SOC ESP8266

Carrera: Tecnicatura Superior en Telecomunicaciones

Materia: Electrónica Microcontrolada

Actividad: Actividad N°2 – ESP8266

Docentes:

- Morales, Jorge E.
- Vera, C. Gonzalo

GRUPO 6: https://github.com/EMTSTISPC/Grupo6.git

Integrantes:	GitHub:
- Guzmán, Lilén	https://github.com/lilenguzman01
- López, Maximiliano	https://github.com/Maxilopez28
- Moyano, Emilio	https://github.com/TerraWolf
- Muguruza, Sergio	https://github.com/sergiomuguruza
- Ramos, Marina	https://github.com/MarinaSR11
- Ripoli, Enrique	https://github.com/enriqueripoli

ESP8266:

Se trata de un chip integrado con conexión WiFi y compatible con el protocolo TCP/IP. El objetivo principal es dar acceso a cualquier microcontrolador a una red. Debido a eso, el uso que le demos dependerá de si lo tenemos como chip o como módulo. Dentro de la gran cantidad de usos caben destacar los siguientes:

- Electrodomésticos conectados.
- Automatización del hogar.
- Casas inteligentes.
- Automatización de la industria.
- Monitor de bebés.
- Cámaras IP.
- Redes de sensores.
- Wereables.
- IoT (Internet of Things o Internet de las Cosas)
- IIoT (Industrial Internet of Things o Internet de las Cosas para el sector Industrial)

Y cualquier aplicación donde se requiera conexión a una red o a Internet.

El ESP8266 viene con las siguientes características:

- Wi-Fi de 2,4 GHz (802.11 b/g/n, compatible con WPA/WPA2)
- 16 Entradas/Salidas de propósitos generales
- Protocolo de comunicación serie Inter-Integrated Circuit (I^2C)
- Conversores de analógico a digital (ADC de 10 bits)
- Protocolo de comunicación serie SPI (Serial Peripheral Interface)
- Interfaces I^2S (Inter-IC Sound) con DMA (Direct Memory Access) (compartiendo pines con GPIO)
- UART (en pines dedicados, más un UART de sólo transmisión que se puede habilitar en la GPIO2)
- Modulación de ancho de pulso (PWM).

Modos de operación

Debido a los sectores a los que va enfocado, wereables, dispositivos del IoT y móviles, el ESP8266 requiere de una gestión de energía eficaz. Dispone de una arquitectura de bajo consumo que trabaja en 3 modos.

- ***Active mode o modo activo***: a pleno rendimiento.
- ***Sleep mode o modo dormido:*** solo el RTC (Real Time Clock) está activo para mantener la sincronización. Se queda en modo alerta de los posibles eventos que le hagan despertar. Mantiene en memoria los datos de conexión y así no hace falta volver a establecer la conexión con la WiFi. Consume entre 0,6 mA y 1 mA.
- ***Deep sleep o modo en sueño profundo***: el RTC está encendido pero no está operativo. Debe pasar por el modo dormido antes de despertar. Hay que tener especial cuidado con los datos ya que en este estado es como si estuviera apagado y todos los datos que no estén almacenados se pierden. Consume alrededor de 20 µA.

Módulos ESP-XX

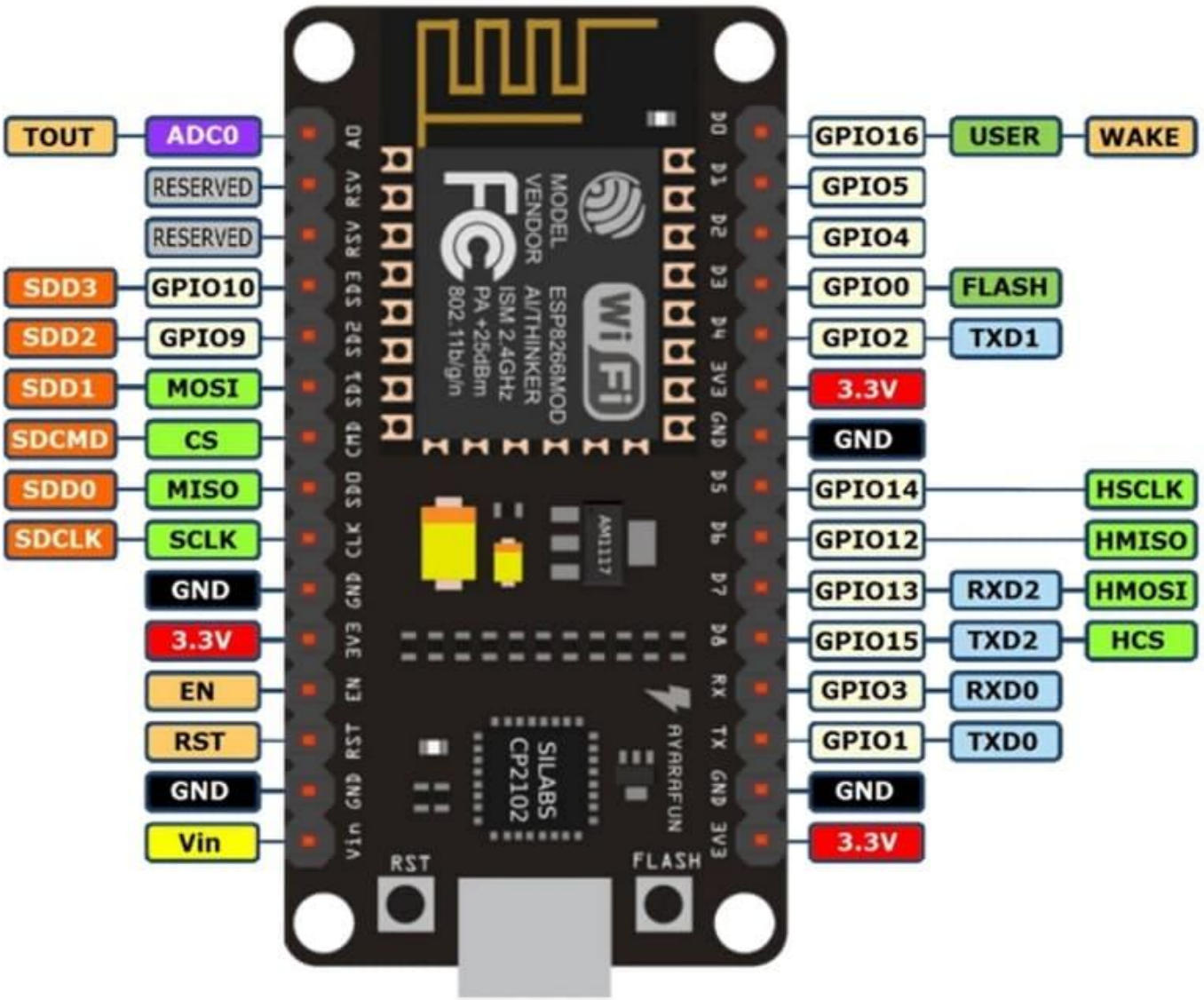
Al igual que con Arduino, donde trabajamos con la placa o circuito integrado, con el ESP8266 ocurre exactamente lo mismo. El fabricante AI-Thinker proporciona la serie ESP con diferentes modelos para diferentes usos. A parte han ido surgiendo diferentes placas que incorporan algún módulo ESP como el **NodeMCU**.

Algunas de las versiones más utilizadas del ESP8266 son:

- ESP8266-12E NodeMCU Kit
- WeMos D1 Mini
- ESP01

Distribución de los pines de un NodeMCU (ESP8266E-12):

La siguiente imagen muestra el pinout para la placa NodeMCU. Una placa NodeMCU tiene 30 pines. En este, 8 pines están relacionados con la alimentación y 2 están reservados. Los 20 pines restantes están asociados con los pines del módulo ESP-12E.

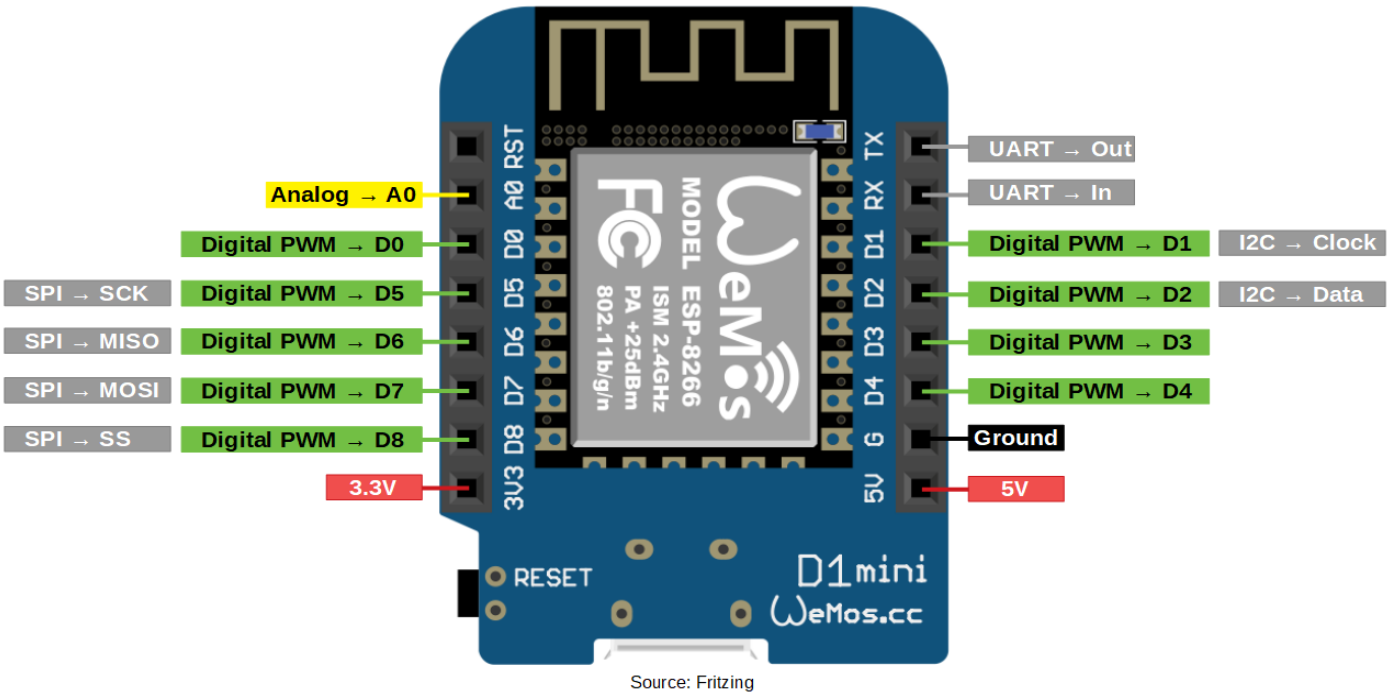


Caracteristicas:

- Voltaje de Alimentación (USB): 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC
- SoC: ESP8266 (Módulo ESP-12E)

- Voltaje de Alimentación (VIN): 5V a 9V Max.
- USB UART CH340
- CPU: Tensilica Xtensa LX3 (32 bit)
- Frecuencia de Reloj: 80MHz/160MHz
- RAM: 32KB
- SRAM: 64KB
- Memoria Flash Externa: 4MB
- Wi-Fi 2.4 GHz, 802.11 b/g/n
- Modos de operación: STA/AP/STA+AP
- Mecanismo de seguridad: WPA / WPA2

Distribución de los pines de un WeMos D1 Mini (ESP8266):



Source: Fritzing

Existen diferentes versiones de este chip, pero por lo general cuentan con:

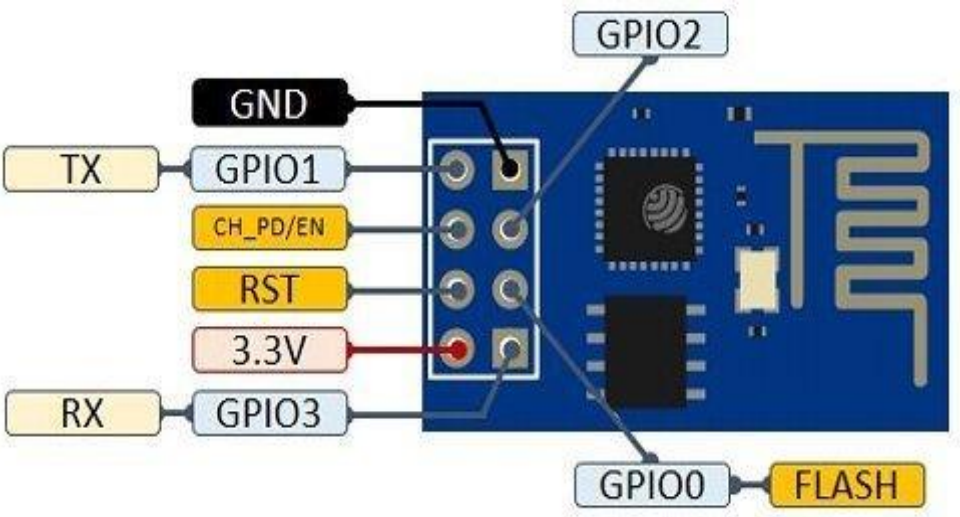
- 8 canales digitales de Entrada/Salida, todos capaces de producir señales PWM.
- 1 interfaz UART.
- 1 interfaz SPI.
- 1 interfaz I2C.
- 1 pin analógico.

- Entradas y Salidas de 3,3V DC y de 5V DC, que pueden dar hasta 500mA.
- Entrada de 5V DC.

Como también está basado en el ESP8266, es posible utilizar los pines de igual manera que en el NodeMCU. De hecho, el WeMos D1 Mini es como una pequeña versión del NodeMCU con menos pines, pero aun así suficiente para proyectos básicos.

Distribución de los pines de un ESP-01:

El ESP-01 es el primer chip basado en el ESP8266 en salir al mercado, es muy pequeño ya que mide 24 mm x 14 mm. La versión mejorada del ESP-01 cuenta con una memoria flash de 1 MB (la versión anterior tenía 512 kB). No viene con un convertidor de USB a serie, lo que significa que necesita un programador FTDI para cargar el código en la placa. Otras características que tiene son:



- 4 canales digitales de Entrada/Salida.
- 1 interfaz UART.
- Frecuencia del reloj interno entre 80MHz y 160MHz.
- 1 procesador Tensilica Xtensa LX3 de 32 bits.
- Entradas y Salidas de 3,3V DC.
- Entrada de 3,3V DC (**IMPORTANTE:** no se debe usar ninguna fuente de alimentación que supere los 3,6V DC porque corre el riesgo de dañar el chip).

IDE y lenguajes:

- Arduino(C++):

  Para programar un ESP32 desde Arduino hay que agregar la URL de las placas ESP32 para poder descargar el núcleo (o core) de ESP32 para Arduino.
  URL: https://dl.espressif.com/dl/package_esp32_index.json
  Una vez hecho eso. Para instalar el soporte para ESP32 y las placas de desarrollo hay que ir a "Herramientas>Placas>Gestor de Tarjetas". Esto abrirá el gestor de placas o tarjetas. En cuanto se inicie, comenzará a actualizar su base de datos, utilizando las URLs que se agregaron anteriormente en preferencias. Una vez termine, hay que escribir "esp32" en la barra de búsqueda para filtrar las placas disponibles y seleccionar la placa a utilizar

- Thonny (micro Python)

  Thonny viene con Python 3.7 incorporado, por lo que solo se necesita un instalador simple y está listo para aprender a programar. (También puede usar una instalación separada de Python, si es necesario). La interfaz de usuario inicial está desprovista de todas las funciones que pueden distraer a los principiantes.

  Dentro de la página sugerida para buscar información, encontramos lo siguiente:
  "*Experimentamos con varios IDE para programar las placas ESP32 y ESP8266 usando Micro Python, y Thonny parecía una buena opción. Aunque hay algunos errores, se actualiza y mejora constantemente.*

  Te permite programar tus placas ESP32 y ESP8266 con Micro Python, y es compatible con Windows, Mac OS X y Linux. Incluso viene instalado por defecto en el sistema operativo Raspberry Pi. Además, es fácil de instalar, por lo que no deberías tener problemas con el proceso de instalación."

  Fuente: https://randomnerdtutorials.com/getting-started-thonny-micropython-python-ide-esp32-esp8266/

- VSC (C++, Micro Python)

  Visual Studio Code (VSCode) es un editor de código sumamente versátil y potente que puede emplearse para programar en una multitud de lenguajes sobre distintas plataformas. Es altamente configurable y sus capacidades se pueden ampliar agregando distintas extensiones Es muy recomendable para programar nuestras placas ESP32 (y ESP8266). Con VS Code podemos escribir código en una amplia variedad de lenguajes aprovechando distintas funcionalidades como coloreo de sintaxis, autocompletado, depuración y optimización de código. También integra soporte para GIT, lo que nos permite llevar de manera ordenada las distintas versiones de los proyectos en los que trabajamos.