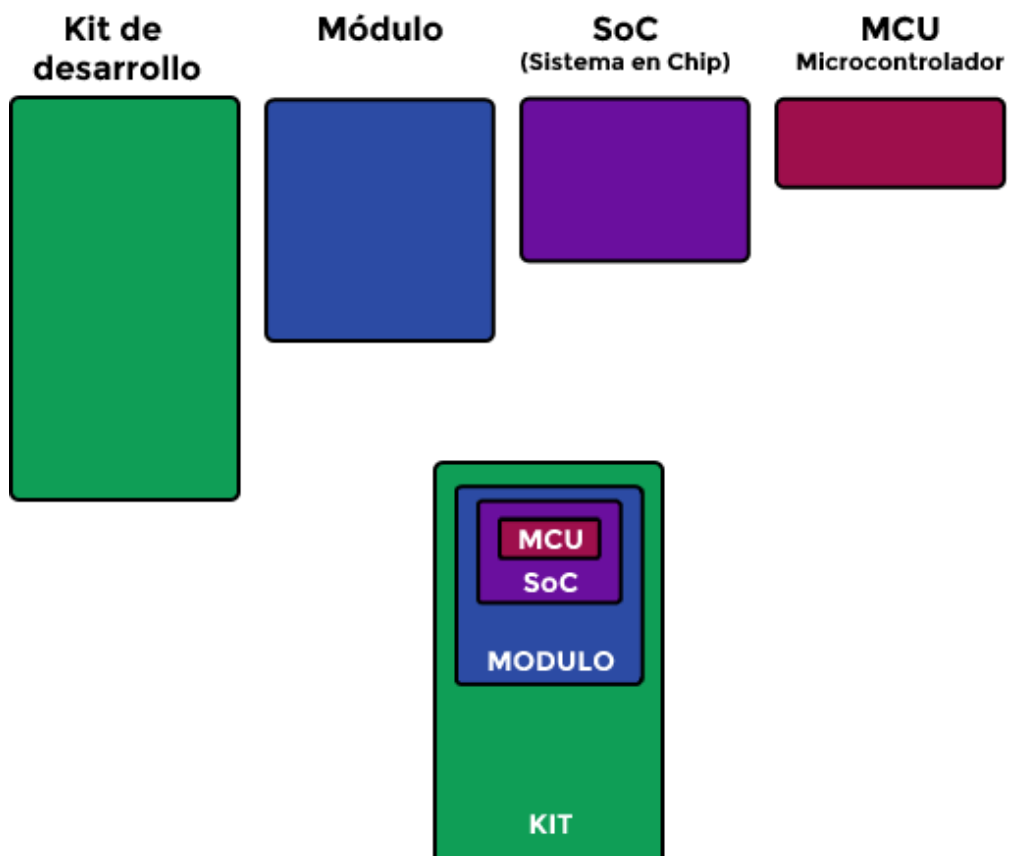
Este SoC es ampliamente usado en domótica, automatización, control industrial, redes de sensores, dispositivos de ubicación, servidores web. Tiene gran factibilidad para realizar IoT.

II. ESPECIFICACIONES:

- Alimentación de 5 [V]
- Pines de 3,3 [V]
- Soc ESP8266 (ESP-12)
- CPU TENSILICA con arquitectura de 32 bits
- Frecuencia de Reloj: 80MHz/160MHz
- Instruction RAM: 32KB
- Data RAM: 96KB
- Memoria Flash Externa: 4MB
- Pines Digitales GPIO: 17 (pueden configurarse como PWM a 3.3V)
- Pin Analógico ADC: 1 (0-1V)
- Wi-Fi Direct (P2P), soft-AP
- Stack de Protocolo TCP/IP integrado
- Conector Micro Usb
- Lenguajes de programación: LUA, C++, MicroPython,






Esquema de la composición de la placa Nodemcu



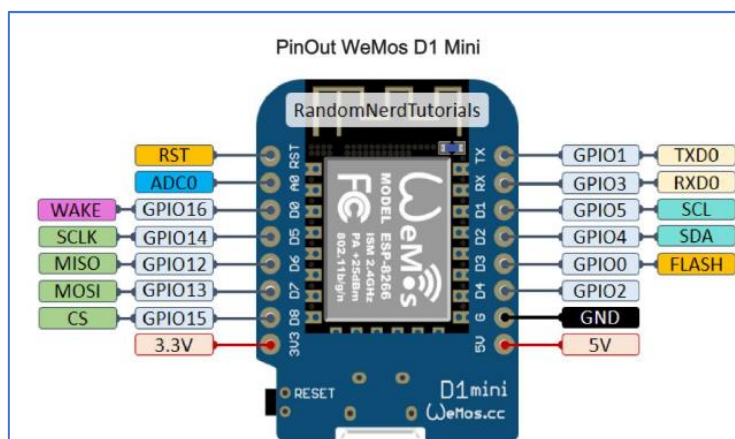
III. TIPOS DE VERSIONES:

El ESP8266 viene en una amplia variedad de versiones, que se diferencian en el número de GPIO, estilos de antena, tamaño, compatibilidad con protoboard, etc... tal como se muestra en la siguiente imagen:



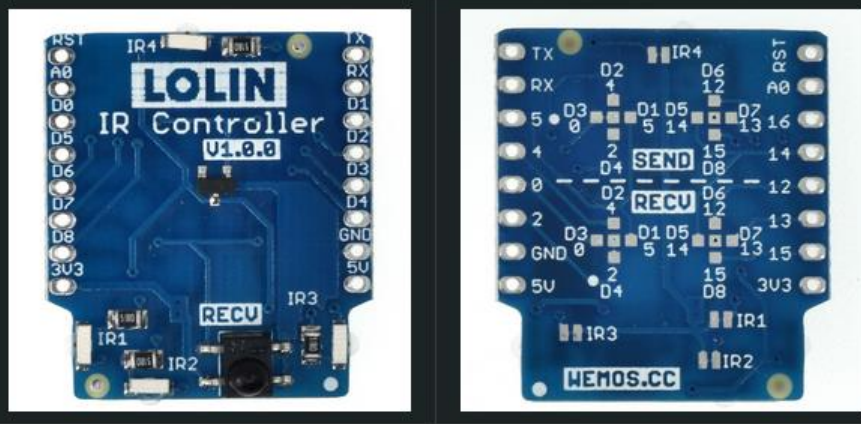
	ESP-01	ESP-12E NODEMCU	WEMOS D1 MINI
			
Enlaces	Banggood amazon.com aliexpress	Banggood amazon.com aliexpress	Banggood amazon.com aliexpress
GPIO	4 (incluyendo TX y RX)	11	11
Pines ADC	1	1	1
Memoria flash	1 MB (versión mejorada)	4 MB	4 MB
Apto para protoboard	X	✓	✓
Convertidor USB a Serie	X	✓	✓
Tamaño	24,75 mm x 14,5 mm (0,97" x 0,57")	48,55 mm x 25,6 mm (1.91" x 1")	34,2 mm x 25,6 mm (1.35" x 1")
Precio	ps	\$\$	\$\$

WeMos D1 Mini

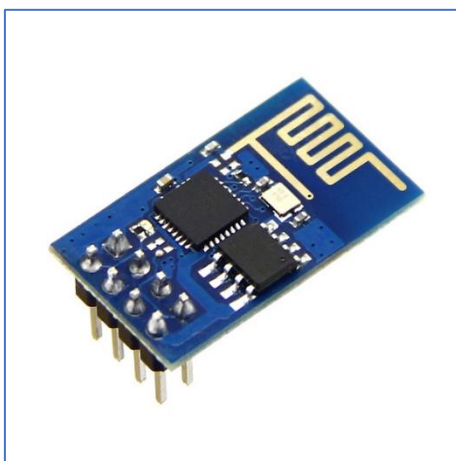


El WeMos D1 Mini ofrece memoria flash de 4 MB, 11 GPIO, 1 pin ADC en una configuración mínima y pequeña. Se han desarrollado una amplia variedad de shields para la miniplaca D1, que le permite crear configuraciones pequeñas y sencillas casi sin necesidad de cableado.

IR Controller Shield



ESP-01

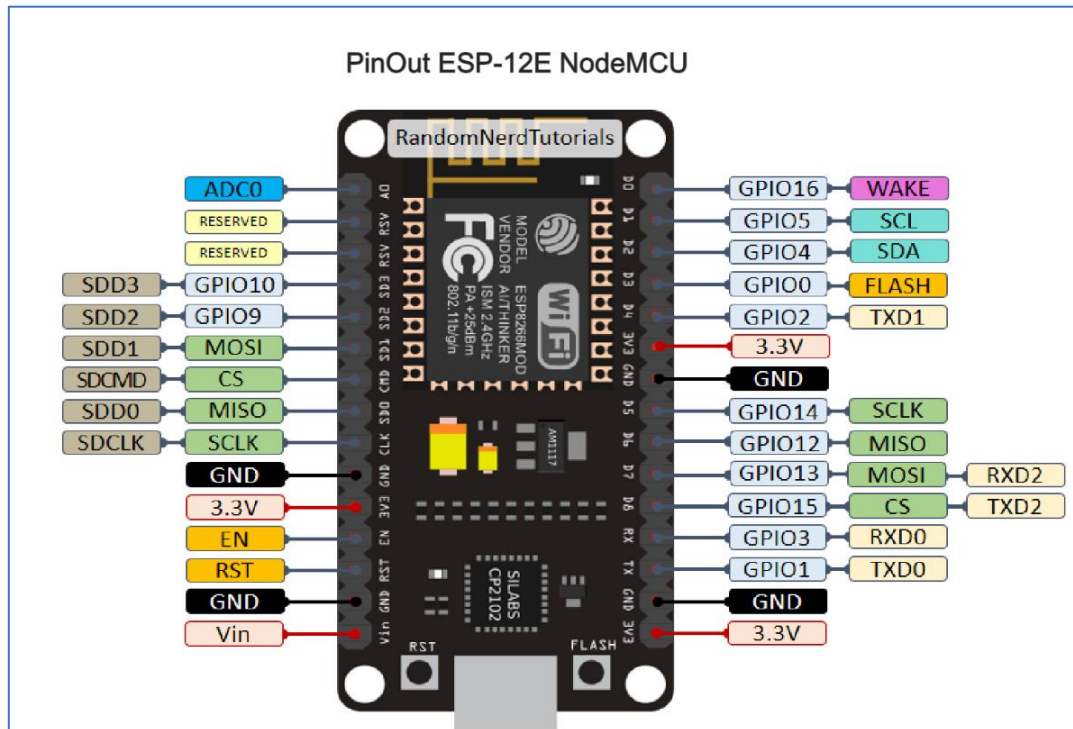


La versión mejorada cuenta con una memoria flash de 1 MB (la versión anterior tenía 512 kB). Además, ofrece cuatro GPIO para controlar y conectar periféricos (dos de los cuales son Tx y Rx para comunicación en serie). Por lo que, si necesitas más periféricos en tus proyectos, es mejor utilizar una de las placas anteriores.

Desafortunadamente, la placa no tiene un regulador de voltaje incorporado, por lo que debe usar una fuente de alimentación de 3.3 [V] o agregar un regulador de voltaje para reducir el voltaje de entrada a 3V3 (se encuentra la versión PRO que sí se alimenta con 5[V]). Además, no viene con un convertidor de USB a serie, lo que significa que necesita un programador FTDI para cargar el código en la placa.

El ESP-01 es excelente para aplicaciones que requieren una pequeña cantidad de periféricos.

IV. PINES (versión que tiene el grupo):



Debe tenerse en cuenta que los siguientes pines están, o pueden estar, en HIGH durante el booteo, por los periféricos que puedan estar conectados:

GPIO16: pin is high at BOOT

GPIO0: boot failure if pulled LOW

GPIO2: pin is high on BOOT, boot failure if pulled LOW

GPIO15: boot failure if pulled HIGH

GPIO3: pin is high at BOOT

GPIO1: pin is high at BOOT, boot failure if pulled LOW

GPIO10: pin is high at BOOT

GPIO9: pin is high at BOOT

El pin analógico es solo uno, de lectura, el A0.

Tiene pines PWM en todos los pines de entrada/salida: GPIO0 to GPIO15.

Cualquier pin puede ser utilizado como pin de interrupción, a excepción del GPIO16.

V. ENTORNO DE PROGRAMACION:

Hay varias formas de programar el ESP8266. A menudo usamos Arduino IDE o MicroPython.

LENGUAJES DEL ESP8266

Este integrado puede ser programado mediante LUA, que no tuvo demasiada aceptación, y mediante Arduino basado en C++. El IDE de Arduino es compatible con ESP8266. Para esto, es necesario seguir unos simples pasos, como se indica en la carpeta Ejemplos_ESP8266.

Si los programas van a tener cierta extensión, entonces VS Code es preferible. Basta instalar VS Code, Python 3 (o superior) y PlatformIO. Es muy sencillo y se realiza como indica el tutorial (<https://randomnerdtutorials.com/vs-code-platformio-ide-esp32-esp8266-arduino/>)- (VS Code y PlatformIO para ESP8266).

VI. MODO SUSPENSIÓN

Dependiendo de la aplicación que realicemos nos interesará que el circuito quede en modo sleep o deep sleep para evitar que consuma energía inútilmente.

El ESP8266 saldrá de la suspensión después de un tiempo fijado (programado). Para esto se conecta RST a D0 y se usa la función ESP.deepSleep(tiempo);, donde tiempo es una cantidad entera. Luego de cargar el código, basta presionar RST para que comience a dormir. De otro modo, basta presionar RST para que salga del modo sleep.

Item	Modem-sleep	Light-sleep	Deep-sleep
Wi-Fi	OFF	OFF	OFF
System clock	ON	OFF	OFF
RTC	ON	ON	ON
CPU	ON	Pending	OFF
Substrate current	15 mA	0.4 mA	~20 uA
Average current (DTIM = 1)	16.2 mA	1.8 mA	—
Average current (DTIM = 3)	15.4 mA	0.9 mA	—
Average current (DTIM = 10)	15.2 mA	0.55 mA	—

Notar que en el mejor caso, **consume 20 [μA]**.