

# Arduino y la ESP32

Introducción al Internet de las Cosas



Marco Teran  
Universidad Sergio Arboleda

2023

# Contenido

- 1 Introducción
- 2 Introducción al proceso de prototipado electrónico
- 3 Introducción a Arduino
  - Arduino UNO
- 4 Programación del Arduino
- 5 Familia ESP
  - ESP-01
  - ESP8266 ESP-12
  - ESP-32
- 6 Agregar tarjetas
- 7 Laboratorio 01: Hola mundo

# Introducción

# Computación física

## Computación física

La computación física es el desarrollo de sistemas embebidos que **interactúan** con el mundo físico mediante **sensores** y **actuadores**. Se combinan *hardware* y *software* para crear soluciones interactivas en aplicaciones como robótica, arte interactivo y monitoreo ambiental.

- Ofrece una experiencia tangible y significativa para el usuario y juega un papel importante en la creación de soluciones innovadoras.

**Interactive  
System**

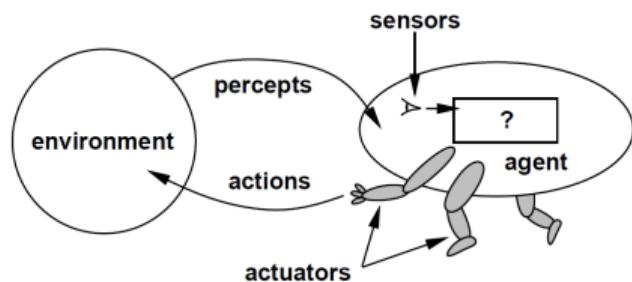
**Sensors**



**Real World**

**Actuators**

# Computación física



- Un **agente inteligente** es un sistema embebido que combina inteligencia y capacidades de *hardware* para interactuar con el mundo físico y realizar tareas específicas.
- Un sistema embebido se utiliza para realizar proyectos de computación física.
  - Tomar información del mundo que nos rodea utilizando entradas (sensores)
  - Responder a esa información con algún tipo de salida
- El sistema embebido puede actuar como el **cerebro** de este tipo de sistema

# **Open-source hardware**

# Open-source hardware

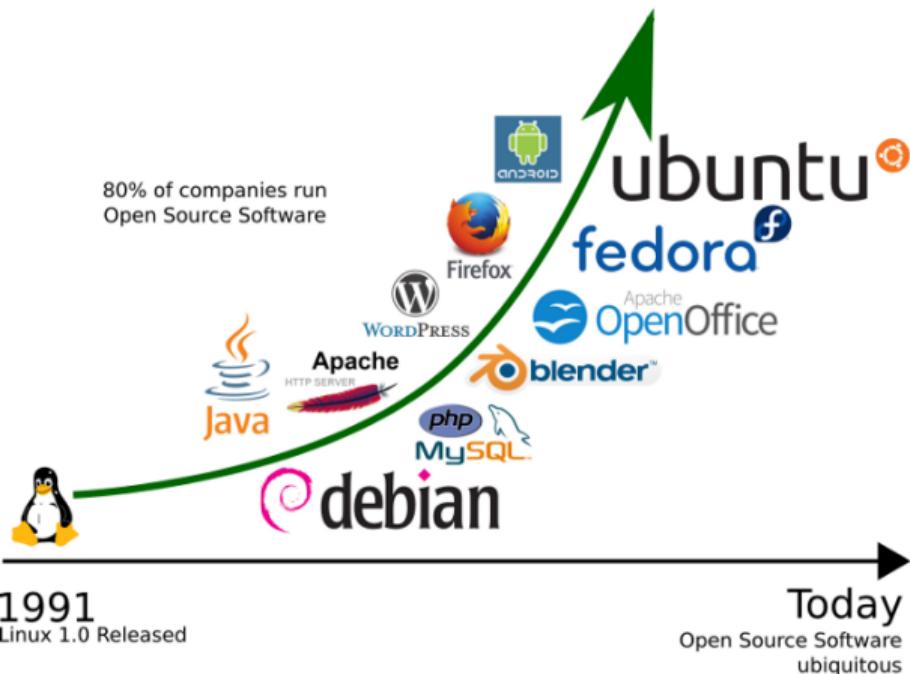


## Hardware de código abierto

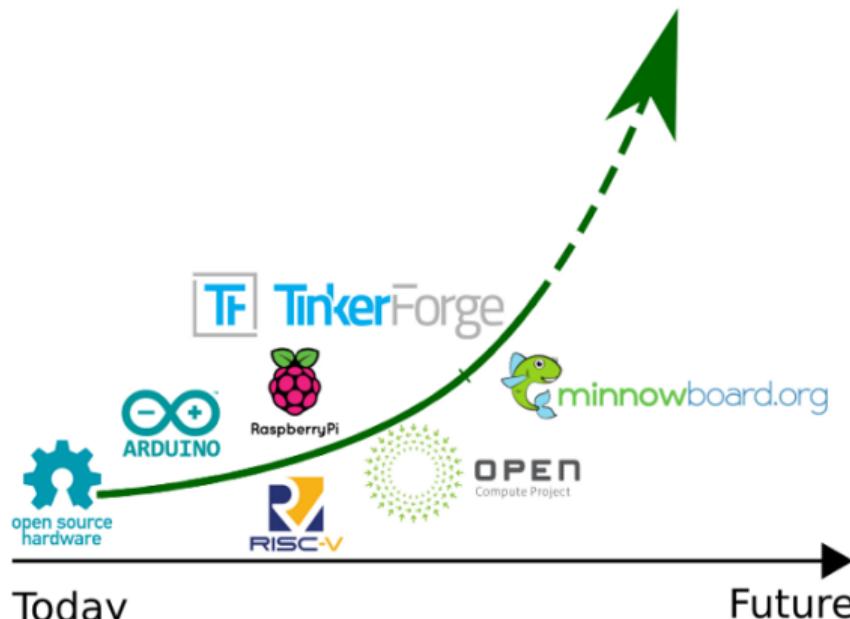
El Open-source hardware (OSHW) es un movimiento que promueve la colaboración y el acceso abierto a la documentación y los diseños de hardware.

- Se basa en la filosofía del **software de código abierto** y busca fomentar la innovación, la educación y la colaboración en el desarrollo de *hardware*.
- Esto permite a los usuarios modificar, mejorar y compartir los diseños
- Impulsa una mayor transparencia y una economía más circular en el sector del *hardware*.

# Open Source Software



# Open Source Hardware



Source: Global Semiconductor Alliance. "Charting a New Course for Semiconductors – Rambus and GSA Report", <https://www.gsaglobal.org/gsa-resources/publications/charting-a-new-course-for-semiconductors/>  
Accessed: 2016 Mar 03

# Open-source hardware

- Al elegir *hardware de código abierto*, los usuarios pueden tener un mayor control sobre su tecnología y pueden crear soluciones personalizadas y sostenibles.
- En el movimiento OSHW, los diseñadores comparten su *hardware* y *software* para fomentar el desarrollo de nuevos proyectos e ideas
- Los **diseños de origen** se comparten en un formato que puede ser modificado, siempre que sea posible

# Open-source hardware

- Los proyectos de OSHW hacen hincapié en la importancia de la documentación y el intercambio, lo que convierte a la comunidad de usuarios en un gran recurso para los estudiantes
- El movimiento de hardware de código abierto facilita el desarrollo de nuevos productos y diseños.

# Características del OSHW

- **Accesibilidad:** Permite el acceso abierto a los diseños y documentación del *hardware*.
- **Colaboración:** Se basa en la colaboración y el trabajo en equipo para mejorar y desarrollar *hardware*.
- **Innovación:** Permite un mayor potencial para la innovación en el desarrollo de *hardware*.
- **Transparencia:** El código y los diseños están disponibles *públicamente*, lo que aumenta la transparencia y confianza en el producto.
- **Modificación:** Los usuarios pueden modificar y personalizar el *hardware* de acuerdo a sus necesidades.

# Características del OSHW

- **Sostenibilidad:** Los usuarios pueden crear soluciones sostenibles y ecológicas.
- **Costo:** Puede ser más asequible que el **hardware propietario** debido a la ausencia de licencias y restricciones.
- Educación: El acceso abierto a los diseños y documentación permite una mayor educación en el desarrollo de *hardware*.
- **Libertad:** Los usuarios tienen un mayor control sobre su tecnología y no están limitados por restricciones de propiedad intelectual.
- **Comunidad:** Fomenta una comunidad activa de desarrolladores y usuarios que trabajan juntos para mejorar y desarrollar *hardware*.

# Open-source hardware

- **Hardware abierto** y **software abierto**, comparten enfoques similares en colaboración y acceso a código.
- El **hardware abierto** se refiere a diseños físicos, mientras que el **software abierto** se refiere a programas digitales.
- La fabricación y ensamblaje son más difíciles en *hardware abierto*.
- La documentación detallada y estandarización son esenciales en *hardware abierto*.
- La compatibilidad y escalabilidad son desafíos clave en *hardware abierto*.

# Open-source hardware

- La propiedad intelectual es más compleja en *hardware* abierto.
- El ciclo de vida del producto es más largo en *hardware* abierto.
- La comunidad y colaboración son cruciales para el éxito en *hardware* abierto.
- El *hardware* abierto puede tener impacto ambiental positivo y ser más seguro que el propietario.
- El *hardware* abierto puede ser más eficiente, asequible y una fuerza para la innovación sostenible.

# Paginas web de OSHW

- **Open-source hardware Association (OSHWA):** <https://www.oshwa.org/>
- **OSH Park :** <https://oshpark.com/>
- **GitHub:** <https://github.com/>
- **The Hackster:** <https://www.hackster.io/>
- **Instructables:** <https://www.instructables.com/>
- **hackaday:** <https://hackaday.io/>
- **Adafruit Industries:** <https://www.adafruit.com/>
- **SparkFun Electronics:** <https://www.sparkfun.com/>
- **Arduino:** <https://www.arduino.cc/>
- **Tindie:** <https://www.tindie.com/>

BOAD  
ADAPTER Matt  
Mar 1SP 2013,  
DEV A

LED2

Microm

Switch

SPI

JP2

D1B

GND

UCC

D0

FTTIN86

D1

SH

ON

GND

D2

D3

D4

D1

D2



P2

R2

# OSH Park

## Shared projects



### Tap Tempo Cardinal Tremolo

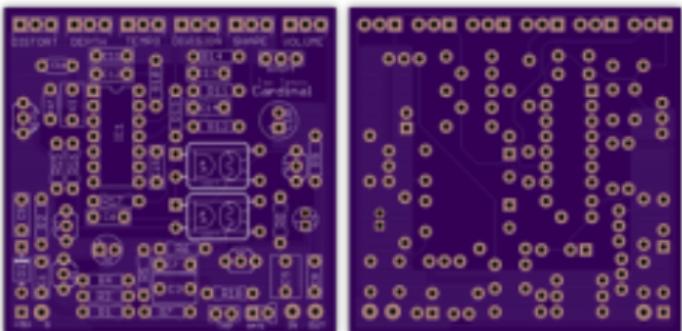
by  [midwayfair](#).

**2 layer board of 2.17x2.15 inches (55.25x54.56 mm).**

Shared on July 31st, 2013 14:42.

Harmonic tremolo with tap tempo

[Order](#). [Download](#). [Share](#).



### Wah

by  [culturejam](#).

**2 layer board of 1.58x1.42 inches (40.06x36.07 mm).**



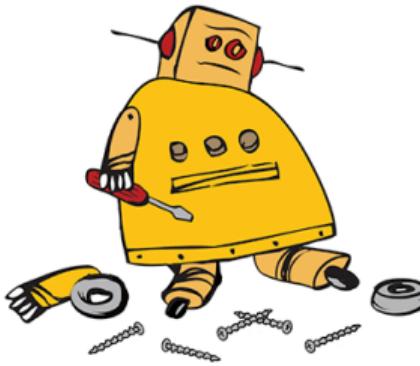
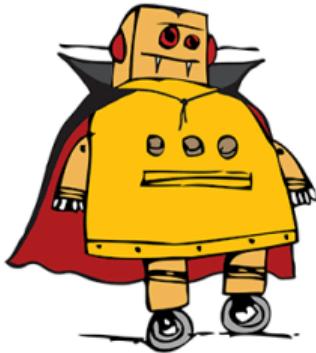
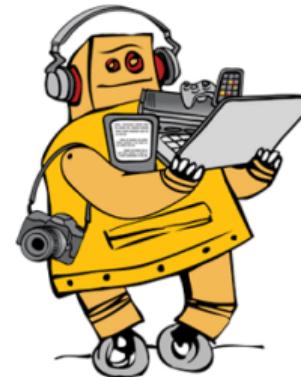
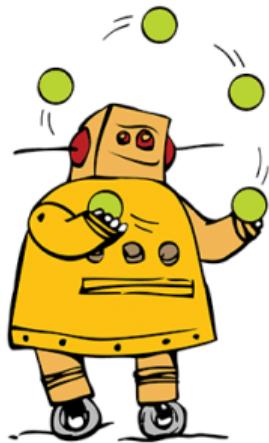
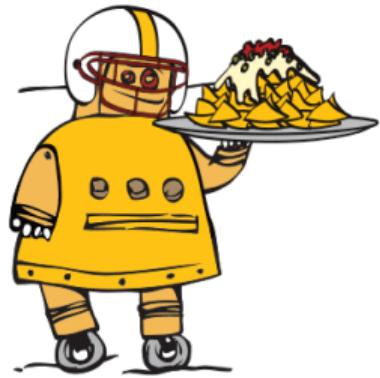


Explore **GitHub**









instructables.com

Instructables - How to make anything

instructables

let's make

Explore Contests Classes Publish

You

Featured: 3D Printing Class Arduino For Teachers

AUTODESK. Make anything.

Let's Make robots

EVERYTHING  
TECHNOLOGY  
WORKSHOP  
CRAFT  
HOME  
FOOD  
PLAY



**instructables**

lets make



Explore

Publish

Classes

Log In Sign Up

shape what you make

Featured:

Intel IoT

Arduino

Yarn

Lighting

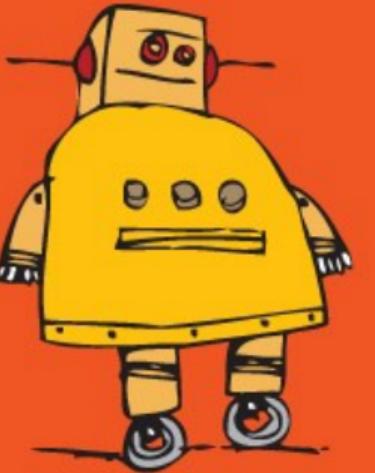
# INSTRUCTABLES CLASSES

Online classes that get you making - no experience required!

\$29.95 for a single class, or upgrade to Premium for unlimited access to all classes starting at \$2.95/month.

[Upgrade to Premium >>](#)





# instructables

Explore. Share. Make.

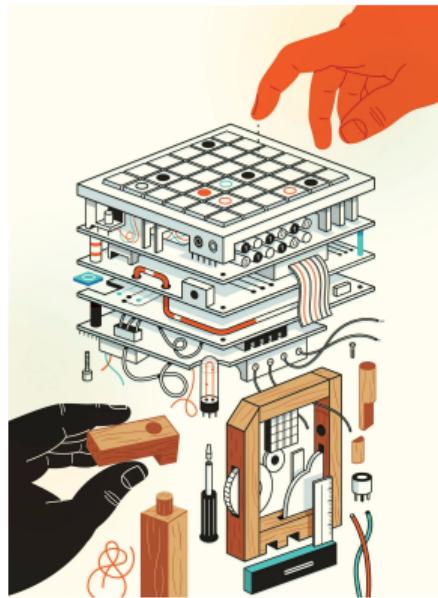




**Makers**



# Makers



## Movimiento *Maker*

El movimiento *Maker* se refiere a una comunidad global de personas que comparten un enfoque DIY y una filosofía de aprendizaje a través de la creación. Los *makers* utilizan tecnologías como la electrónica y la programación para construir proyectos creativos y resolver problemas cotidianos.

- La cultura *Maker* se basa en la colaboración y el intercambio de ideas, lo que la convierte en una comunidad vibrante y en constante evolución.

# Características del movimiento Makers

- Enfoque **DIY** (*Do it yourself*)
- Filosofía de aprendizaje a través de la creación
- Uso de tecnologías como la electrónica y la programación
- Construcción de proyectos creativos y resolución de problemas cotidianos
- Comunidad vibrante y en constante evolución
- Colaboración y intercambio de ideas
- Fuente importante de innovación y desarrollo tecnológico
- Fomento del pensamiento crítico y solución de problemas
- Acceso a herramientas y recursos
- Experimentación y creación como parte integral del proceso.



# THE MAKER MOVEMENT

## A TIMELINE OF KEY MODERN EVENTS

**JAN 2005**

First issue of **MAKE** magazine, a publication devoted to DIY projects, hits newsstands.

**JUN 2005**

Etsy launches, providing an online platform for artisans to sell their crafts.

**Etsy****APR 2006**

The Bay Area hosts the first Maker Faire, a public annual event celebrating the arts, crafts, engineering, science projects and the DIY community.

**OCT 2006**

The first TechShop opens in Menlo Park, California.

**FEB 2009**

Square, Inc., launches, providing crafters and artisans a vehicle to collect credit card payments directly from customers.

**APR 2009**

MakerBot, one of the first desktop 3D printers, is available for sale.

**APR 2009**

Kickstarter is formed and entrepreneurs begin to raise capital via crowdfunding.

**MAR 2011**

South-by-Southwest (SXSW) celebrates the 25th year of convening musicians, artists, and tech junkies in Austin, Texas.

**SEP 2012**

The New York Times reports that the "Made in America" label is regaining popularity, with more consumers buying domestic, local goods they feel represent "old-school craftsmanship, even luxury."

**MADE IN USA****JAN 2013**

*The Maker Manifesto: Rules for Innovation in the New World of Crafters, Hackers, and Tinkerers*, written by TechShop CEO Mark Hatch, is published.

**MAY 2013**

President Obama announces the National Network for Manufacturing Innovation (NNMI), a collaboration between manufacturers and academia to problem-solve and help advance the manufacturing industry.

**MAY 2014**

The White House launches Mayors Maker Challenge to encourage support of Makers at the local level, and hosts the first-ever White House Maker Faire.

**FEB 2015**

Raspberry Pi, a computer the size of a credit card and as inexpensive as \$5, reached five million units in sales.

**NATIONAL LEAGUE of CITIES**

JOIN THE Revolution!

# The Maker Movement

## ~~MANIFESTO~~

Mark Hatch

# Radicalize!

Tools of Industrial Revolution

Agriculture  
Banking  
SAVE LIVES!  
MAKERS ARE CHANGING...

If we could bring more Folks...

TO MAKER SPACES

we can CHANGE THE WORLD!

Make is Fundamental.

Share

Give away small pieces of yourself.

Learn

Tool up

embrace Change

Make something For Christmas!

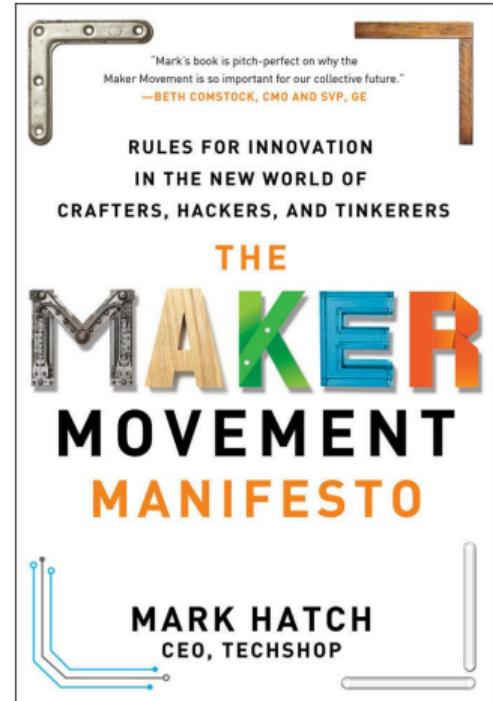
Participate

Support

# *Maker Movement Manifesto*

El **Manifiesto Maker** es un conjunto de valores y creencias que define la *cultura y filosofía* del movimiento *Maker*. Se enfoca en el aprendizaje a través de la creación y la experimentación, y fomenta el uso de tecnologías y herramientas accesibles. Además, valora la colaboración y el intercambio de ideas, y promueve la innovación y la resolución de problemas a través de la creación.

- El **Manifiesto Maker** también fomenta el pensamiento crítico y la toma de riesgos en el proceso de creación
- Es una guía para aquellos que desean participar en el movimiento *Maker* y en la cultura DIY.



# WE ARE THE *Makers.*

THE MAKERS OF OUR FUTURE, THE PROPHETS OF OUR PROSPERITY, THE SOURCE OF OUR OWN SUSTAINABILITY.  
WE KNOW THE TRANSFORMATION WE MAKE IN OURSELVES,  
**SO WE MAKE IN THE WORLD.**  
WE REJECT ANTIQUATED SYSTEMS AND IDEOLOGIES

THAT KEEP US SMALL & SCARED,

THAT BLEED OUR RESOURCES AND STIFLE THE INHERENT PROMISE IN US ALL.  
WE BELIEVE THERE IS NO COMPETITION; THERE IS ONLY COMMUNITY, SPARKING INSPIRATION AND COLLABORATION.

*WE speak OUR TRUTH.  
WE seek THE TRUTH.  
AND WE lead, WITH LOVE.*

WE GIVE OF OURSELVES AUTHENTICALLY & COURAGEOUSLY,  
KNOWING THAT OUR CONTRIBUTION – SPANNING A MOMENT OR A MILLENNIUM –  
**IS VITAL & WORTHY.**  
WE CREATE BY FOLLOWING OUR DEEPEST DESIRES,  
OUR SECRET AMBITIONS, OUR PASSIONS, AND OUR VISIONS OF WHO WE KNOW OURSELVES TO BE.  
WITH EACH INSPIRED CREATION, WE MOVE CLOSER TO THE WORLD WE EACH INDIVIDUALLY WANT TO LIVE IN, AND WE KNOW, THROUGH CREATIVITY AND LOVE, IS WITHIN REACH.  
WE IMAGINE THE IMPOSSIBLE. AND WE MAKE IT POSSIBLE.  
**WE DREAM. WE INNOVATE. WE INSPIRE.**  
WE PLAY. WE SHARE. WE ARE VULNERABLE. WE LISTEN. WE EMBRACE. WE RESPOND.  
WE INFLUENCE ECONOMIES. WE BUILD. WE COMPOSE. WE WRITE.  
WE RAISE FAMILIES. WE ENGINEER. WE PAINT. WE DANCE. WE PROPEL MOVEMENTS.

WE CREATE.  
*We are Makers.*  
YOU ARE WELCOME.

# Introducción al proceso de prototipado electrónico



# Prototipado

## ¿Qué es el prototipado?

El **prototipado** con *sistemas embebidos* es un proceso en el que se utiliza *hardware* y *software* especializado para crear un **modelo funcional** de un producto o sistema. El prototipado con sistemas embebidos permite a los desarrolladores probar y mejorar su diseño antes de la **producción en masa**, lo que resulta en un producto más refinado y de alta calidad.

- Puede implicar muchas fases
  - Desde los bocetos (*sketches*) iniciales
  - A través de planos detallados y una serie de refinamientos
  - Hasta la construcción de un modelo totalmente funcional que pueda reproducirse.
- Puede ser una actividad única y rápida para poner a prueba una idea (**prueba de concepto**).

# Prototipado

## Modelo funcional

Un **modelo funcional** de un producto o sistema es una representación física o virtual que imita el comportamiento y las características esperadas del producto o sistema final.

- El **modelo funcional** es una versión *inicial* o *parcial* del producto o sistema diseñado para ser *probado* y *evaluado* antes de la **producción en masa**
- Se puede asegurar que el producto o sistema final sea de alta calidad y cumpla con las especificaciones deseadas.

## Cost-Efficiency Analysis

Concept Validation

Development Speed Boost

## BENEFITS OF PROTOTYPING

Fresh Perspective

Cost Reduction

Testing Components

Raising Investments

## Características del prototipado

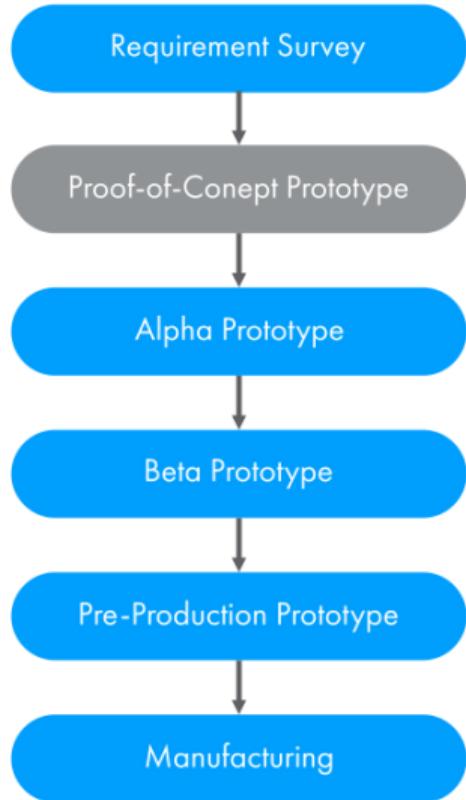
- La fase de prototipado es importante en el desarrollo de productos electrónicos.
- **Ofrece varios beneficios valiosos:** validación de conceptos, análisis de eficiencia de costos, aceleración del desarrollo, perspectiva fresca y reducción de costos.
- La fase de prototipado ayuda a resolver problemas clave al principio y a encontrar alternativas más baratas.
- Las empresas que no están especializadas en el diseño de *hardware* electrónico suelen contratar a empresas especializadas en prototipado electrónico.
- Durante la fase de prototipado, los ingenieros pueden descomponer diseños complejos en partes más simples para probarlos individualmente.

## Características del prototipado

- Los prototipos a menudo se utilizan para mostrar el dispositivo a inversores potenciales y obtener financiamiento para el desarrollo posterior del producto.
- La fase de prototipado conlleva ciertos riesgos, como la viabilidad económica del concepto o el fracaso en la fabricación a gran escala.
- El mercado global de fabricación y diseño de contrato electrónico se espera que alcance \$730.5 billones para 2027.
- El desarrollo de hardware, electrónica, firmware y software requiere especialistas altamente capacitados y experiencia vasta.

# **Proceso de creación de prototipos electrónicos y tipos de prototipos**

## ELECTRONICS PROTOTYPING PROCESS

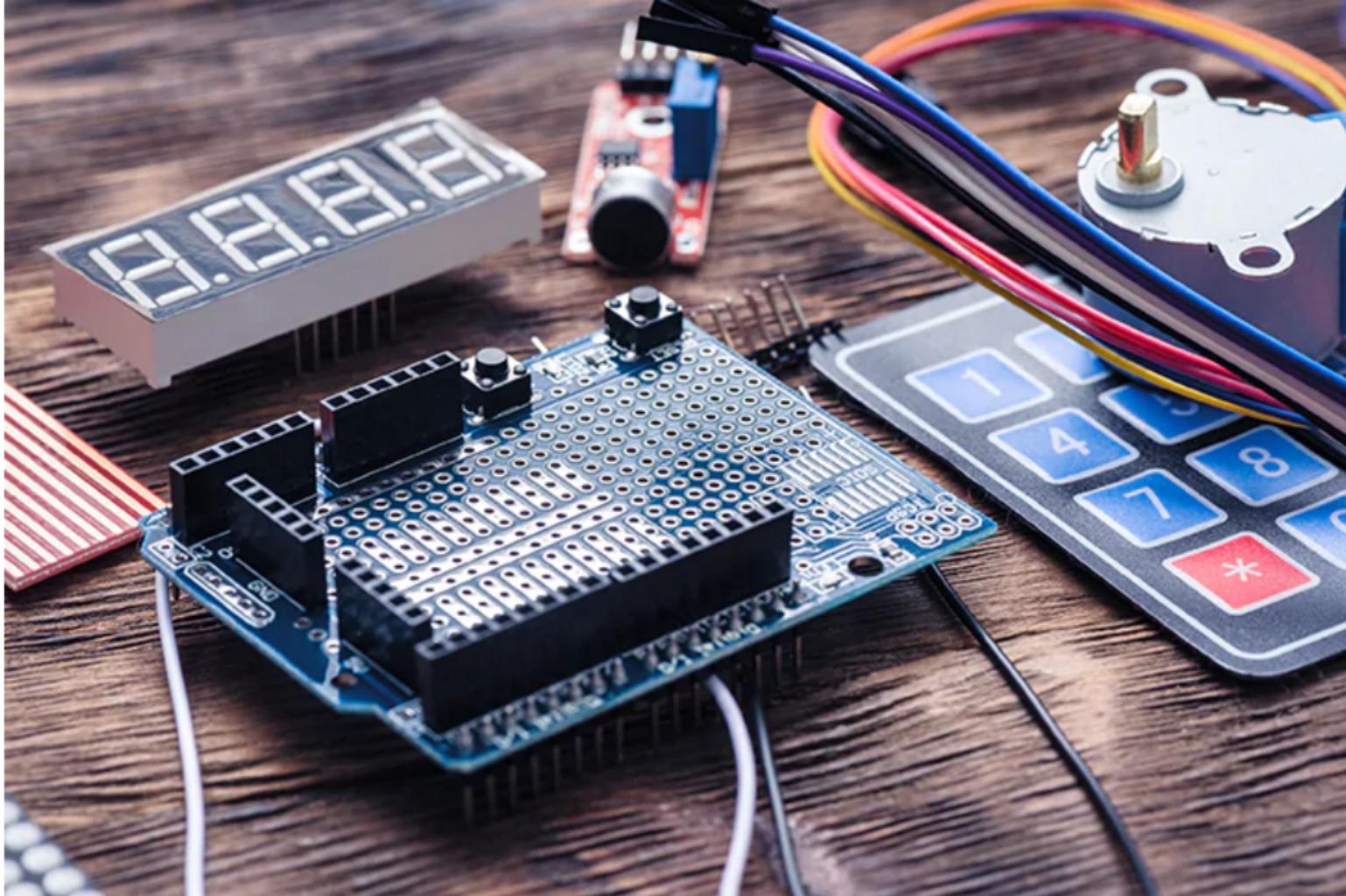


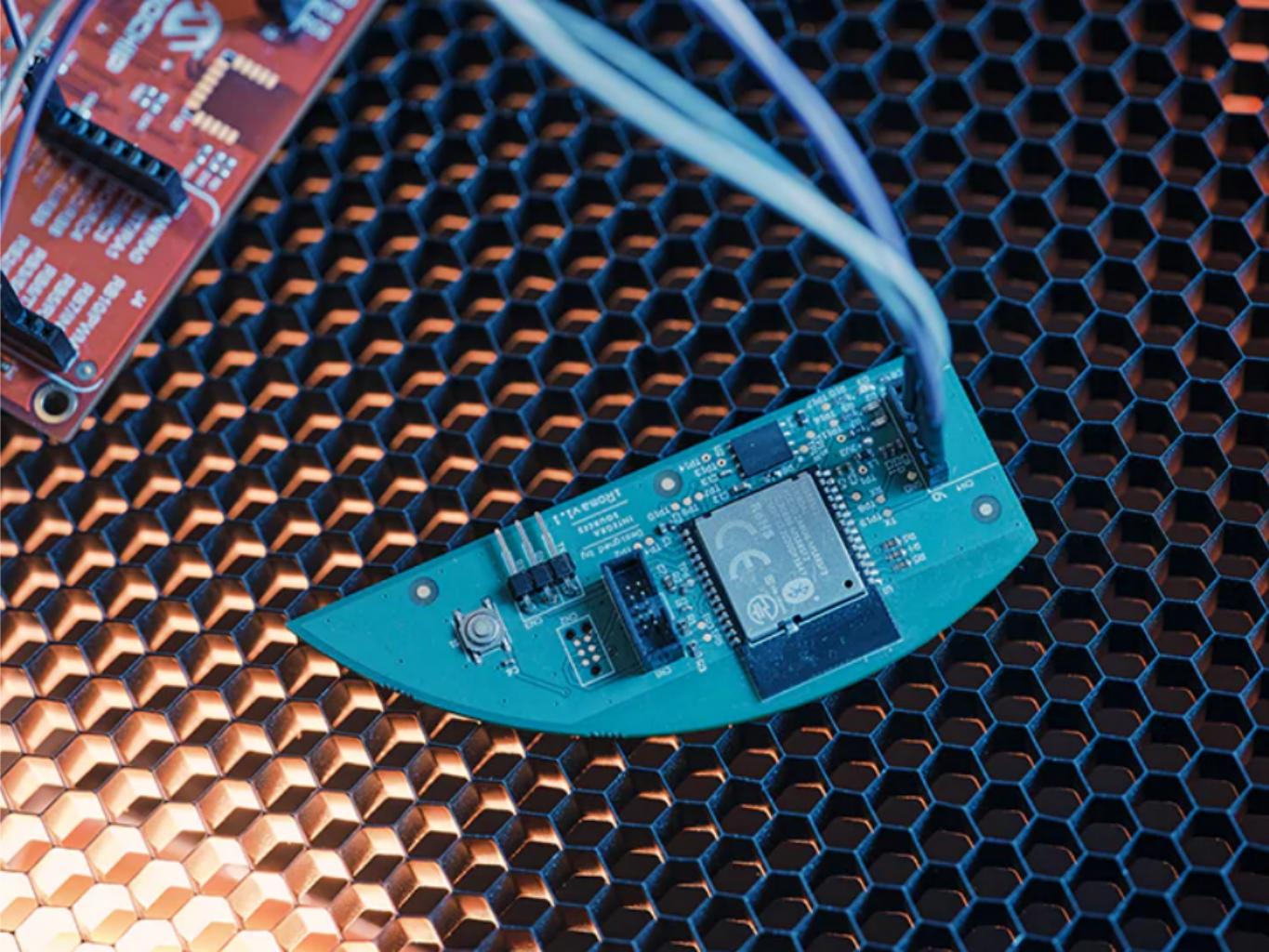
## Fase 1: Encuesta de requisitos

- Se contacta a los **clientes/usuarios finales** para discutir las características y funciones del dispositivo
- Importancia de tener información clara para un proceso de desarrollo rápido y sin problemas

## Fase 2: Prototipo de prueba de concepto (POC)

- Se crea un **POC** para demostrar que el concepto es viable
- Se diseñan los esquemas eléctricos de las placas de circuito impreso (PCB)
- **Métodos de prototipado:** prototipado virtual, usando kits de desarrollo y placas de prototipado electrónico, o diseñando PCB personalizadas





## Fase 3: Prototipo Alfa (MVP)

### Minimum viable product (MVP)

- Se crea un prototipo que realiza su función principal
- Puede faltar funciones secundarias, pero se agregarán más tarde
- Es suficiente para comenzar a trabajar en su *firmware*

## Fase 4: Prototipo Beta

- Se espera tener un prototipo totalmente funcional, optimizado para la fabricación en masa
- En la práctica, es difícil eliminar todos los problemas
- El Beta es un prototipo que está casi listo para su producción en masa.

## Fase 4: Prototipo Beta

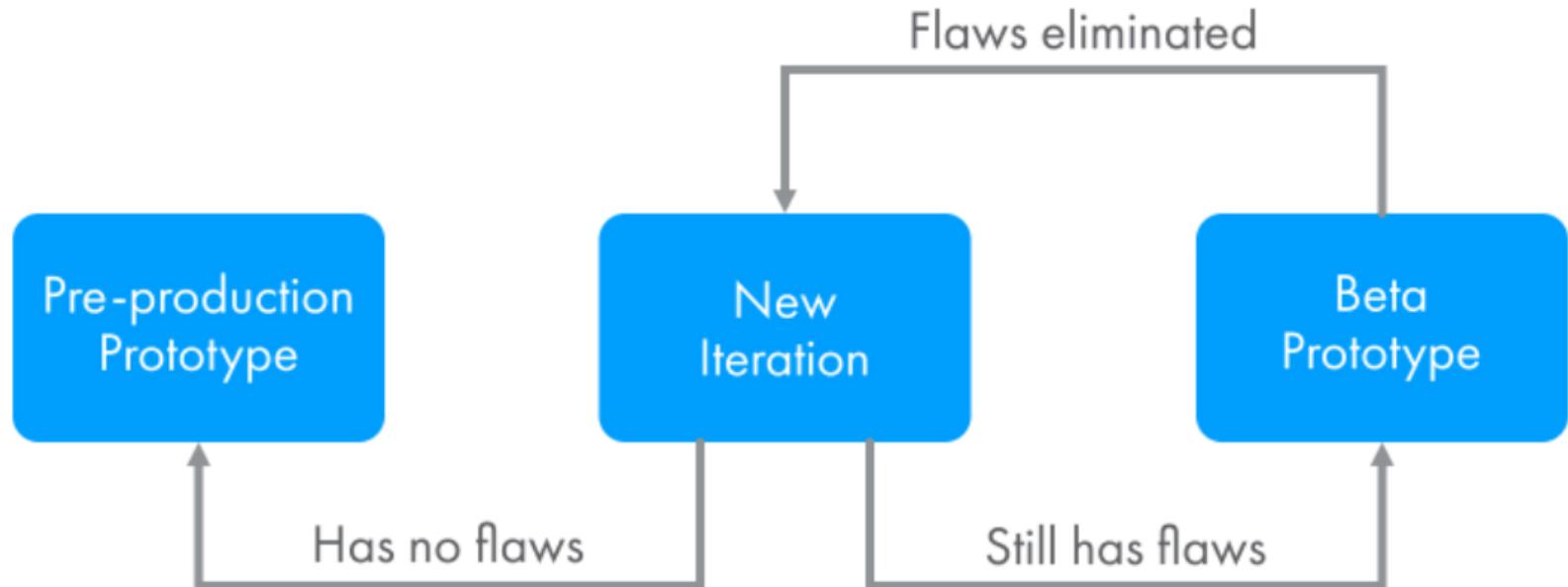


Figure 1: Este esquema muestra cómo un prototipo beta puede revisarse varias veces antes de convertirse en un prototipo de preproducción.

# Problemas frecuentes

- Falta de información adecuada
- Falta de objetivo claro
- Caída en la tentación de agregar más funciones
- Enfocarse en lo que funciona bien en lugar de lo que no
- Parar demasiado pronto

# Conclusiones

- Prototipado es importante en el diseño de productos electrónicos
- Ayuda a estimar viabilidad económica en etapas tempranas
- Requiere experiencia en electrónica y programación
- Desarrollo de un producto desde cero puede tomar meses a años
- Contratar un equipo de desarrollo requiere un acuerdo de confidencialidad.

# Introducción a Arduino

# Introducción a Arduino

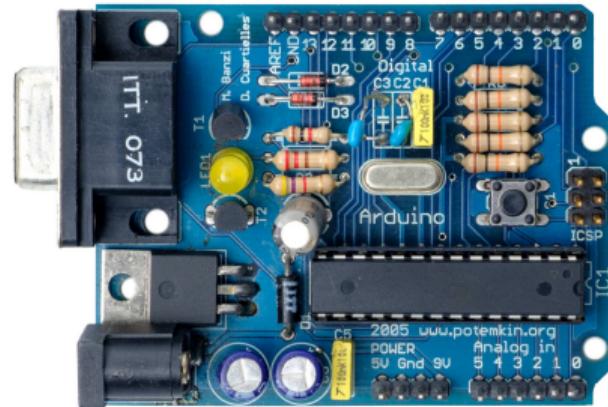
## Arduino

Plataforma de *hardware* y *software* libre que permite desarrollar proyectos interactivos con sensores, **actuadores** y otros dispositivos electrónicos. Consta de una placa de **microcontrolador** y un entorno de programación integrado basado en **Processing/Wiring**, que facilita la escritura de código en lenguaje de programación C/C++.



# Introducción a Arduino

- El Arduino es una plataforma de creación de prototipos
- Arduino es un ordenador sencillo, asequible y a pequeña escala que se centra en la interacción con el mundo exterior
- Un Arduino permite tomar información del mundo exterior con sensores
- El Arduino fue desarrollado originalmente por profesores para que sus estudiantes de diseño que no eran ingenieros pudieran crear objetos y entornos interactivos.
- El Arduino original fue lanzado en 2005 (se han vendido más de 1 millón)



# Introducción a Arduino



Figure 2: Massimo y David con Fabio Violante, Director General de Arduino

# Introducción a Arduino

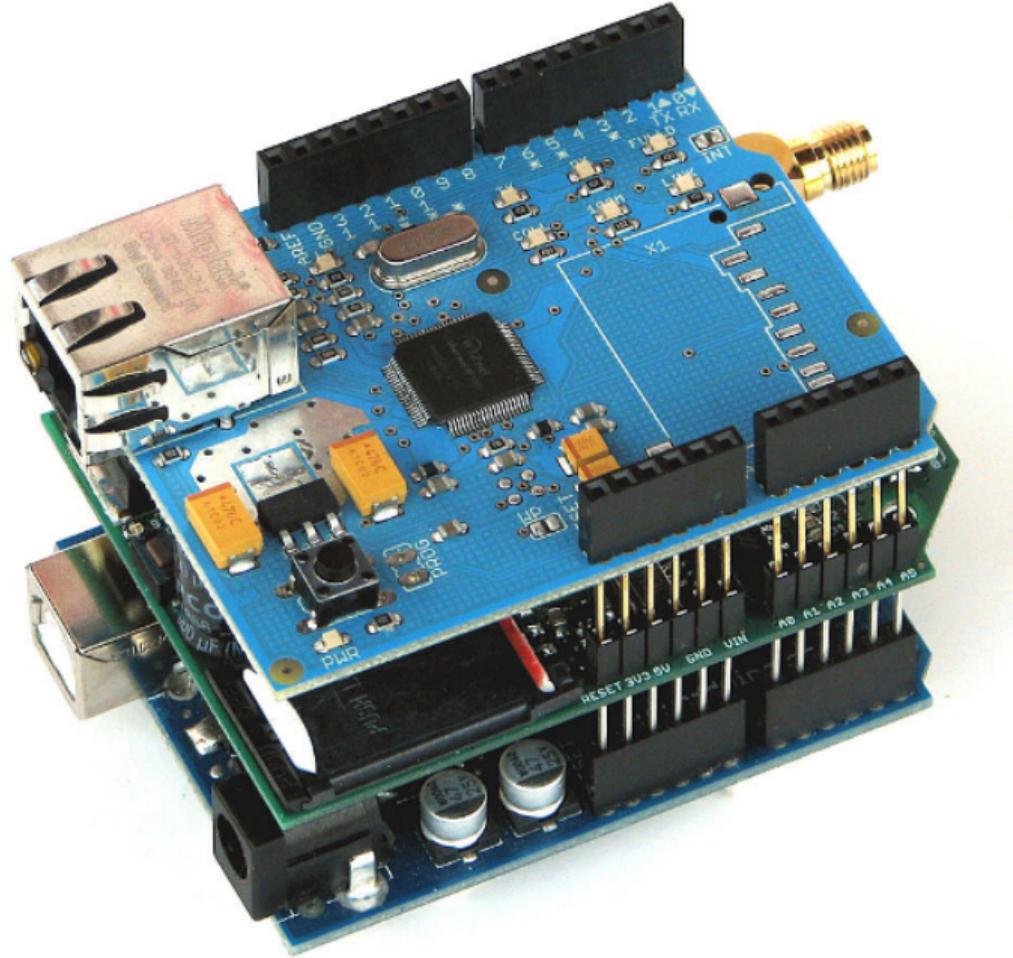
Arduino cuenta con una amplia comunidad de usuarios que ha desarrollado una amplia gama de **librerías** y **shields** (módulos de expansión) para una amplia variedad de aplicaciones, lo que lo hace ideal para el aprendizaje de robótica, Internet de las Cosas y el diseño de sistemas embebidos.

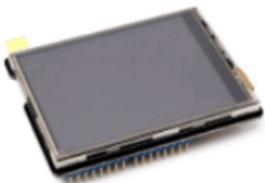
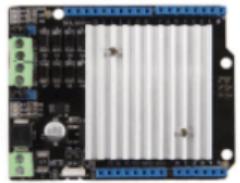
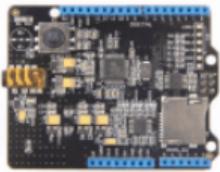
# Shield (tarjeta de expansión)

## Shield

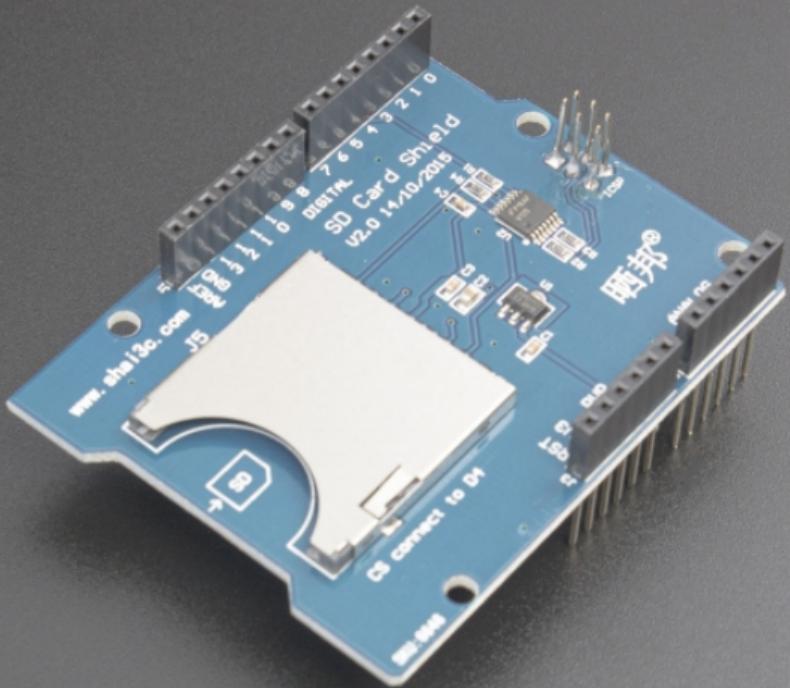
Un *shield* de Arduino es una tarjeta de expansión que se coloca encima de una placa de Arduino para agregar nuevas funciones y capacidades a la placa base.

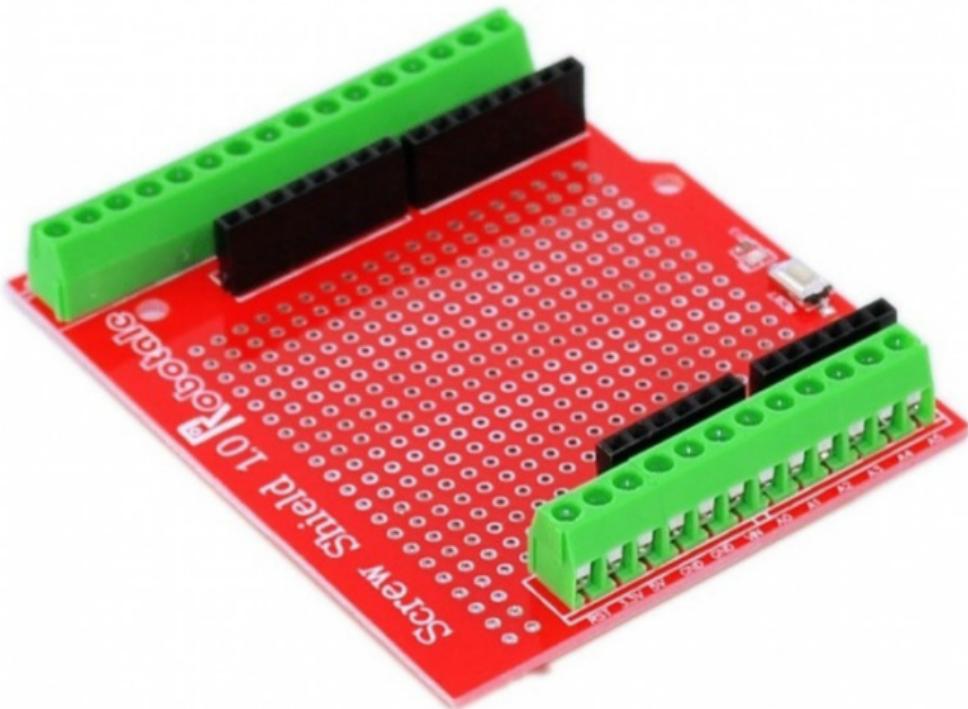
- Los *shields* son diseñados para ser *plug-and-play*, lo que significa que se conectan fácilmente a la placa base y se pueden usar inmediatamente sin necesidad de soldar ni realizar otros trabajos de ingeniería.
- Hay una amplia variedad de *shields* disponibles para las placas de Arduino, que van desde módulos de comunicación inalámbricos hasta sensores y actuadores.
- Permite a los desarrolladores seleccionar los módulos que necesitan para su proyecto y ampliar fácilmente sus capacidades sin tener que diseñar y construir sus propios circuitos.

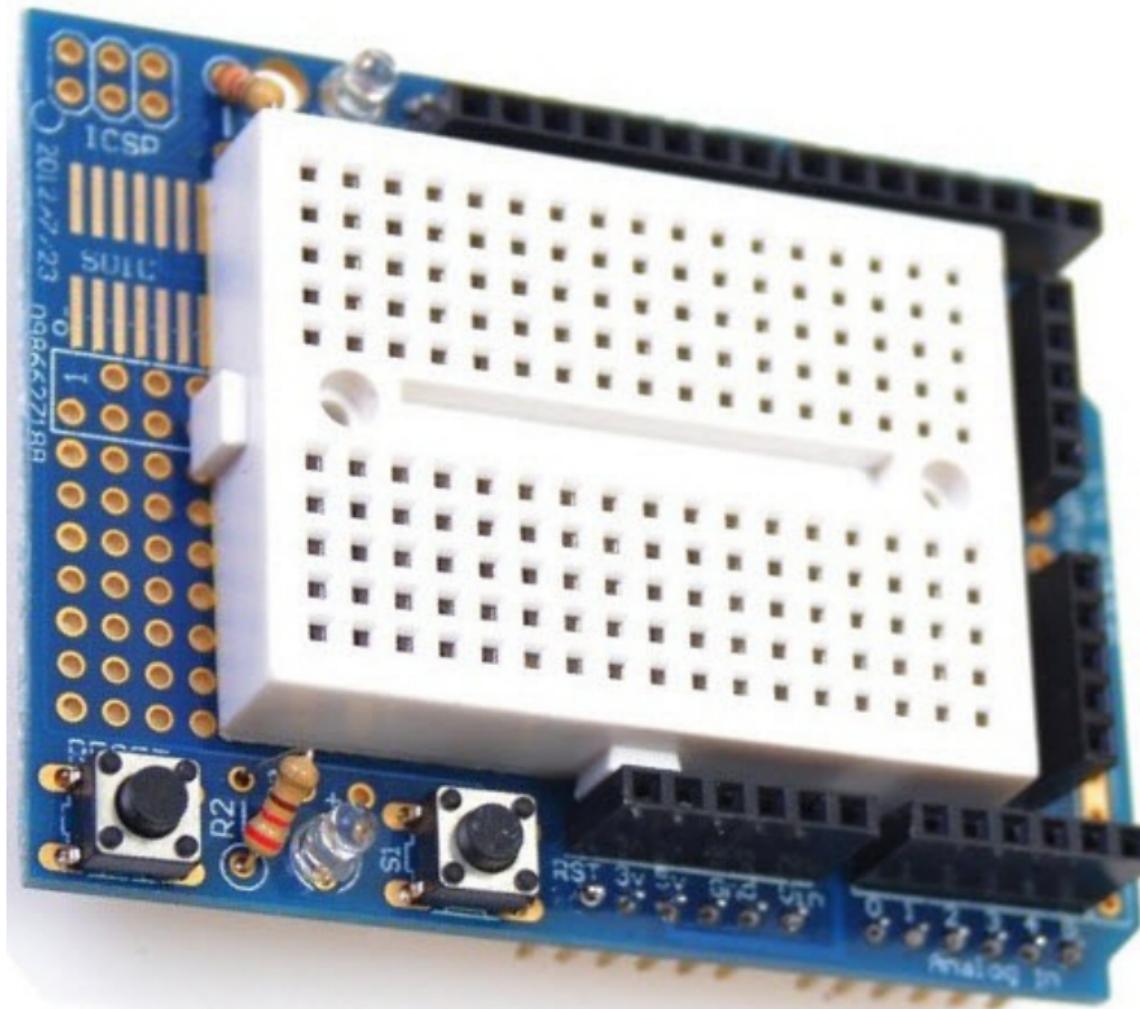


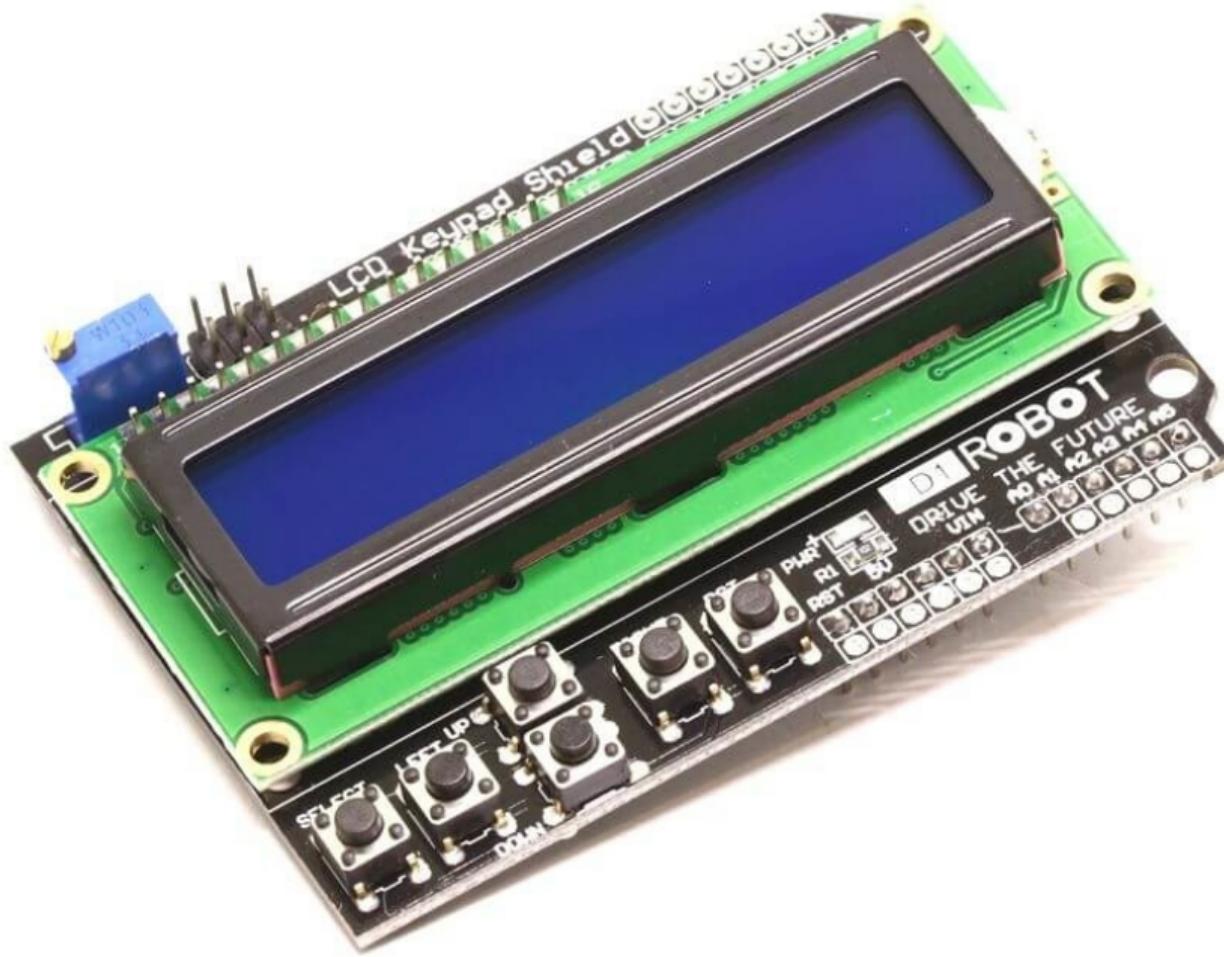


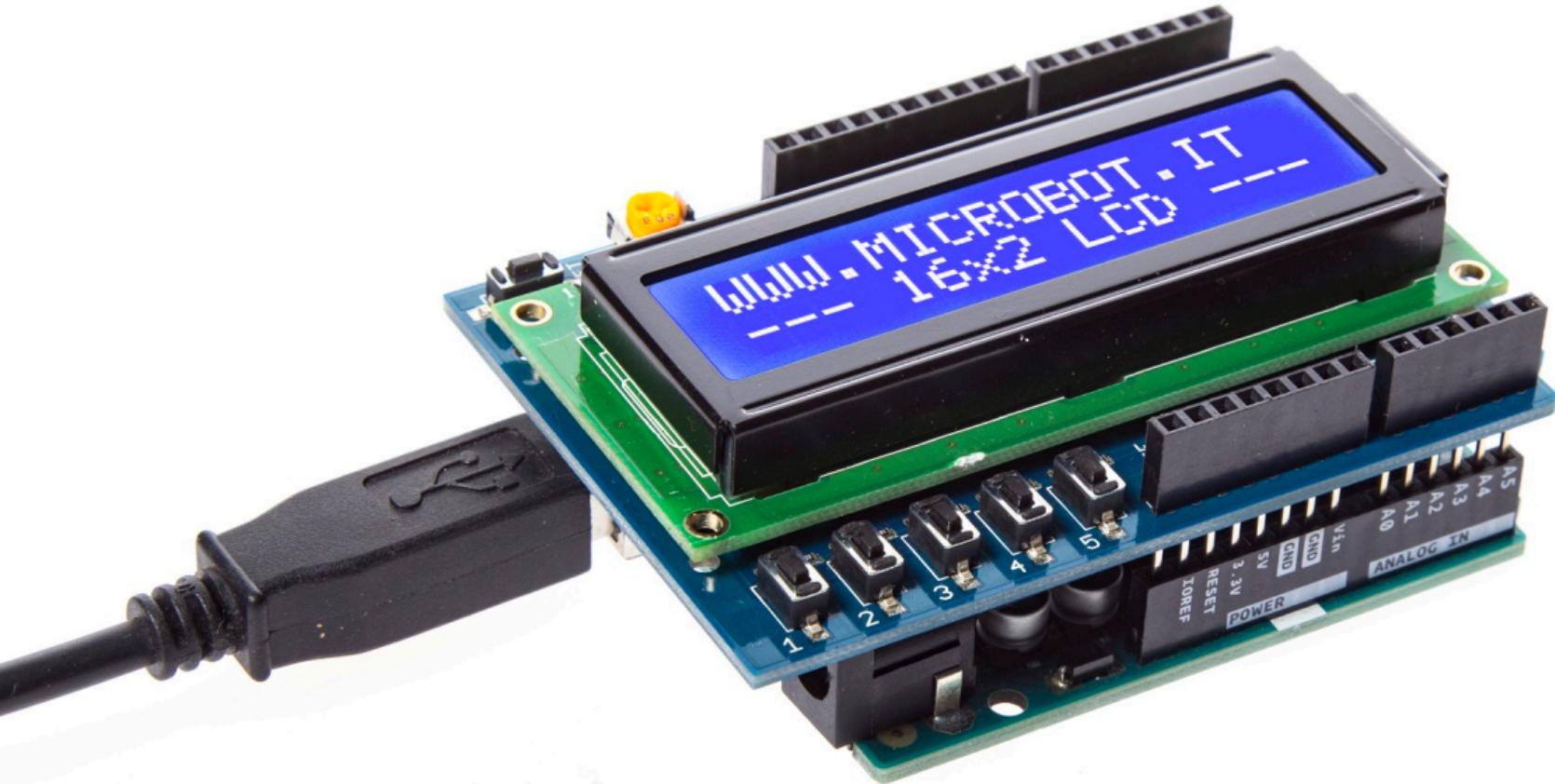




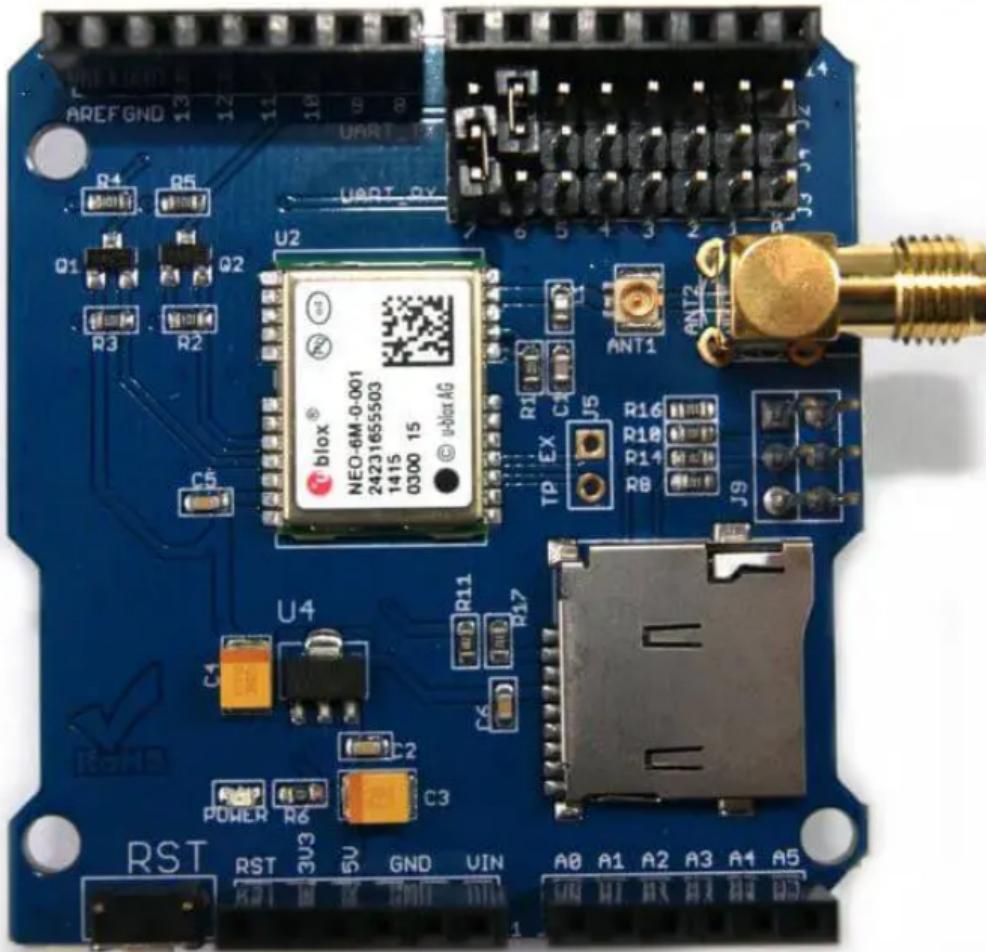


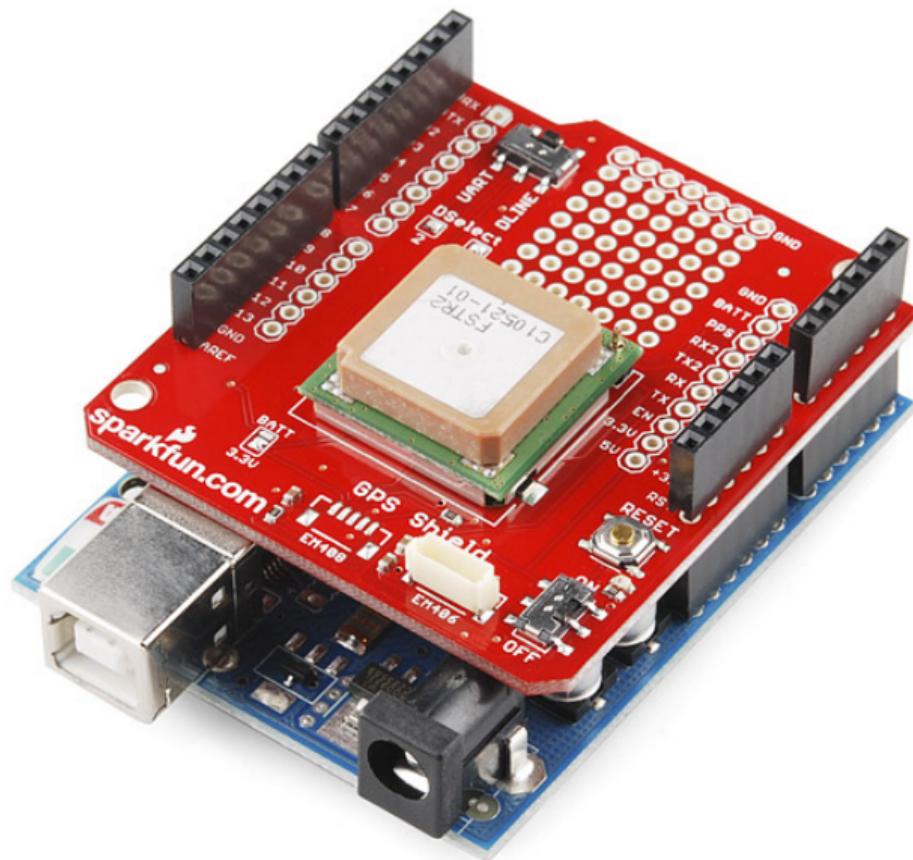






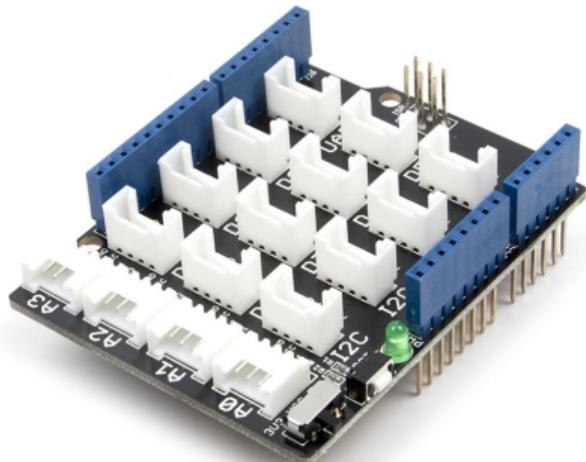






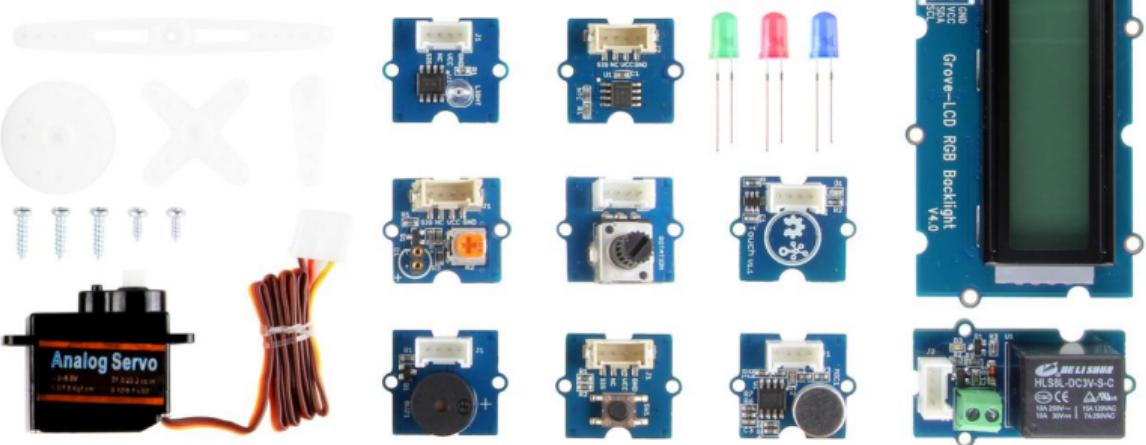




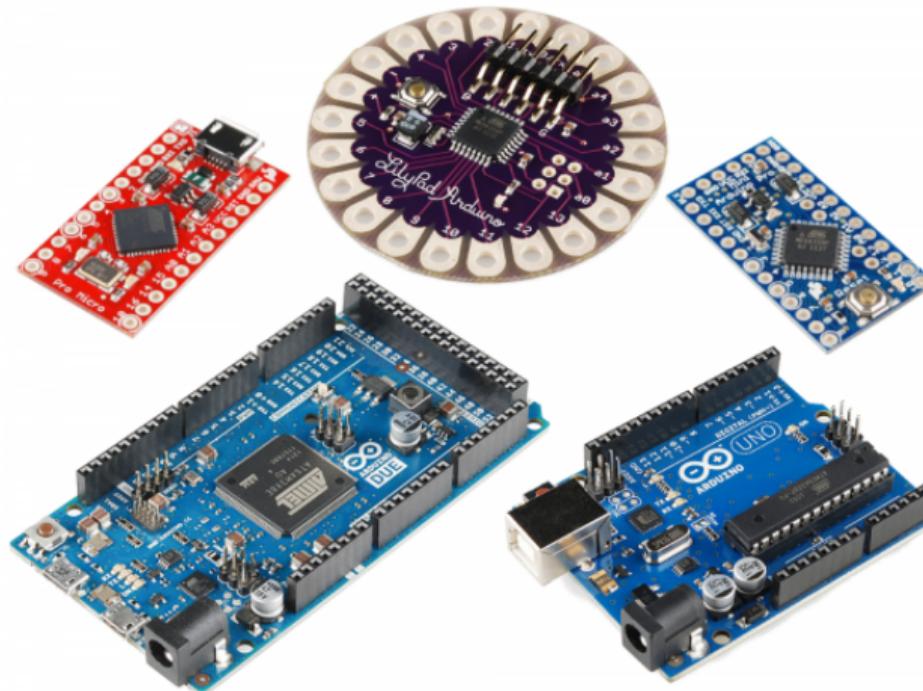




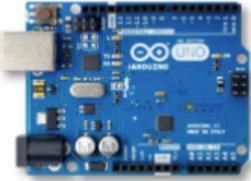
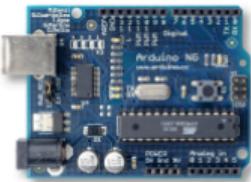
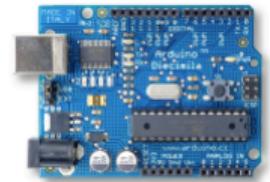
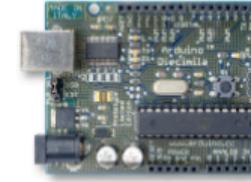
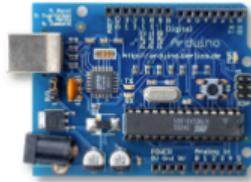
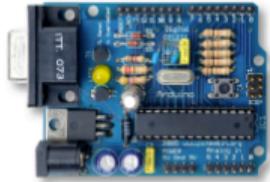
# Grove-Starter Kit



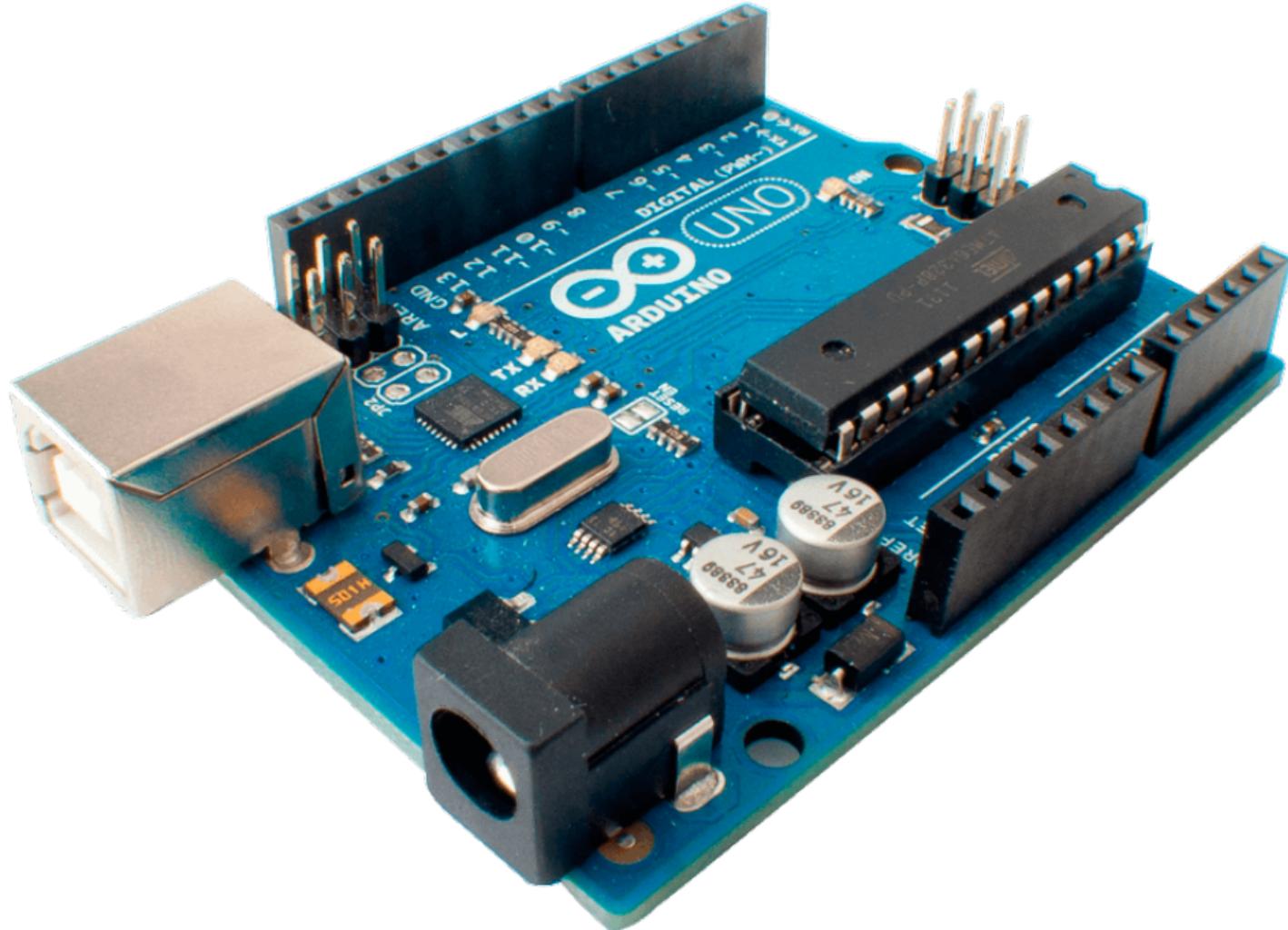
# Introducción a Arduino



- Hay muchas versiones del Arduino, y cada una está diseñada para una función específica.



**Arduino UNO**



# Características del Arduino UNO

- Utiliza un microcontrolador **ATmega328P** (chip de 8 bits )
- **Entrada/Salida:** 14 Digitales y 6 analógicas (6 pines PWM)
- **Alimentación:** 5V a través del puerto USB o Vin de 7-12V
- **Frecuencia de reloj:** 16 MHz
- **Memoria:** 32KB de memoria Flash, 2 KB usado por Bootloader, SRAM: 2 KB, EEPROM: 1 KB
- Interfaz USB
- Fue desarrollado en 2005 por un grupo de estudiantes y profesores de la Universidad de Ivrea, Italia.

MADE  
IN ITALY

AREF  
GND

13

GND

12

~11

~10

~9

8

7

~6

~5

4

~3

2

TX → 1

RX ← 0

DIGITAL (PWM~)



UNO

ON

ARDUINO

RESET-EN

EOL

RESET

WWW.ARDUINO.CC

RESET

3.3V

POWER

GND

Vin

A0

A1

A2

A3

A4

A5

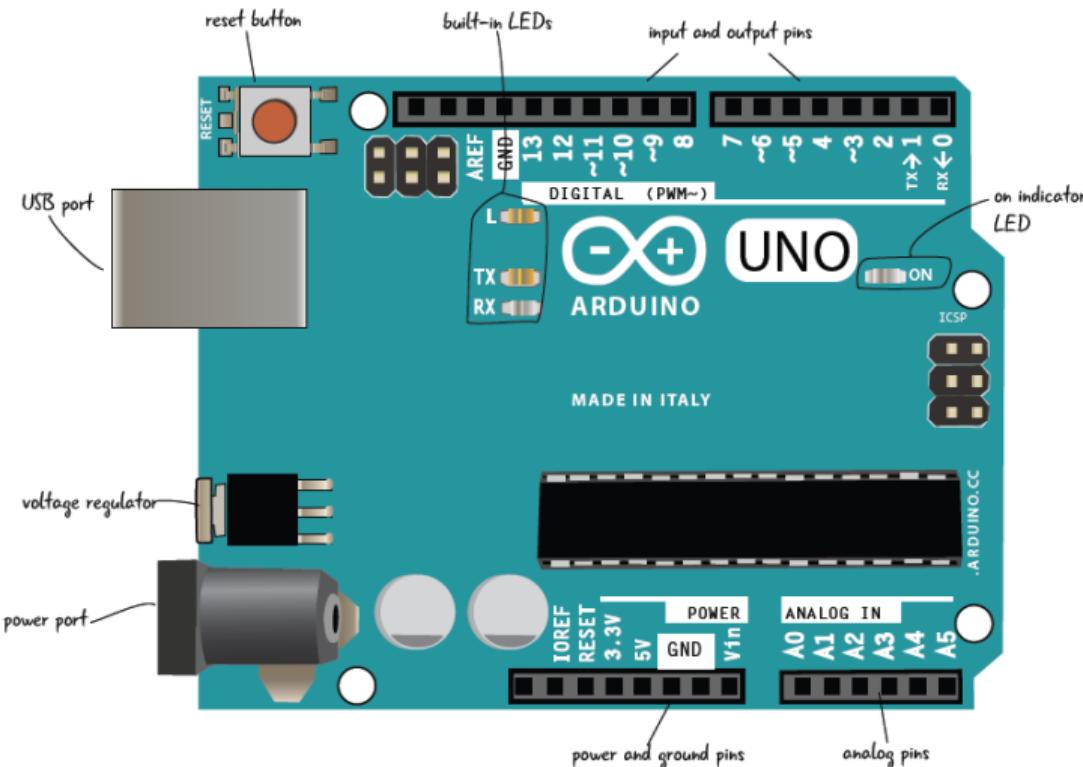
100S

AD  
GND  
A6  
A7  
A8  
A9  
A10  
A11  
A12  
A13

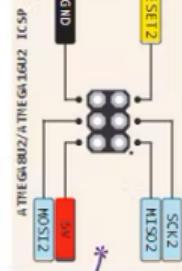
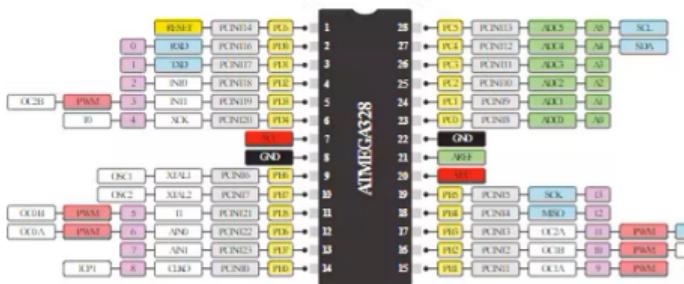
h0  
47  
25V

h0  
47  
25V

# Partes del Arduino

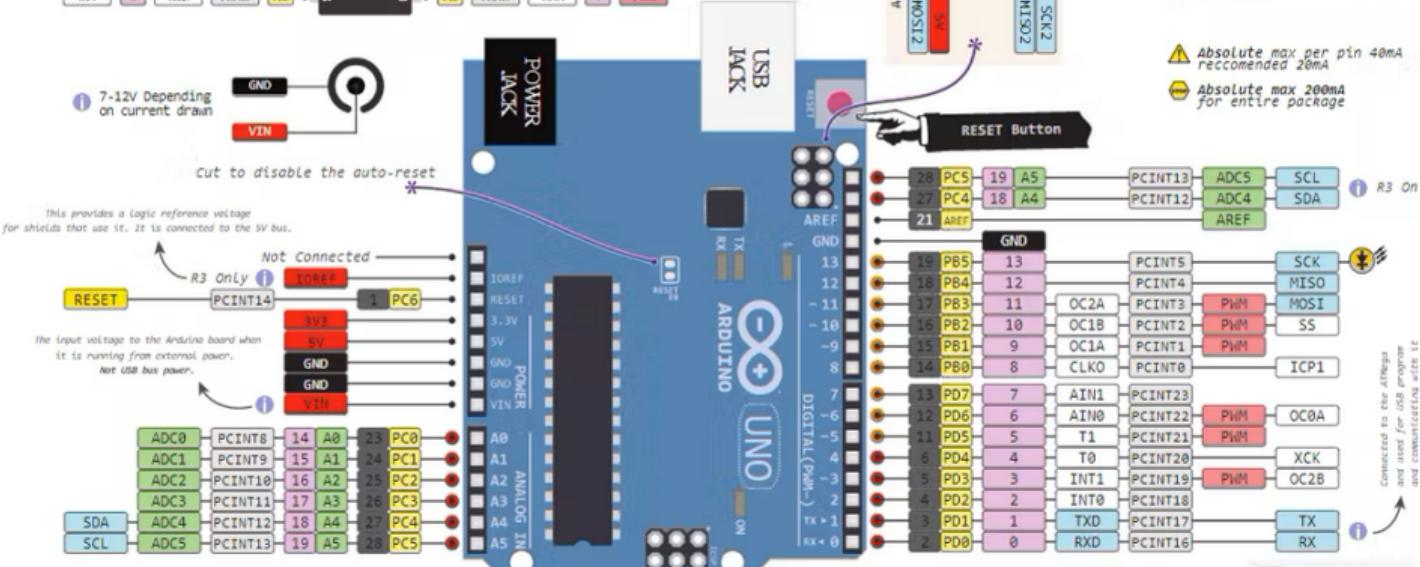


THE  
DEFINITIVE  
**ARDUINO**  
**UNO**  
PINOUT DIAGRAM



**⚠ Absolute max per pin 40mA recommended 20mA**

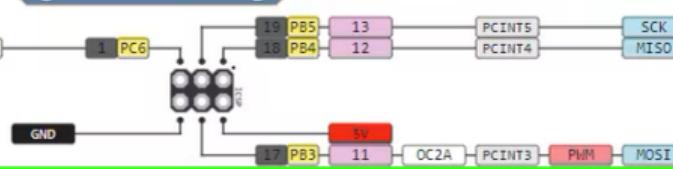
**⚠ Absolute max 200mA for entire package**



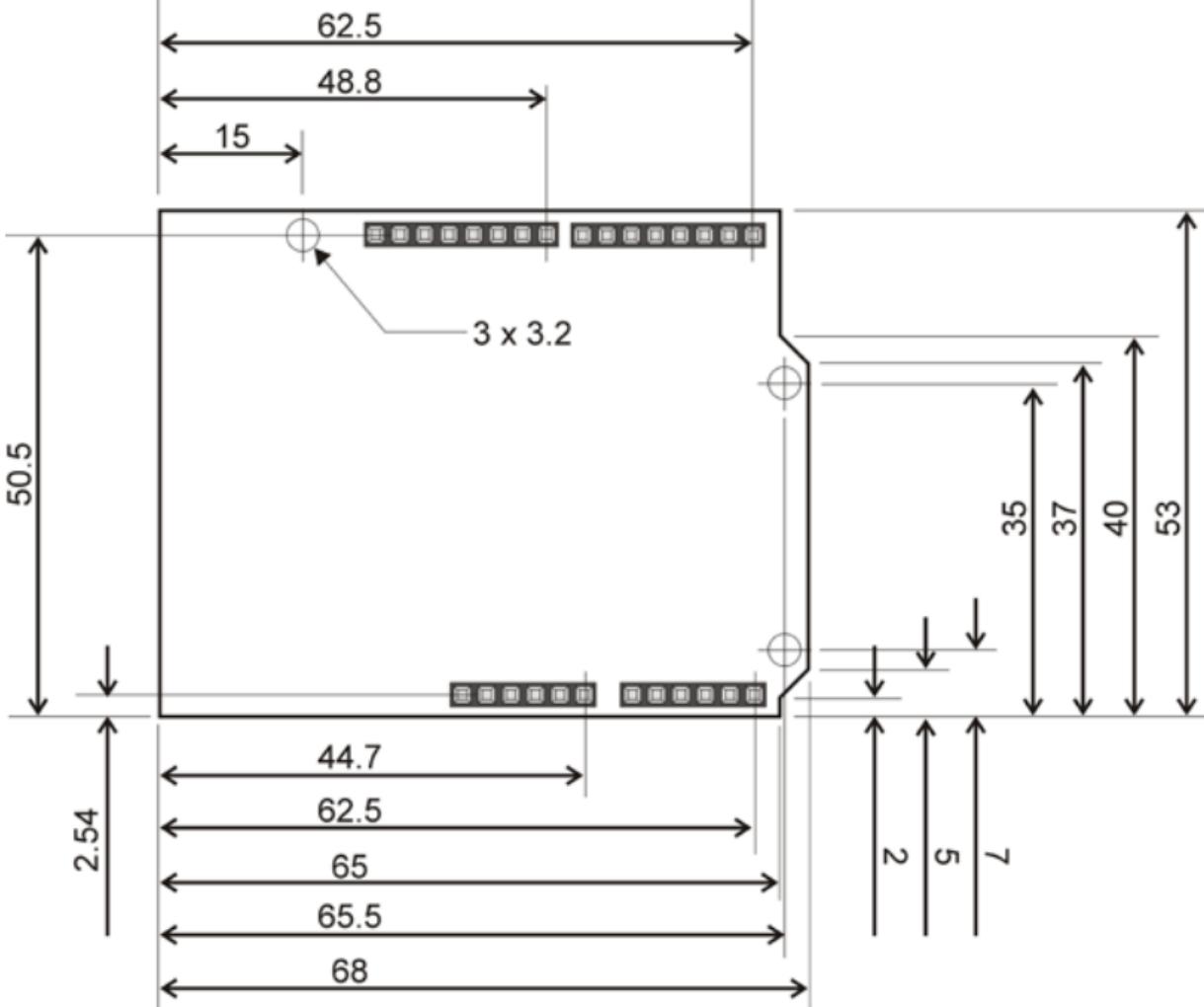
www.parrot.com

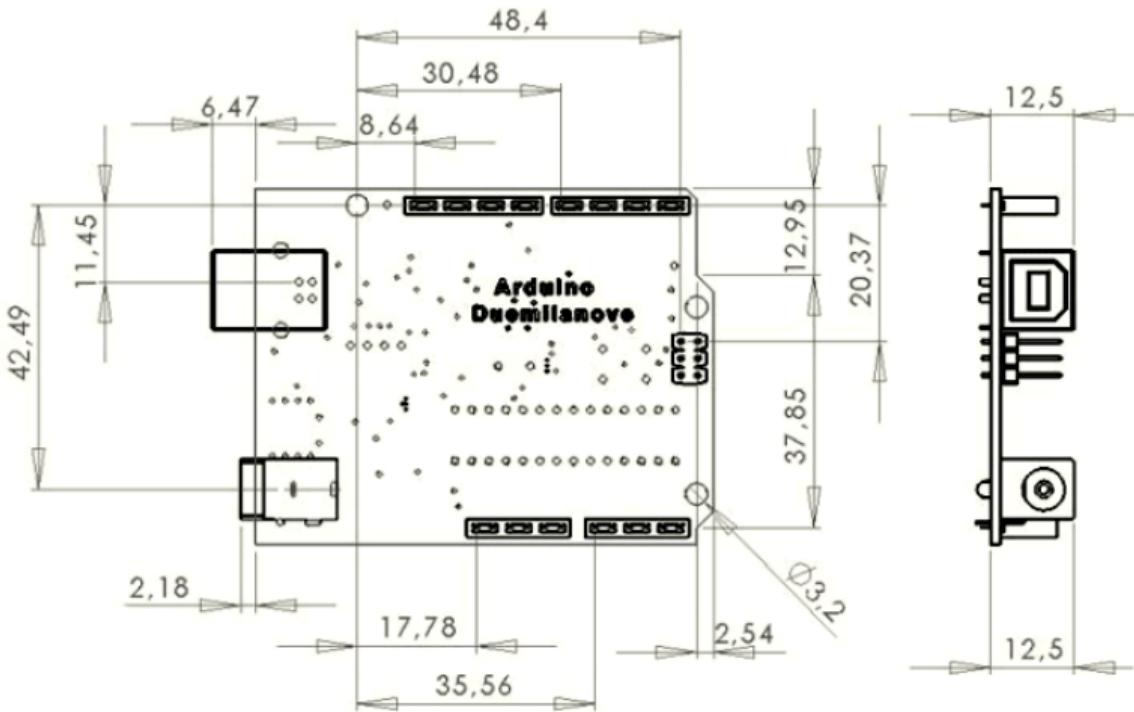
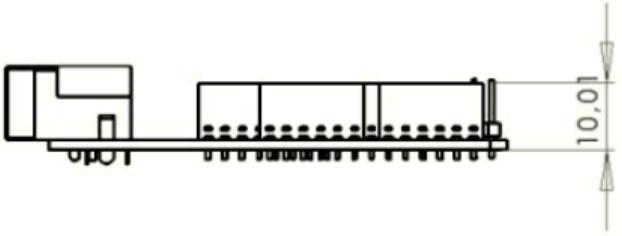
18 FEB 2013

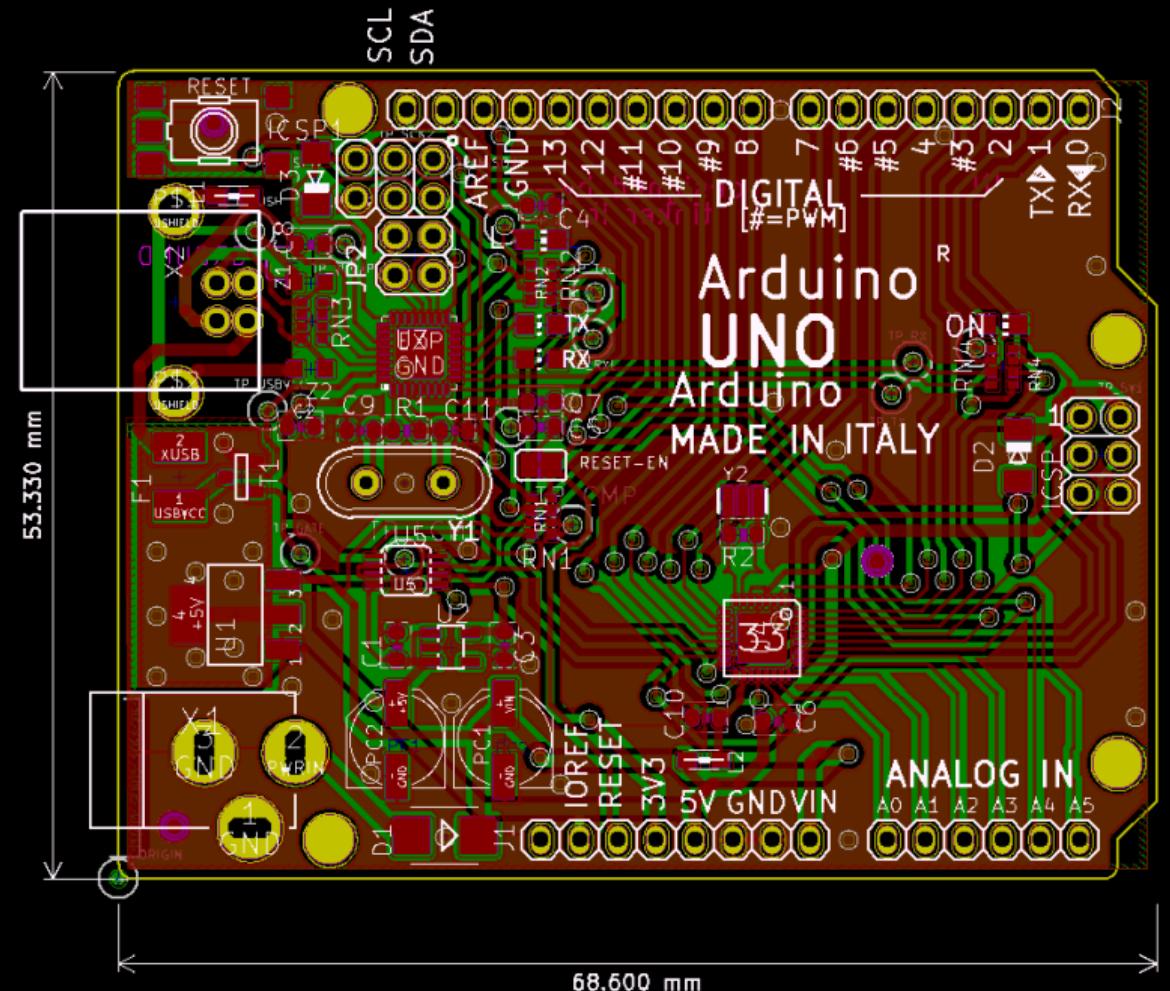
ver 2 rev 2 - 05/03/2013

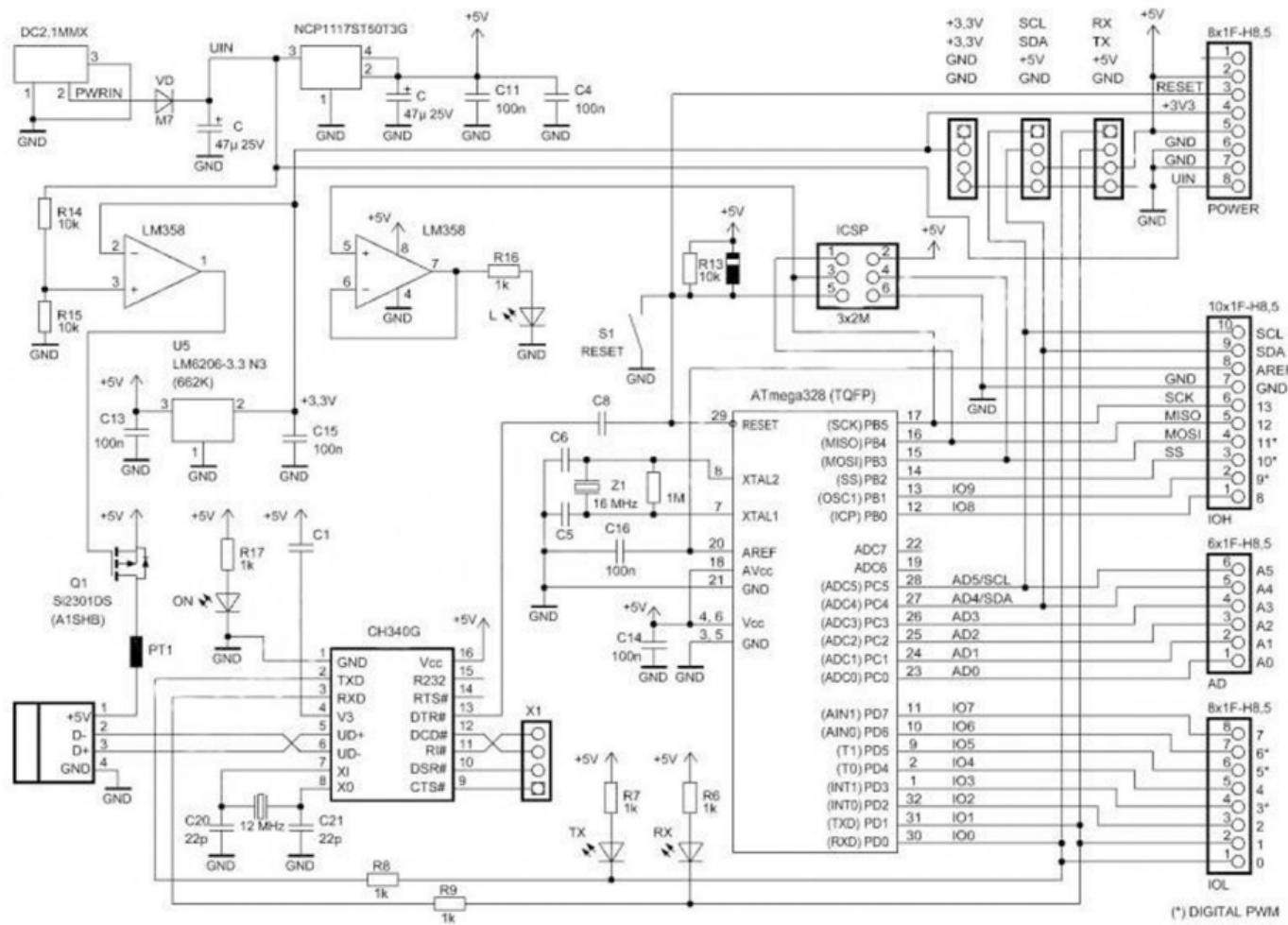


GND  
Power  
Control  
Port Pin  
Pin Function  
Digital Pin  
Analog Related Pin  
PWM Pin  
Serial Pin  
IDE  
Source Total 150mA









(\*) DIGITAL PWM

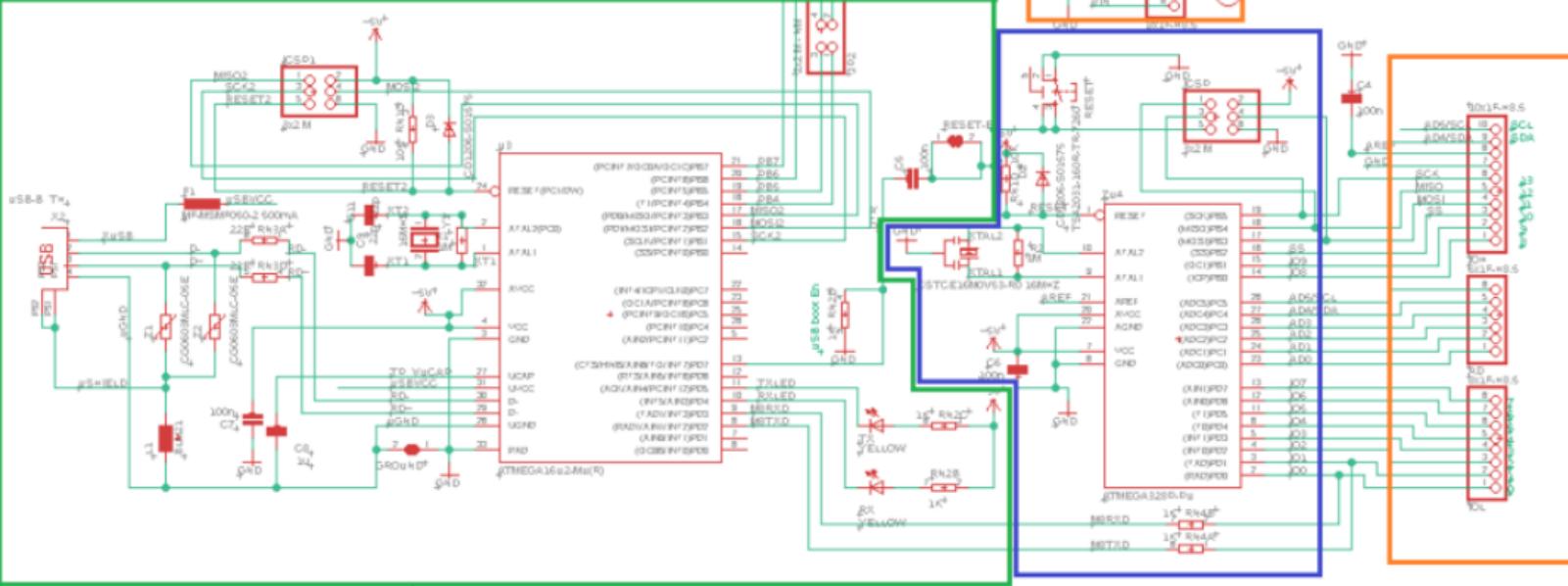
● Power Supply

● USB

● Microcontroller

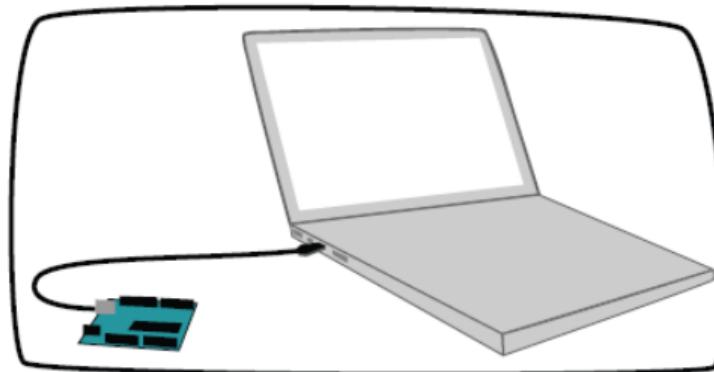
● I/O

## Arduino(TM) UNO Rev3

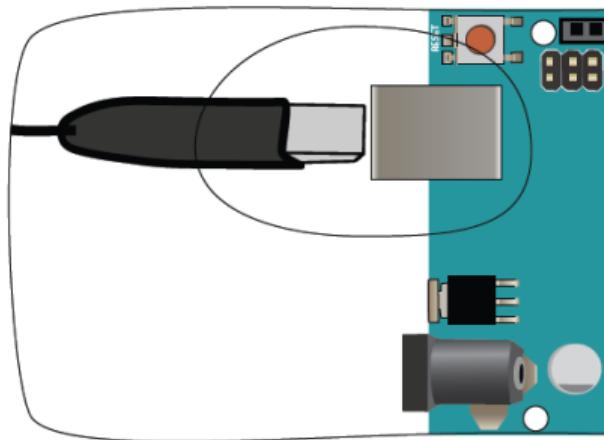


**Conecta tu Arduino al ordenador**

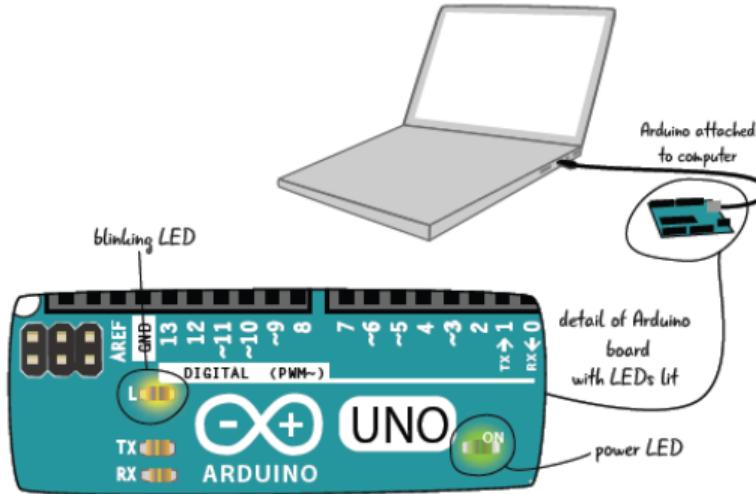
# Conecta tu Arduino al ordenador



# Conecta tu Arduino al ordenador

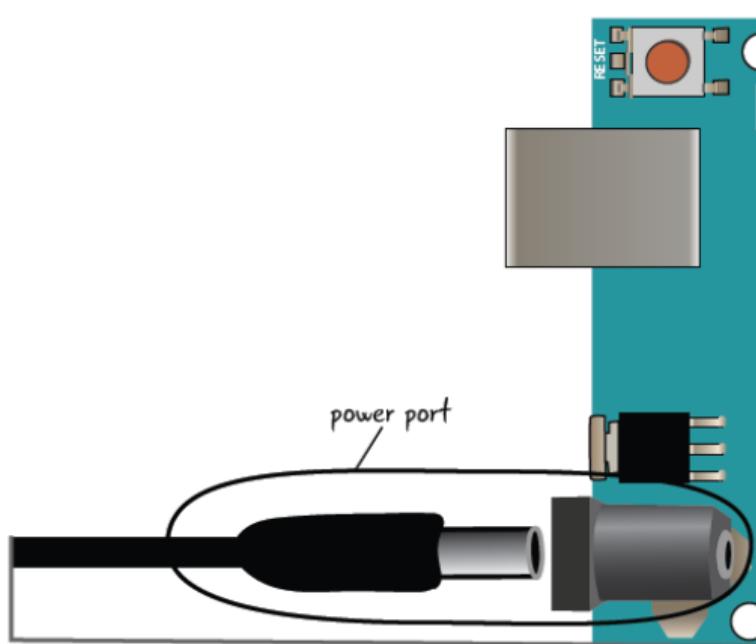


# Conecta tu Arduino al ordenador

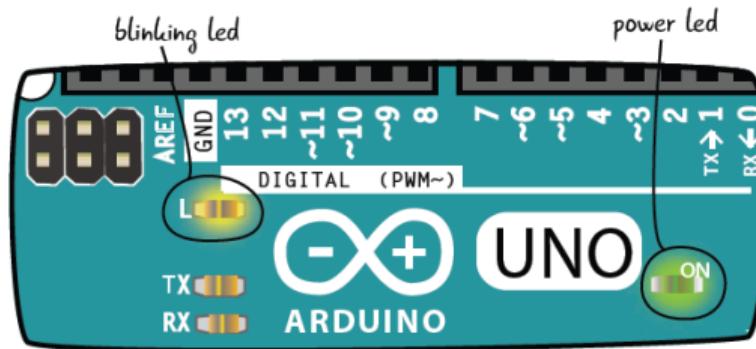


Los LEDs se encienden cuando el Arduino recibe energía del ordenador.

# Conecta tu Arduino al ordenador



# Conecta tu Arduino al ordenador



# Programación del Arduino

code for Arduino in this circuit

The screenshot shows the Arduino IDE interface with the title bar "LEA4\_Blink | Arduino 1.8.3". The code editor contains the following sketch:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO,
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino model, check
  the Technical Specs of your board at https://www.arduino.cc/mkR1000/Products

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald

  modified 2 Sep 2016
  by Arturo Guadalajara

  modified 8 Sep 2016
  by Colby Newren
 */

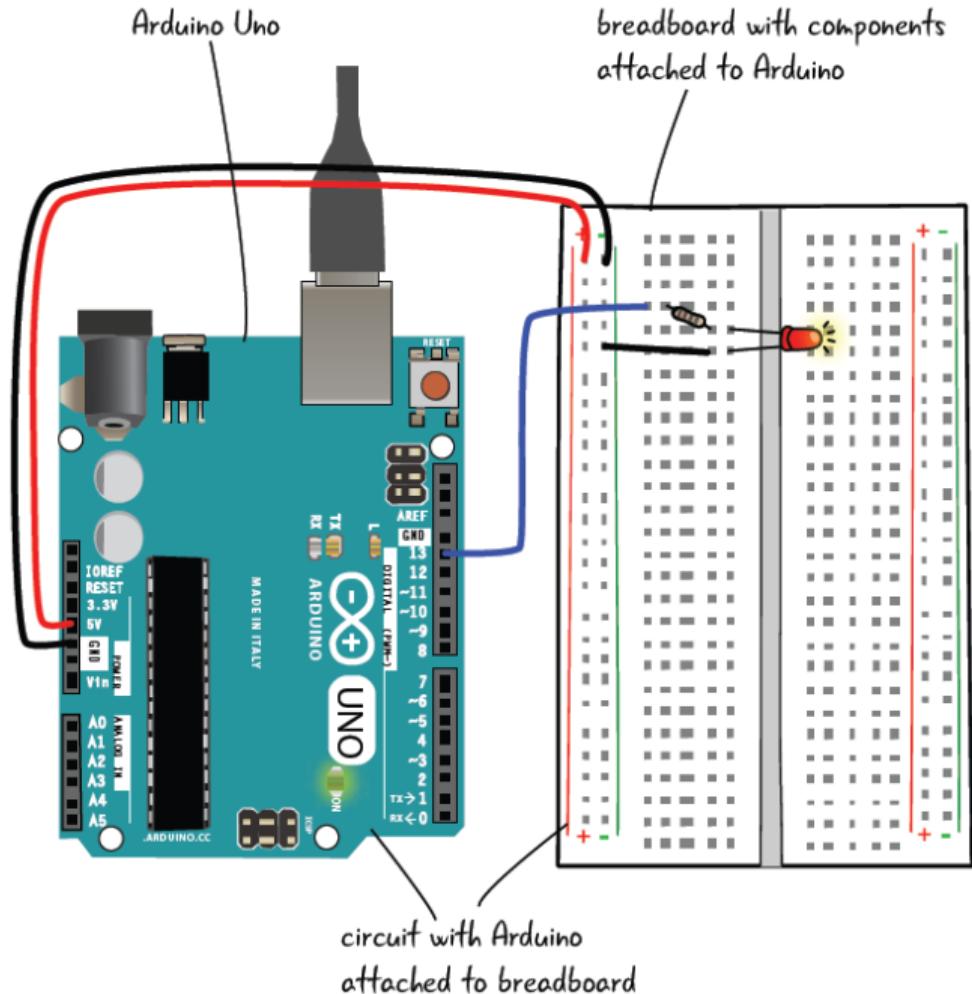
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Done Saving.
```

At the bottom of the IDE window, it says "Arduino/Genuino Uno on /dev/cu.usbmodem1421".

Arduino Uno

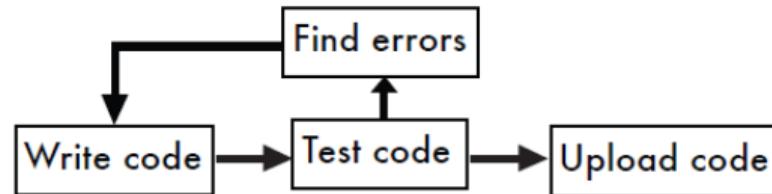


# ¿Qué es un IDE?

## IDE

Un entorno de desarrollo integrado (IDE) es una aplicación de software que permite escribir código y probar ese código en el lenguaje de programación que soporta el IDE.

- El equipo de Arduino ha diseñado un IDE para usar con sus dispositivos que tiene todas las características que necesitas.
- Tiene un editor de código incorporado



# Pasos para descargar e instalar el Arduino IDE

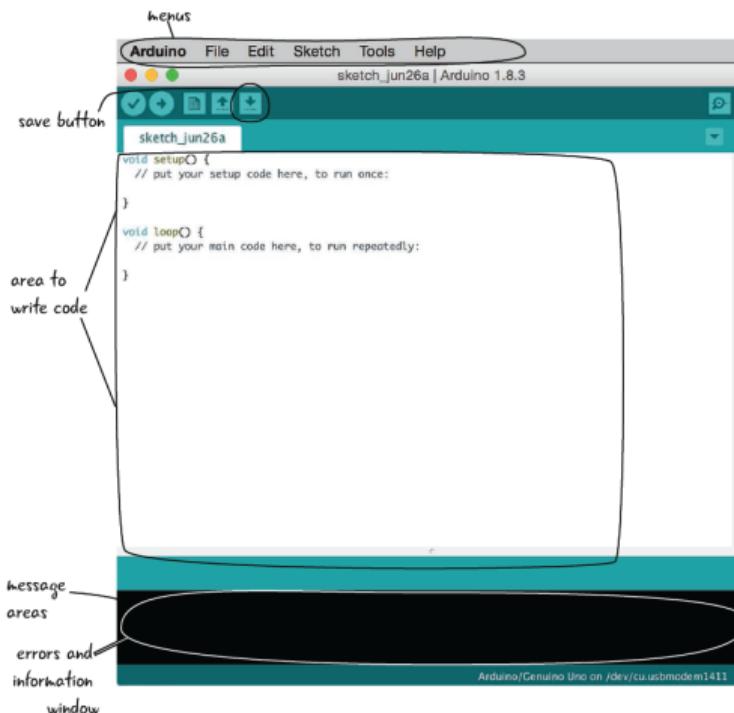
El IDE está disponible gratuitamente en el sitio web de Arduino

- 1 Descargar el software:** Ir a la página oficial de Arduino (<https://www.arduino.cc/en/software>) y seleccionar la versión adecuada para tu sistema operativo (Windows, Mac OS X o Linux).
- 2 Instalación:** Ejecutar el archivo descargado y seguir los pasos de la instalación. Aceptar los términos y condiciones y seleccionar la ruta de instalación preferida.
- 3 Verificación de la instalación:** Abre el Arduino IDE. Verifica si se pueden cargar ejemplos y si se reconoce el puerto serie y la placa.
- 4 Instalación de controladores:** Si la placa no es reconocida, es necesario instalar los controladores correspondientes.
- 5 Verificación final:** Conectar la placa y verificar que el Arduino IDE la reconoce. Puedes hacer esto seleccionando Herramientas>Placa y seleccionando la placa correcta.

# ¿Qué hay en el IDE de Arduino?

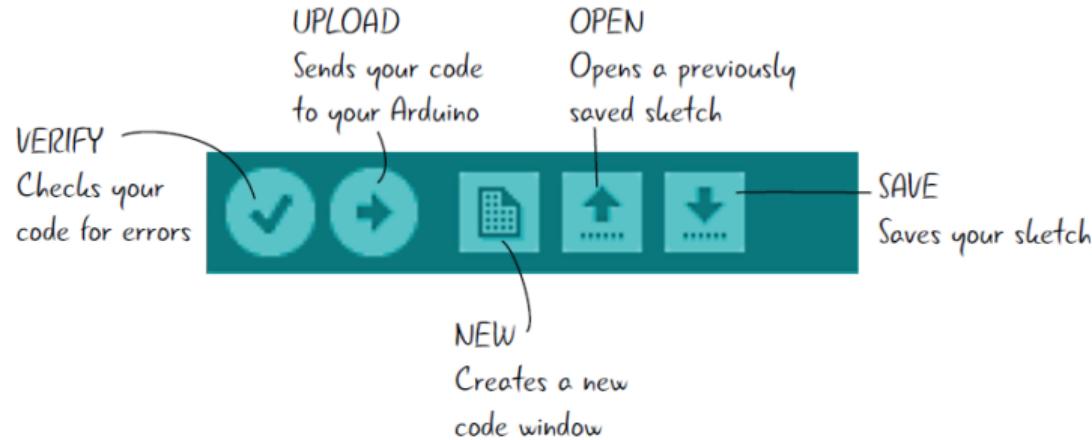
- Una ventana del editor de código donde se escribe el código
- Un área de mensajes que da información sobre su código
- Una consola de errores que ofrece información detallada y ayuda a la depuración
- Menús que te permiten configurar las propiedades de tu Uno y cargar ejemplos de código y otras funciones
- Botones para comprobar el código, cargarlo en Arduino, guardar el código, crear una nueva ventana de código, etc.

# Arduino IDE



# Arduino IDE

## Botones en el IDE de Arduino



**Familia ESP**

# Familia de microncontroladores ESP



## Familia de microcontroladores ESP

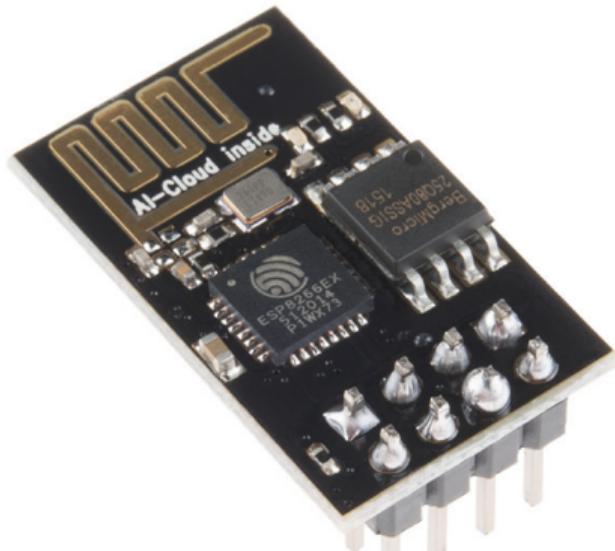
Es una serie de microcontroladores desarrollados por **Espressif Systems** que incluyen la serie ESP8266 y ESP32. Estos microcontroladores son conocidos por ser eficientes en términos de energía y de bajo costo, y tienen una amplia gama de aplicaciones en el Internet de las cosas (IoT), la automatización del hogar y proyectos de robótica.

- Ambos dispositivos incluyen Wi-Fi integrado, lo que los hace ideales para conectarse a Internet y enviar o recibir datos.
- También tienen una amplia comunidad de desarrolladores y una gran cantidad de documentación y recursos disponibles en línea.

**ESP-01**

# ESP8266 V1.0

- El ESP8266 es un chip de bajo costo Wi-Fi con un stack TCP/IP completo y un microcontrolador, fabricado por Espressif, una empresa afincada en Shanghái, China<sup>1</sup>
- El primer chip se hace conocido en los mercados alrededor de agosto de 2014 con el módulo ESP-01, desarrollado por la empresa AI-Thinker.
- Este pequeño módulo permite a otros microcontroladores conectarse a una red inalámbrica Wi-Fi y realizar conexiones simples con TCP/IP usando comandos al estilo Hayes.<sup>2</sup>

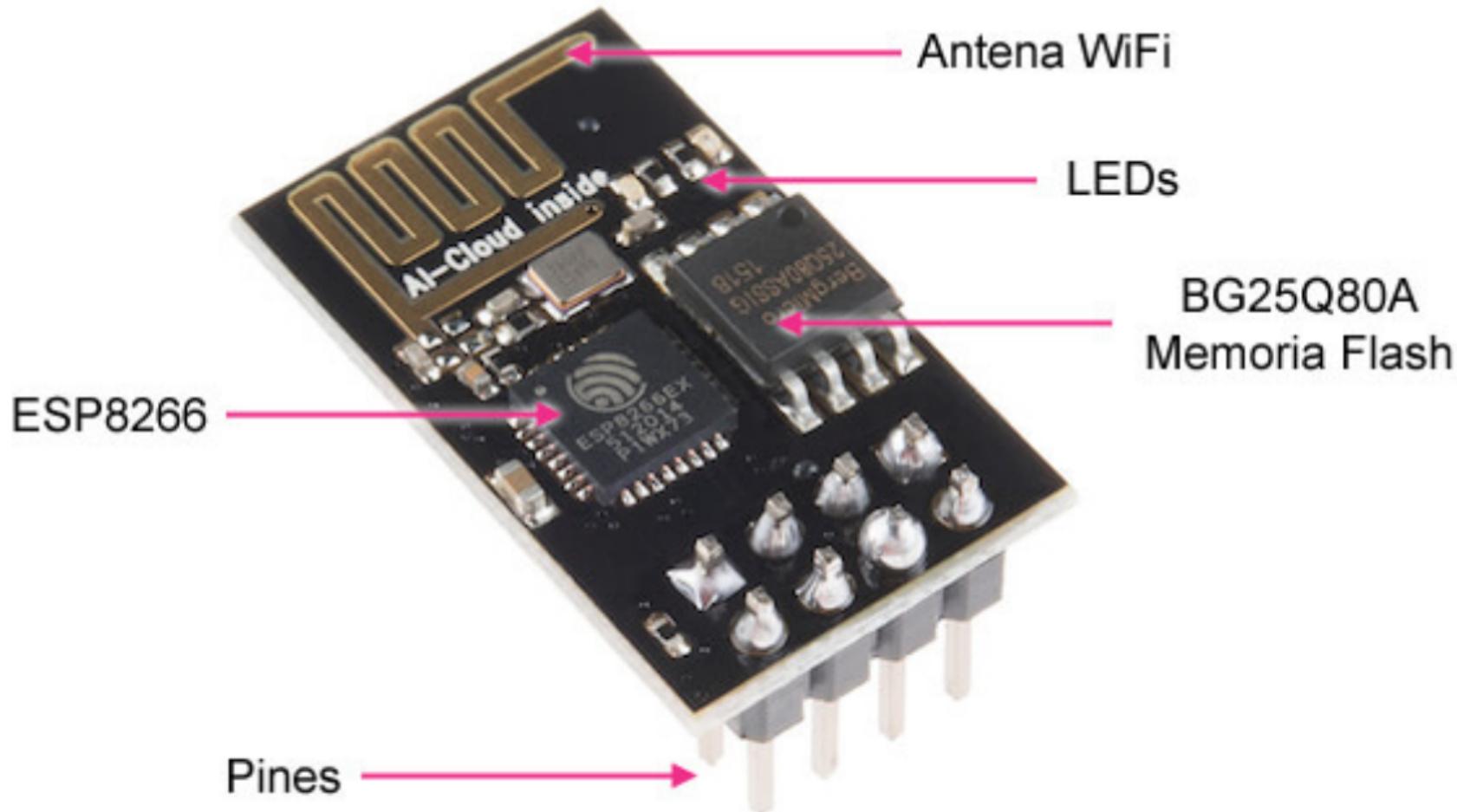


# ESP8266 V1.0

- El ESP8285 es como un ESP8266 pero con 1 MB de memoria flash interna, para permitir a dispositivos de un chip conexiones de Wi-Fi.3
- El sucesor de estos módulos es el ESP32.

# Características

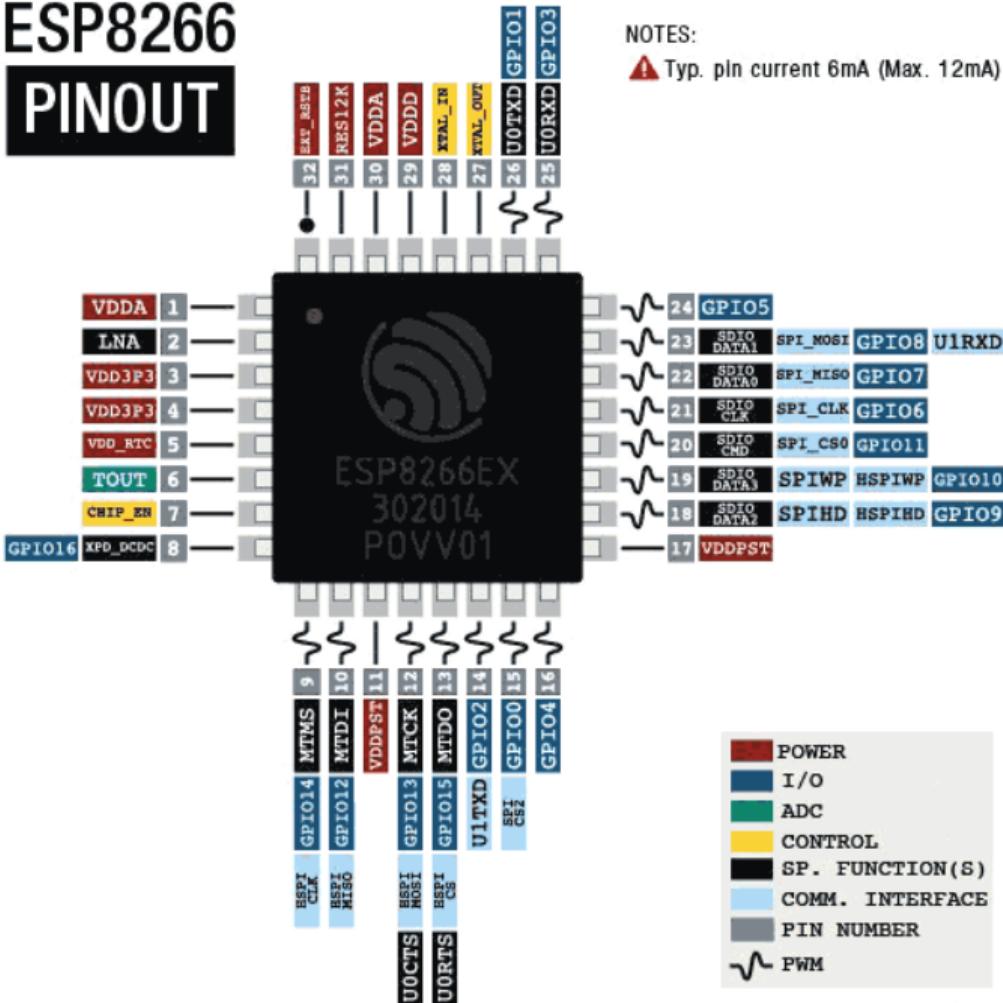
- CPU RISC de 32-bit: Tensilica Xtensa LX106 a un reloj de 80 MHz
- RAM de instrucción de 64 KB, RAM de datos de 96 KB
- Capacidad de memoria externa flash QSPI - 512 KB a 4 MB\* (puede soportar hasta 16 MB)
- IEEE 802.11 b/g/n Wi-Fi
  - Tiene integrados: TR switch, balun, LNA, amplificador de potencia de RF y una red de adaptación de impedancias
  - Soporte de autenticación WEP y WPA/WPA2
- 16 pines GPIO (Entradas/Salidas de propósito general)
- SPI, I2C,
- Interfaz I2S con DMA (comparte pines con GPIO)
- Pines dedicados a UART, más una UART únicamente para transmisión que puede habilitarse a través del pin GPIO2
- 1 conversor ADC de 10-bit



# ESP8266-IC



# ESP8266 PINOUT

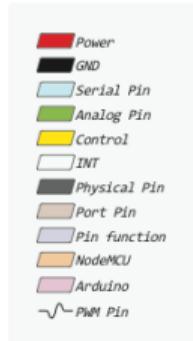
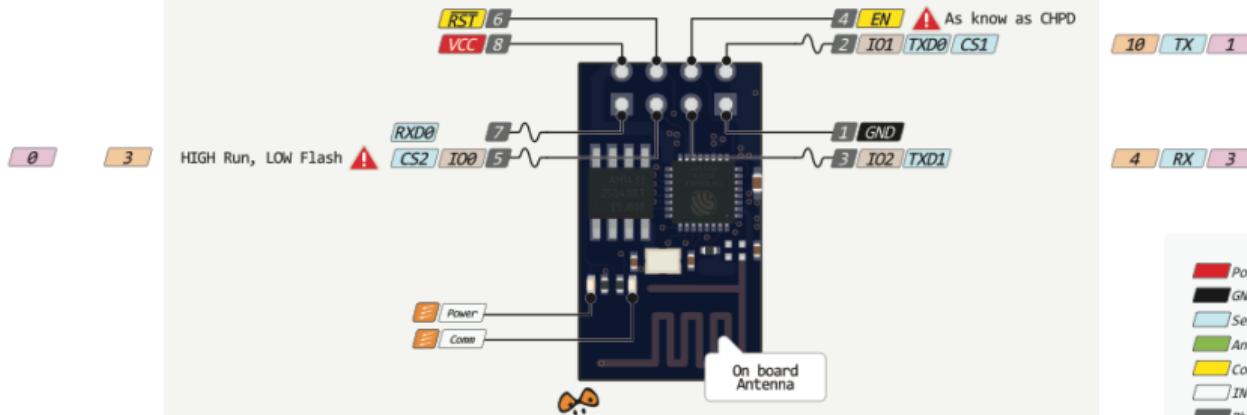


# ESP8266 PINOUT

 All ESP8266 in/outs  
are NOT 5V tolerant!

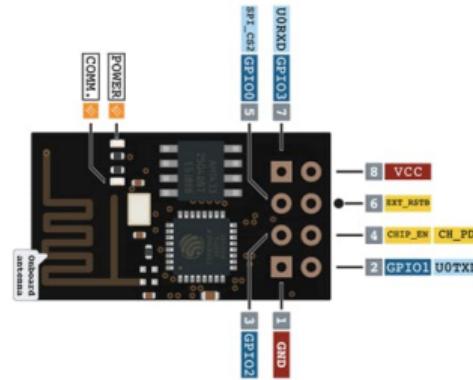
ESP-01

**Absolute MAX per pin 12mA  
recommended 6mA**



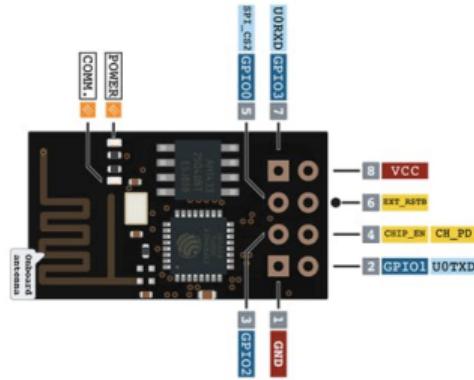
# ESP8266 pinout

- GND es la toma de tierra.
- GPIO2 es una entrada salida de propósito general. Es el pin digital número 2.
- GPIO0 es una entrada salida de propósito general. Es el pin digital número 0.



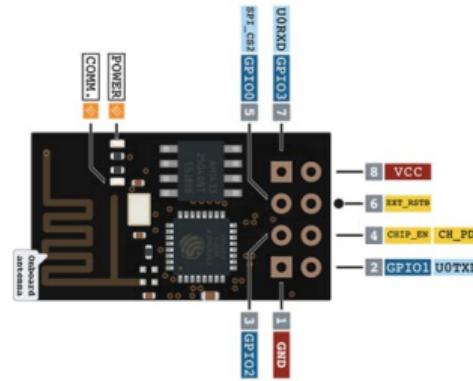
# ESP8266 pinout

- RXD es el pin por donde se van a recibir los datos del puerto serie. Trabaja a 3,3 V.  
También se puede utilizar
  - como pin digital GPIO: sería el número 3.
- TXD es el pin por donde se van a transmitir los datos del puerto serie. Trabaja a 3,3 V.  
También se puede utilizar como pin digital GPIO: sería el número 1.

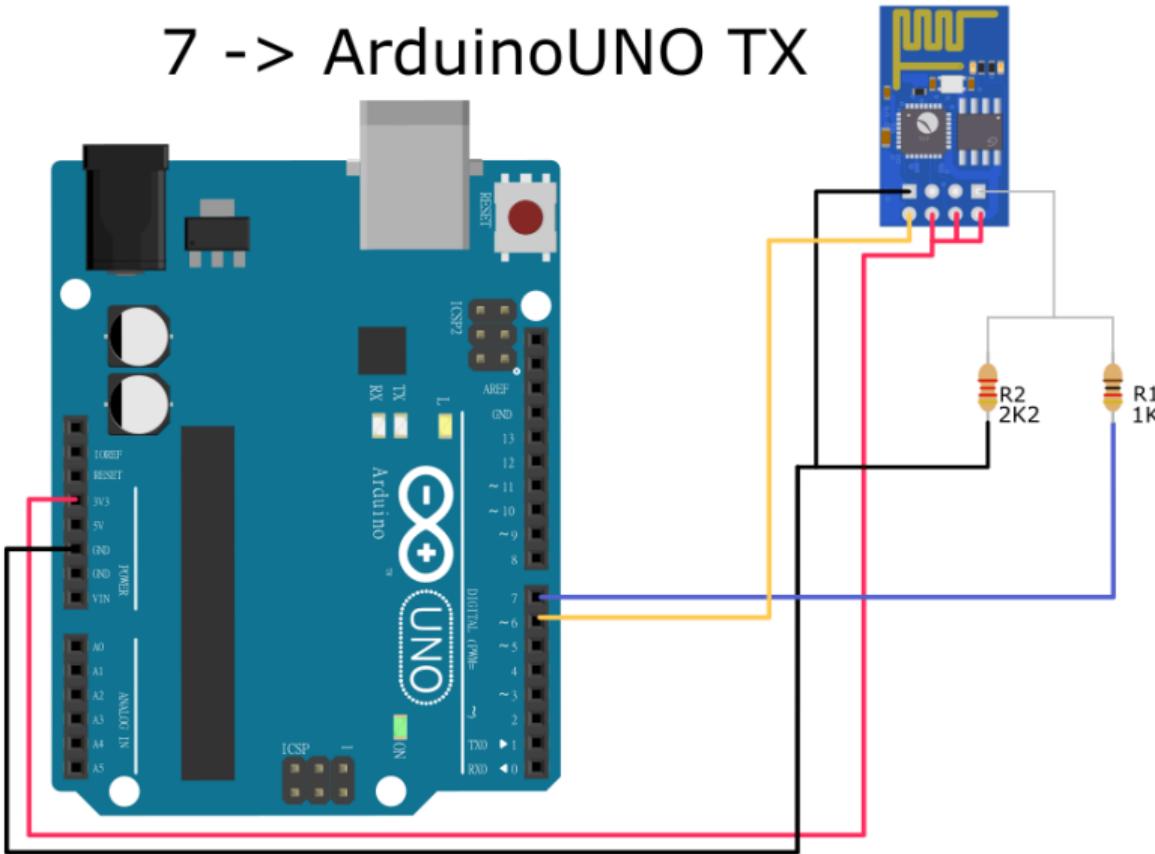


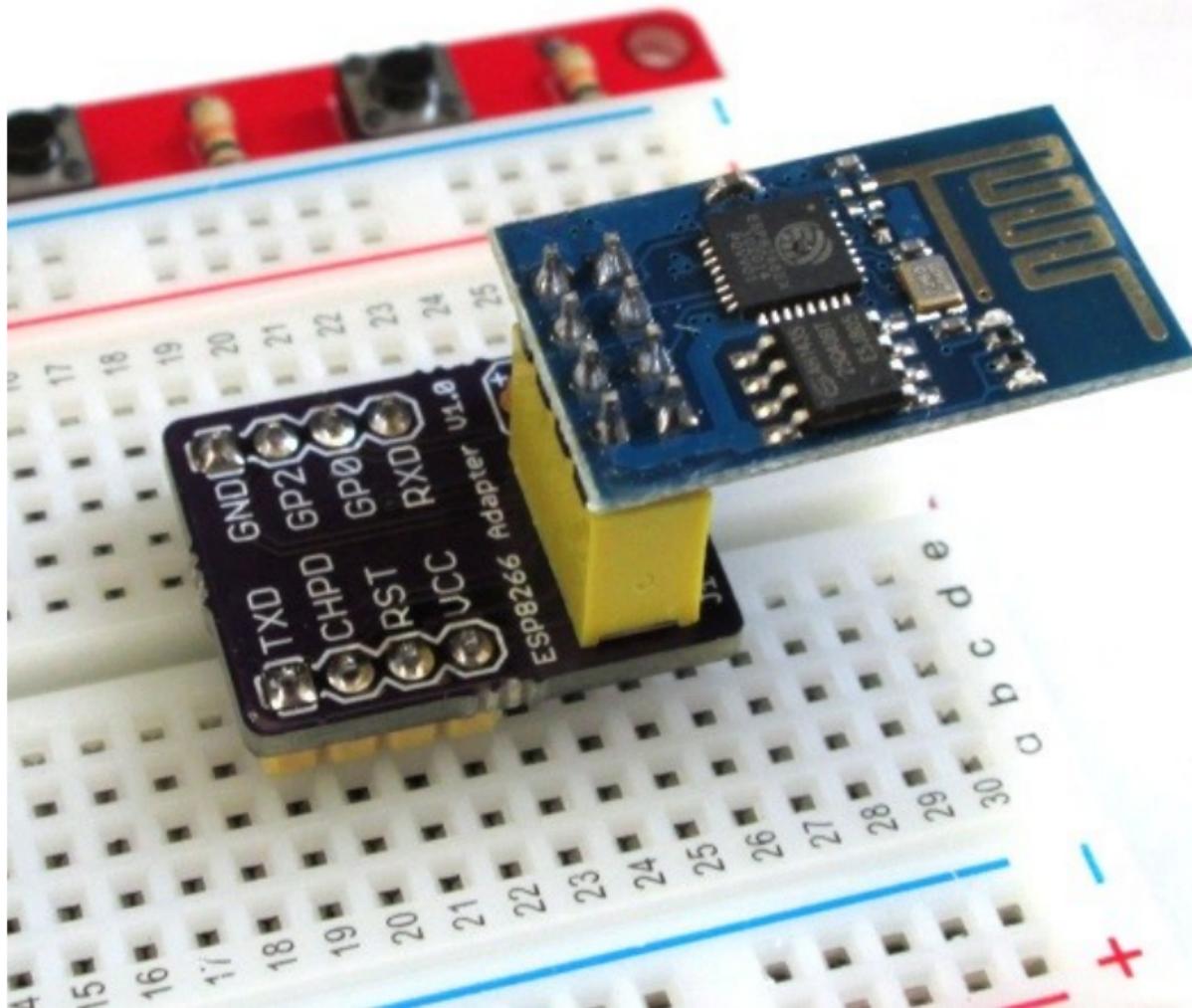
# ESP8266 pinout

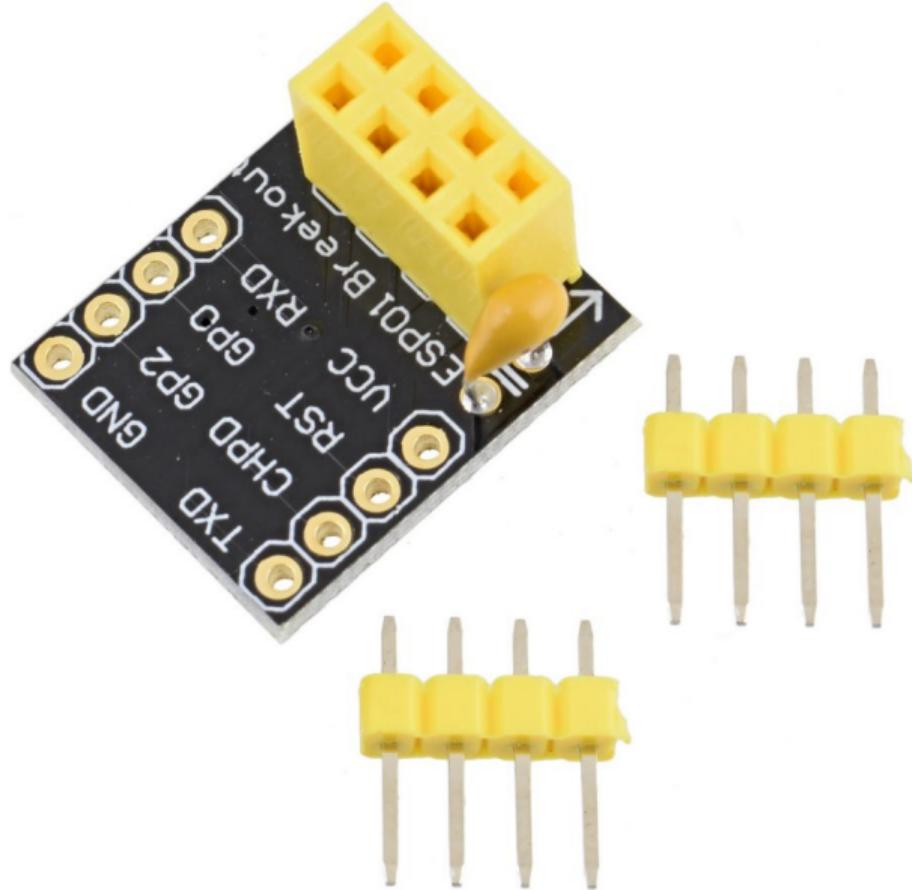
- CH\_PD pin para apagar y encender el ESP-01: si lo ponemos a 0 V (LOW) se apaga, y a 3,3 V (HIGH) se enciende.
- RESET pin para resetear el ESP-01: si lo ponemos a 0 V (LOW) se resetea.
- Vcc es por donde alimentamos el ESP-01. Funciona a 3,3 V y admite un máximo de 3,6 V. La corriente suministrada debe ser mayor que 200 mA.



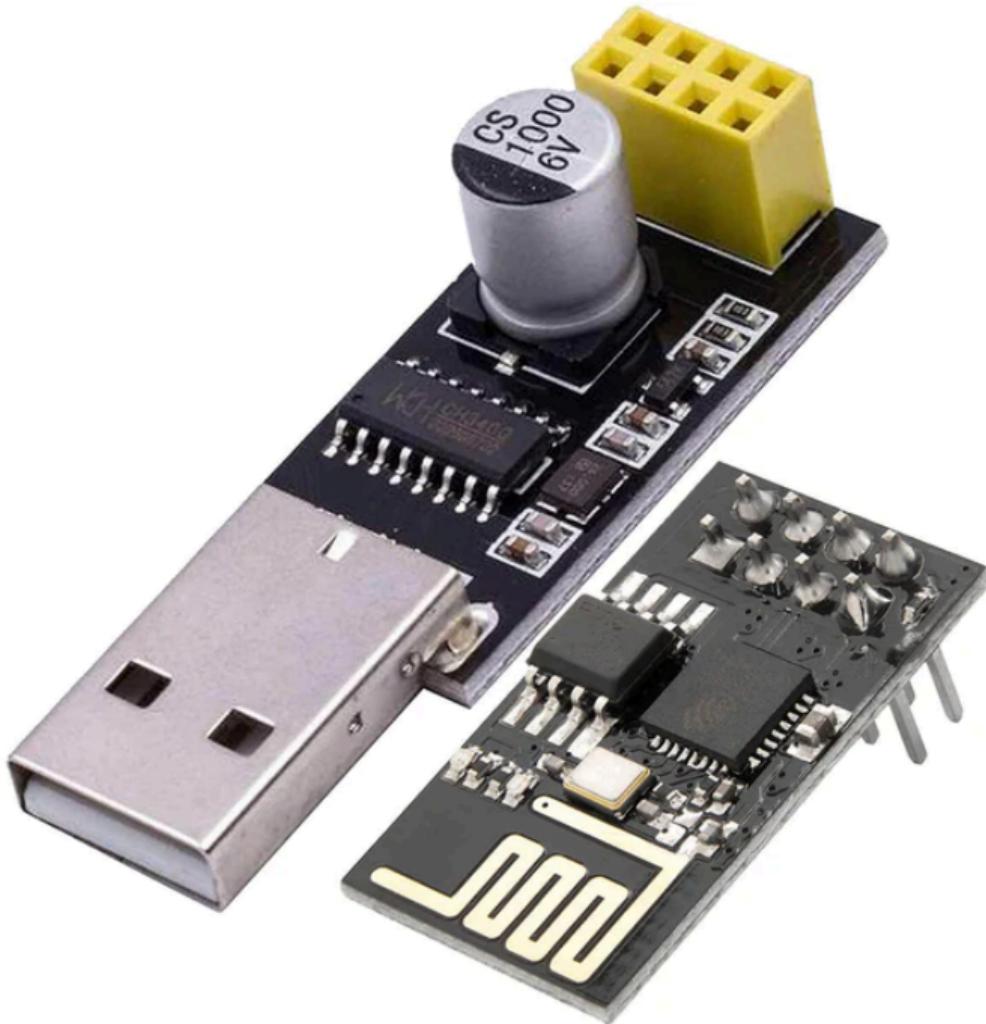
SoftwareSerial  
6 -> ArduinoUNO RX  
7 -> ArduinoUNO TX



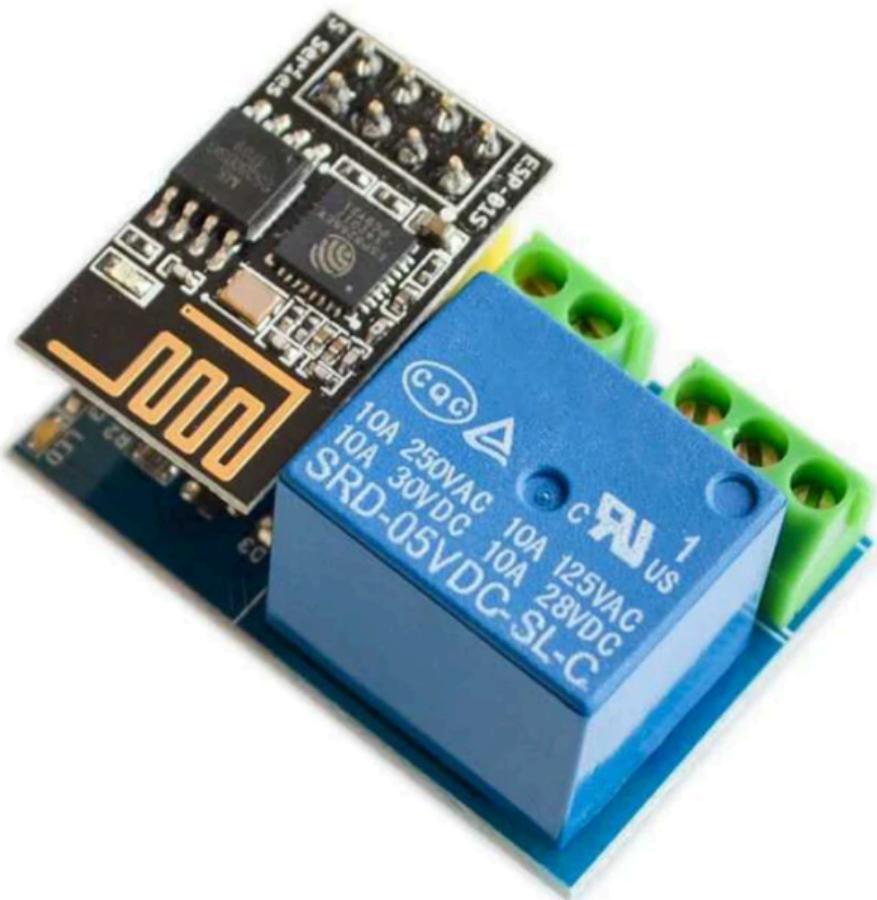




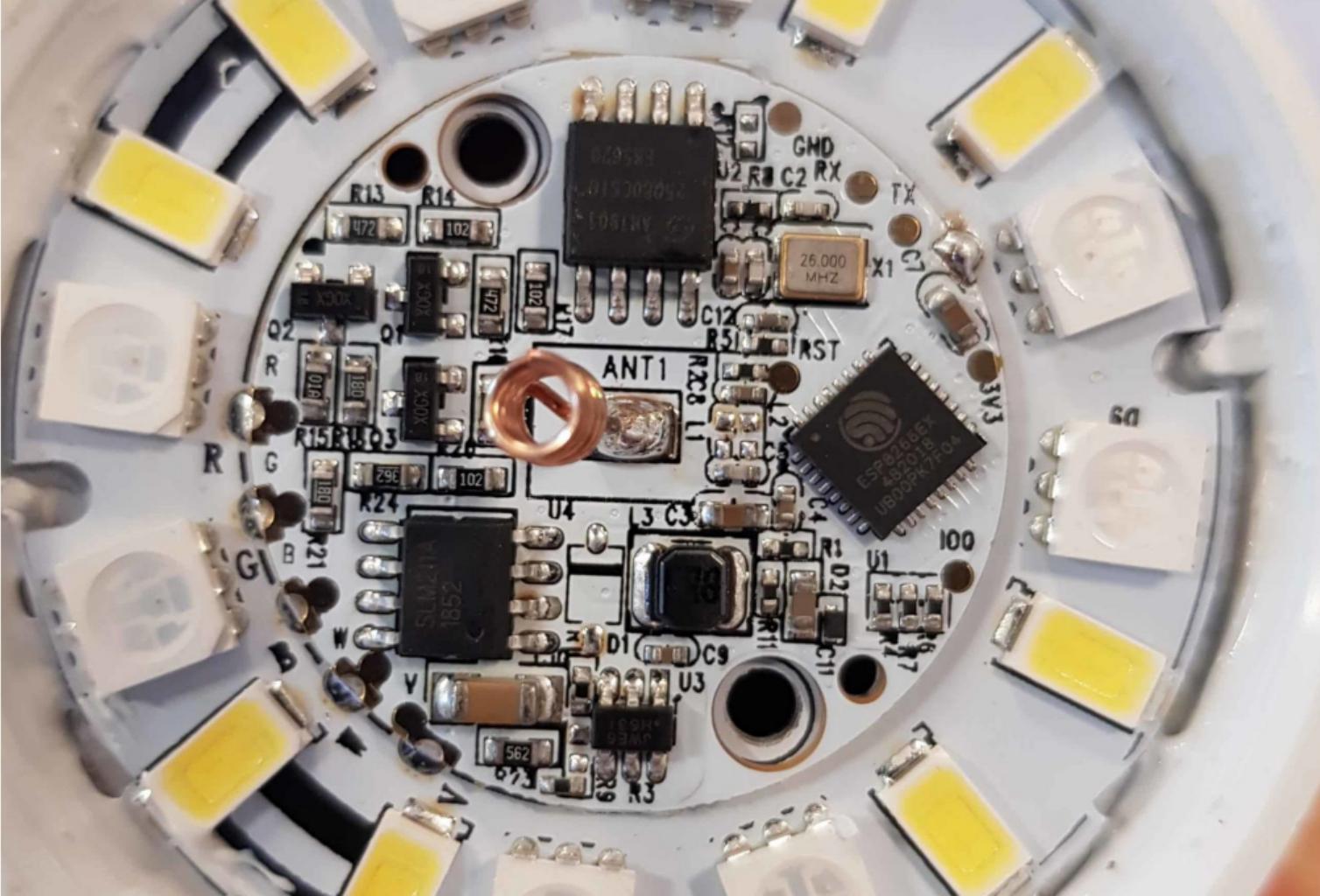


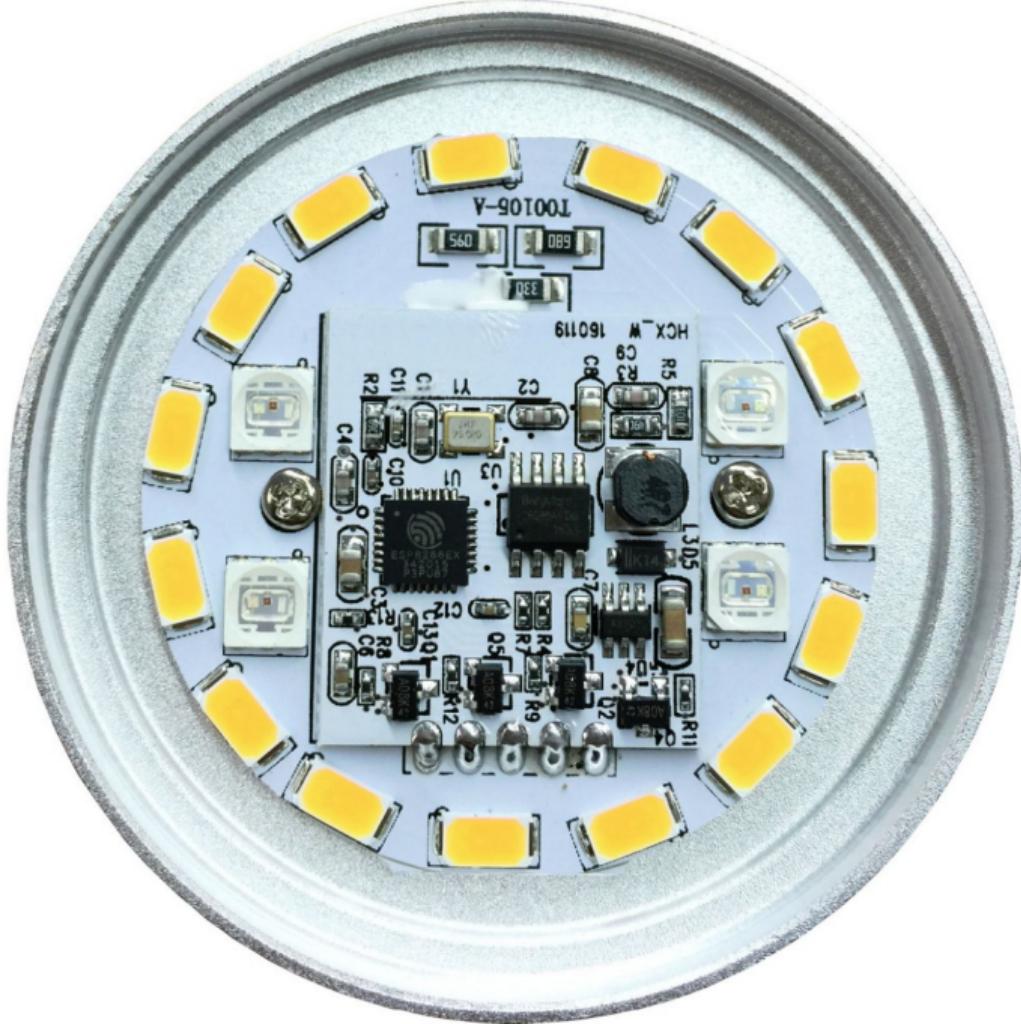




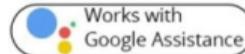
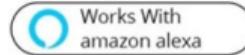








# Sonoff®



# ***MINI WiFi Switch***



Voice  
Control



Group  
Control



App  
Control

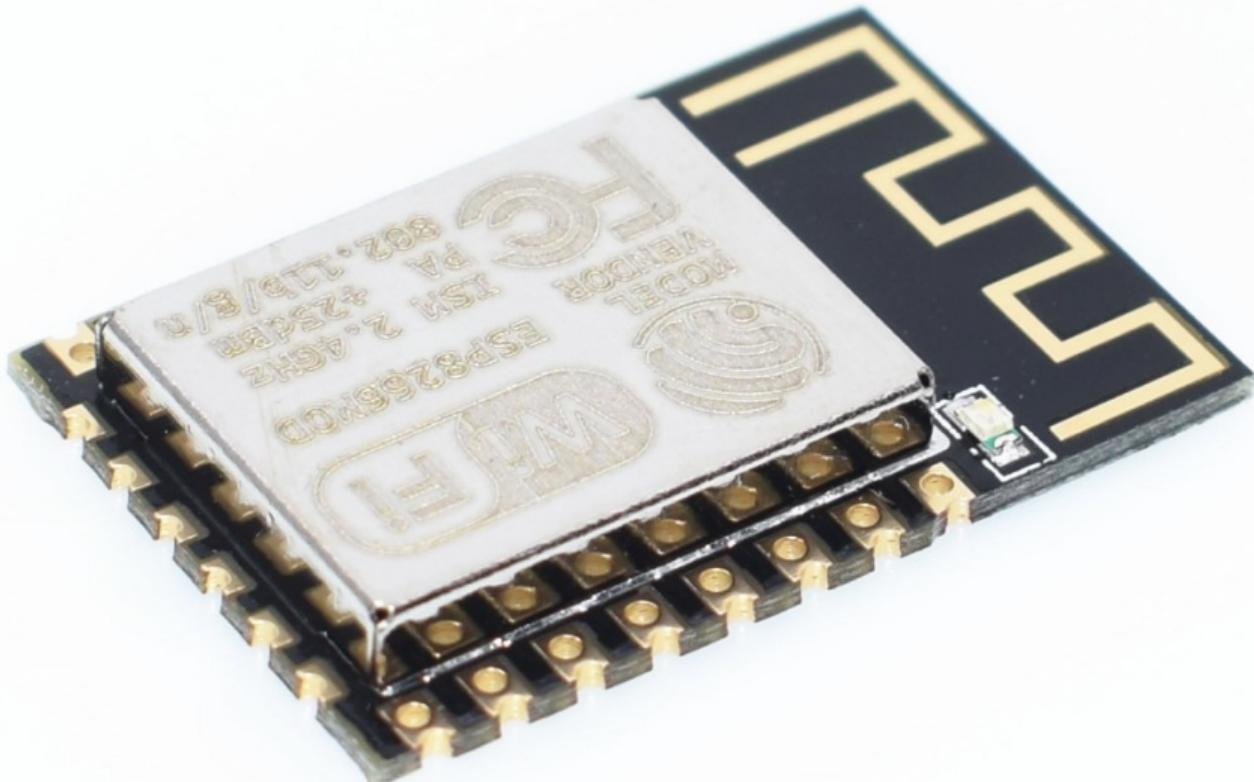


Timing  
Countdown





**ESP8266 ESP-12**



EYEWINK



ESP-12E



ESP-12S

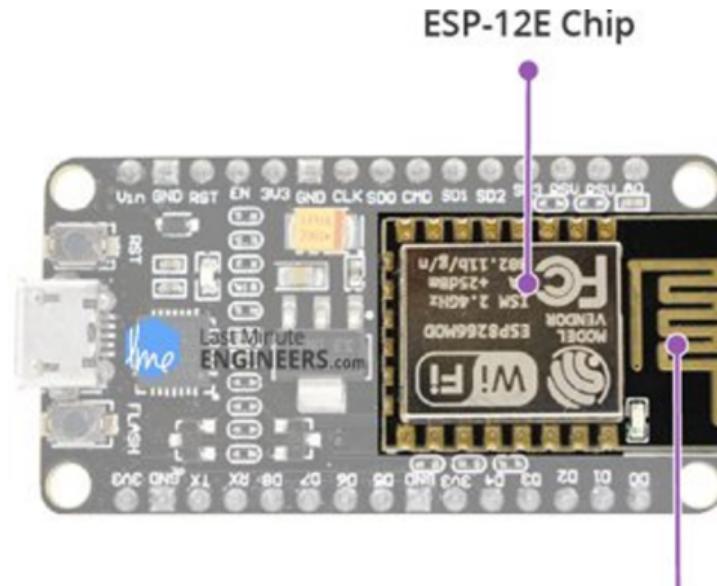


ESP-12F



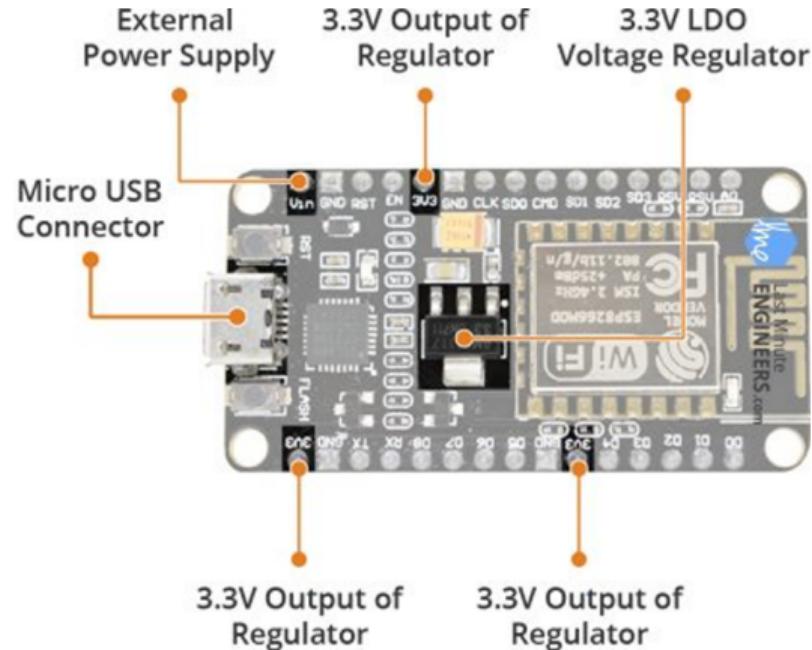
# Características

- ESP-12E Chip
- Tensilica Xtensa® 32-bit LX106
- 80 to 160 MHz Clock Freq.
- 128kB internal RAM
- 4MB external flash
- 802.11b/g/n Wi-Fi transceiver



# Power Requirement

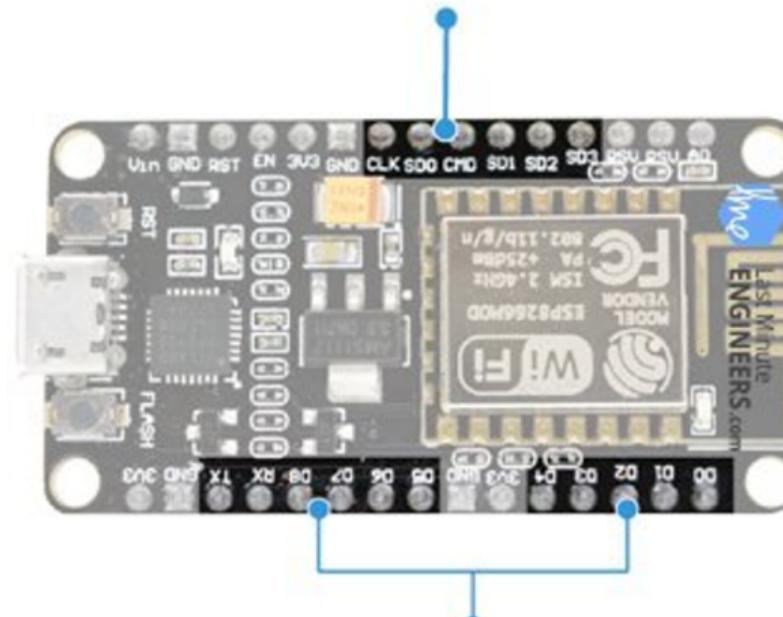
- Operating Voltage: 2.5V to 3.6V
- On-board 3.3V 600mA regulator
- 80mA Operating Current
- 20  $\mu$ A during Sleep Mode



# Peripherals and I/O

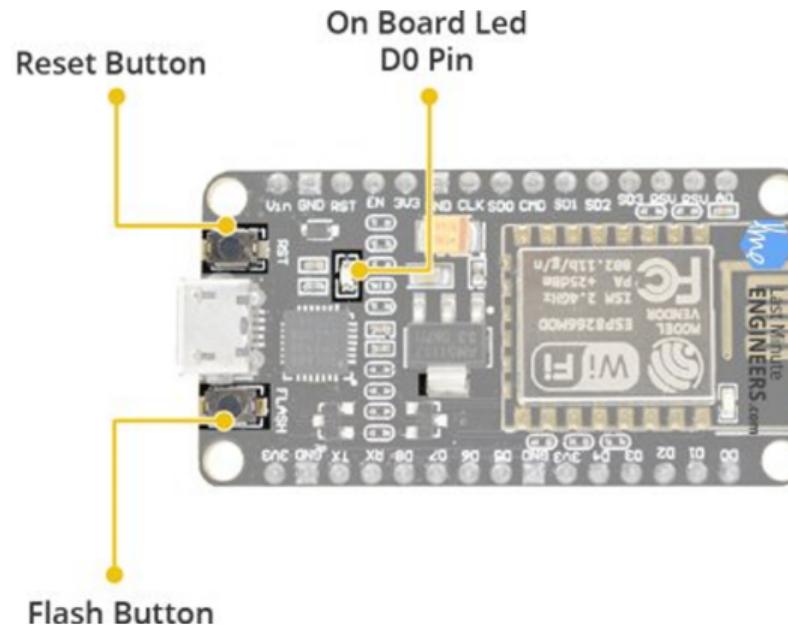
- 1 ADC channels
- 2 UART interfaces
- 4 PWM outputs
- SPI, I2C & I2S interface

Multiplexed GPIO



# On-board Switches & LED Indicators

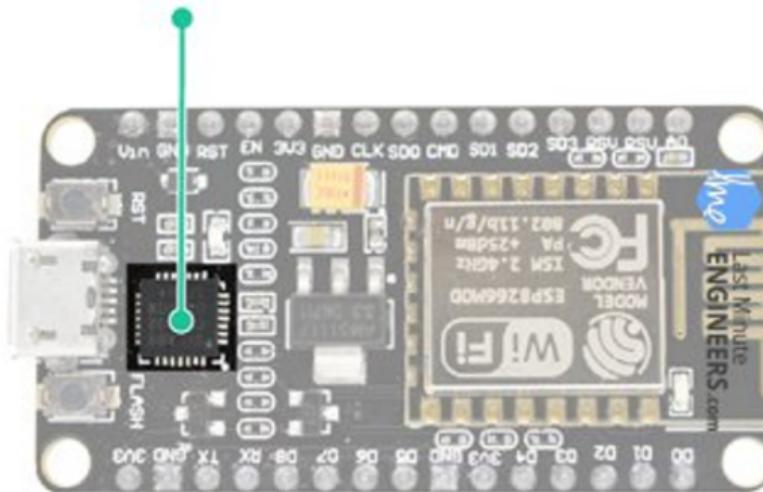
- Switches & Indicators
- RST – Reset the ESP8266 chip
- FLASH – Download new programs
- Blue LED – User Programmable



# Serial Communication

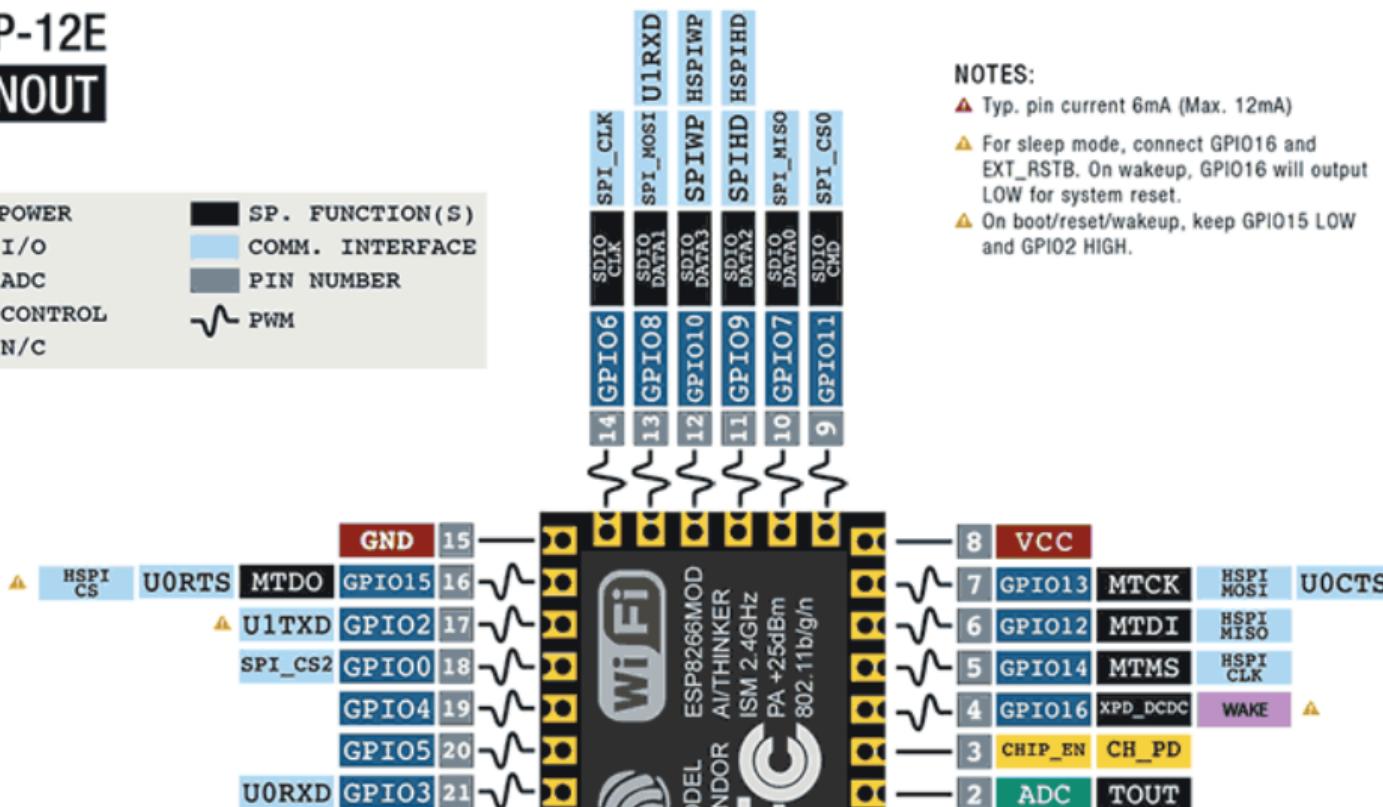
- CP2102 USB-to-UART converter
- 4.5 Mbps communication speed
- Flow Control support

USB To TTL Converter  
CP2102



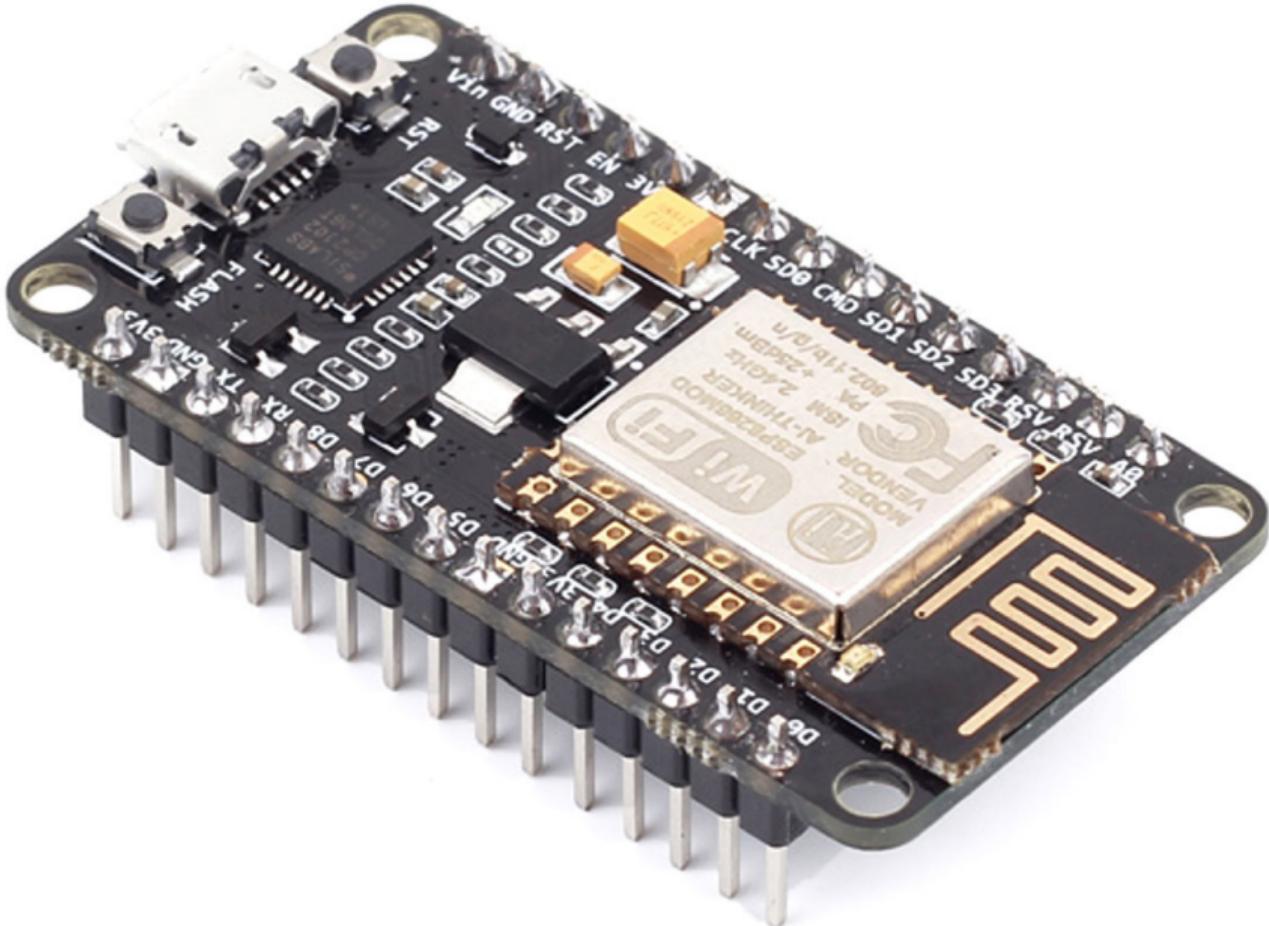
# ESP-12E

## PINOUT



### NOTES:

- ▲ Typ. pin current 6mA (Max. 12mA)
- ⚠ For sleep mode, connect GPIO16 and EXT\_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- ⚠ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.



# ESP-32

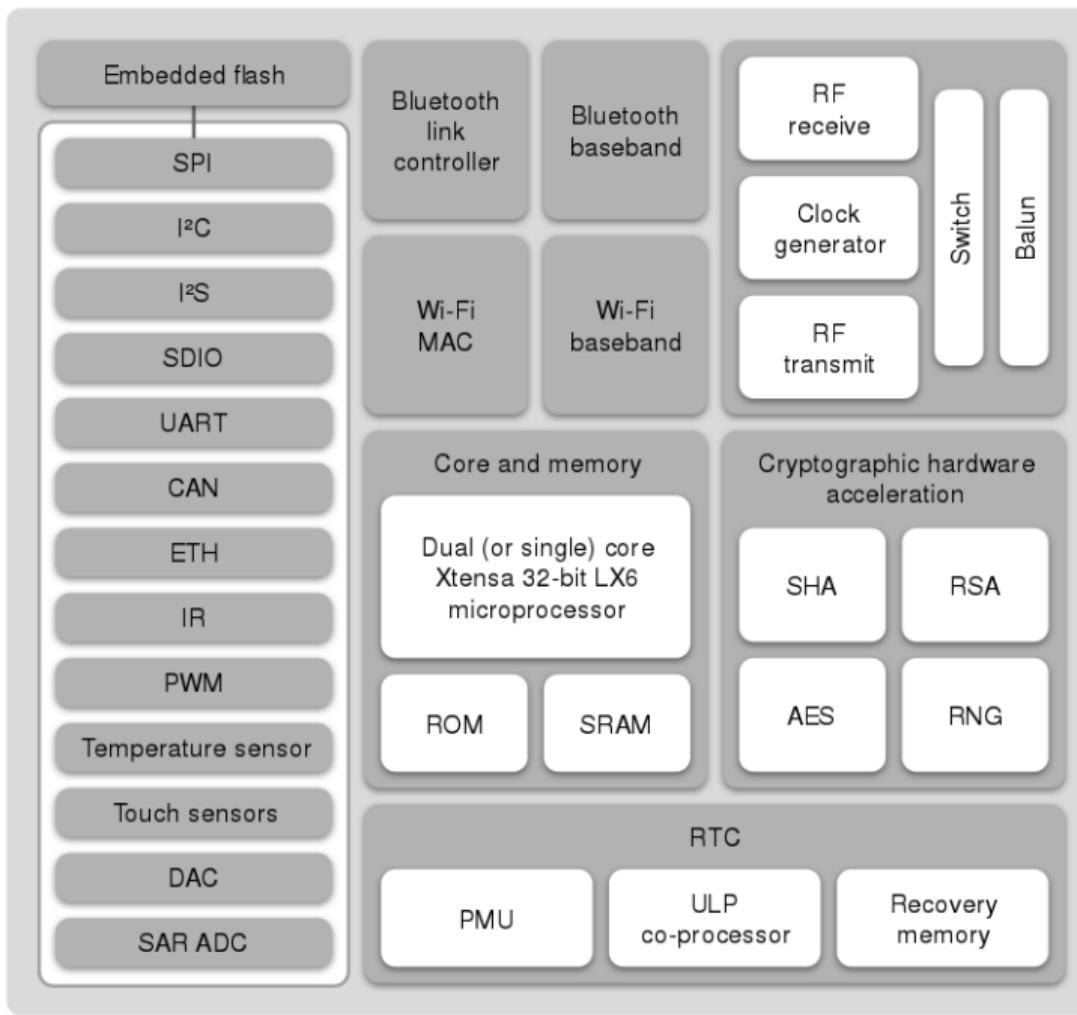
# ESP-32

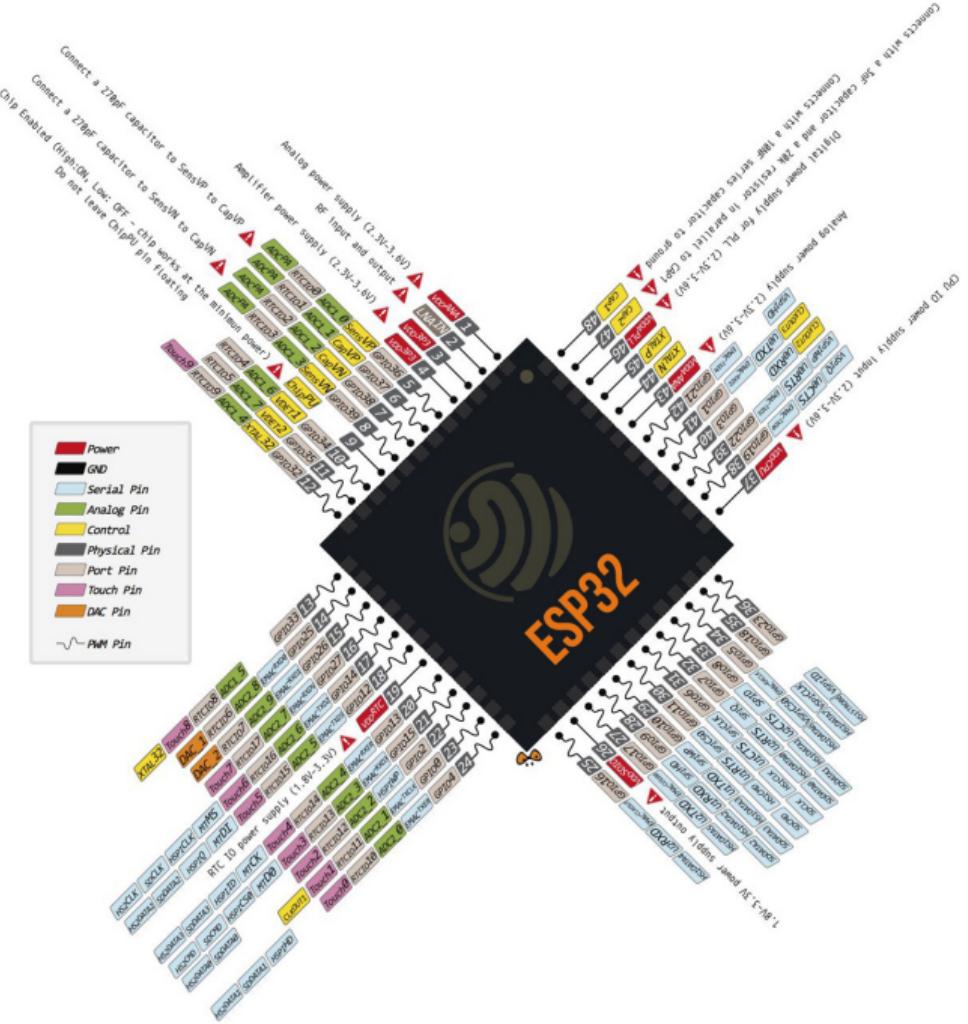


## ESP32 características

- Voltaje de Alimentación (USB): 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC
- Placa: ESP32 DEVKIT V1 (Espressif)
- SoM: ESP-WROOM-32 (Espressif)
- SoC: ESP32 (ESP32-D0WDQ6)
- CPU: Dual core Tensilica Xtensa LX6 (32 bit)







## ESP32 características

- Frecuencia de Reloj: hasta 240Mhz
- Desempeño: Hasta 600 DMIPS
- Procesador secundario: Permite hacer operaciones básica en modo de ultra bajo consumo
- Wifi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz hasta 150 Mbit/s)
- Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE)

# ESP32 características

## ■ Memoria:

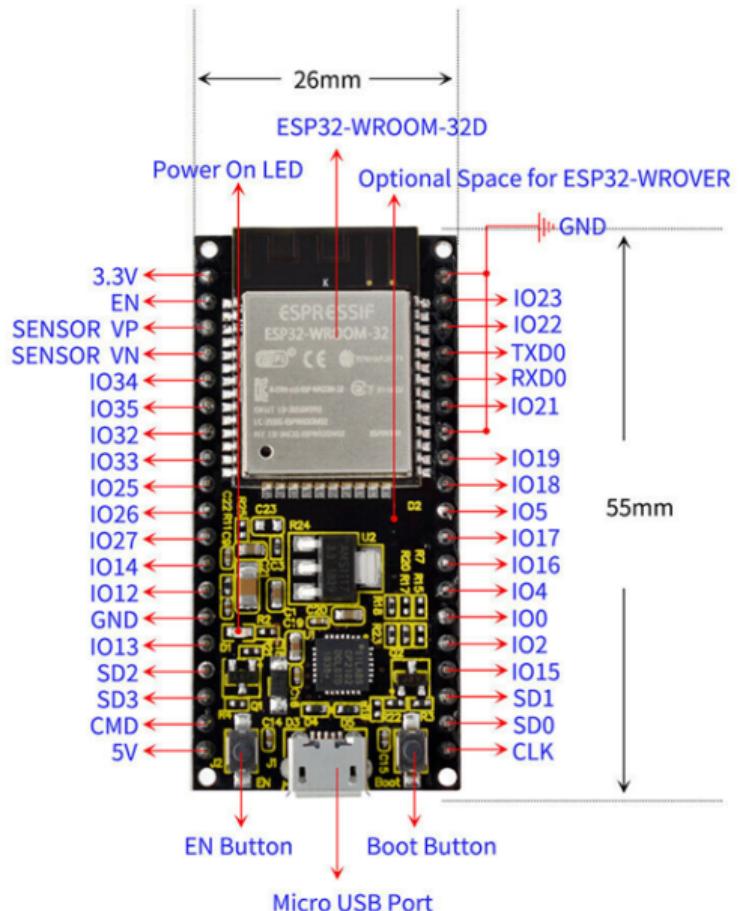
- 448 KByte ROM
- 520 KByte SRAM
- 16 KByte SRAM in RTC
- QSPI Flash/SRAM, 4 MBytes

# Características del ESP32

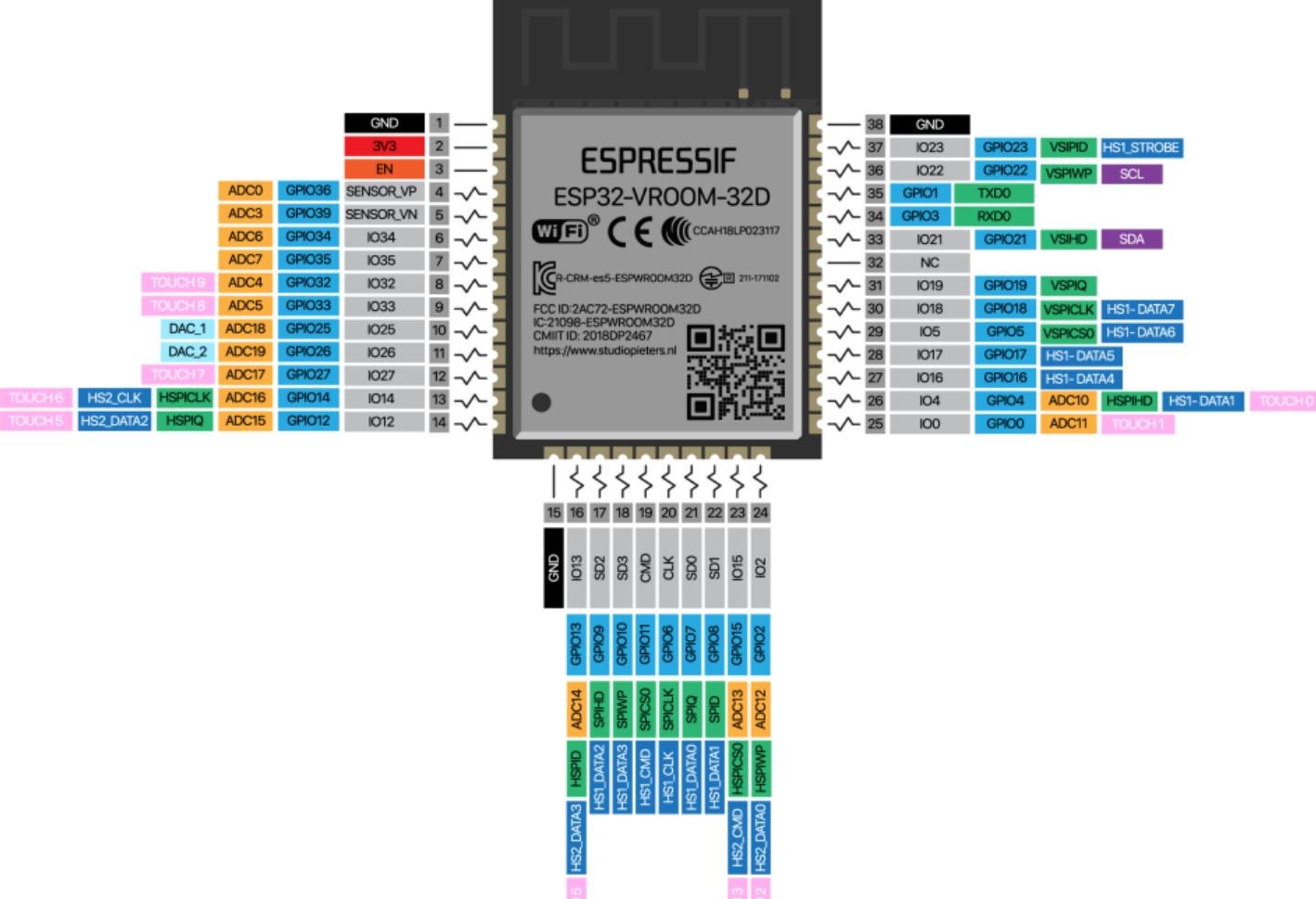
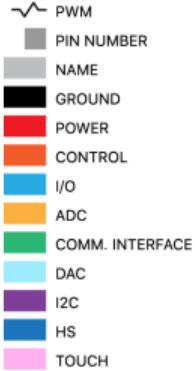
- Pines Digitales GPIO: 24 (Algunos pines solo como entrada)
- Pines PWM: 16
- Pines Analógicos ADC: 18 (3.3V, 12bit: 4095, tipo SAR, ganancia programable)
- Conversor Digital a Analógico DAC: 2 (8bit)
- UART: 2
- Chip USB-Serial: CP2102

# Características del ESP32

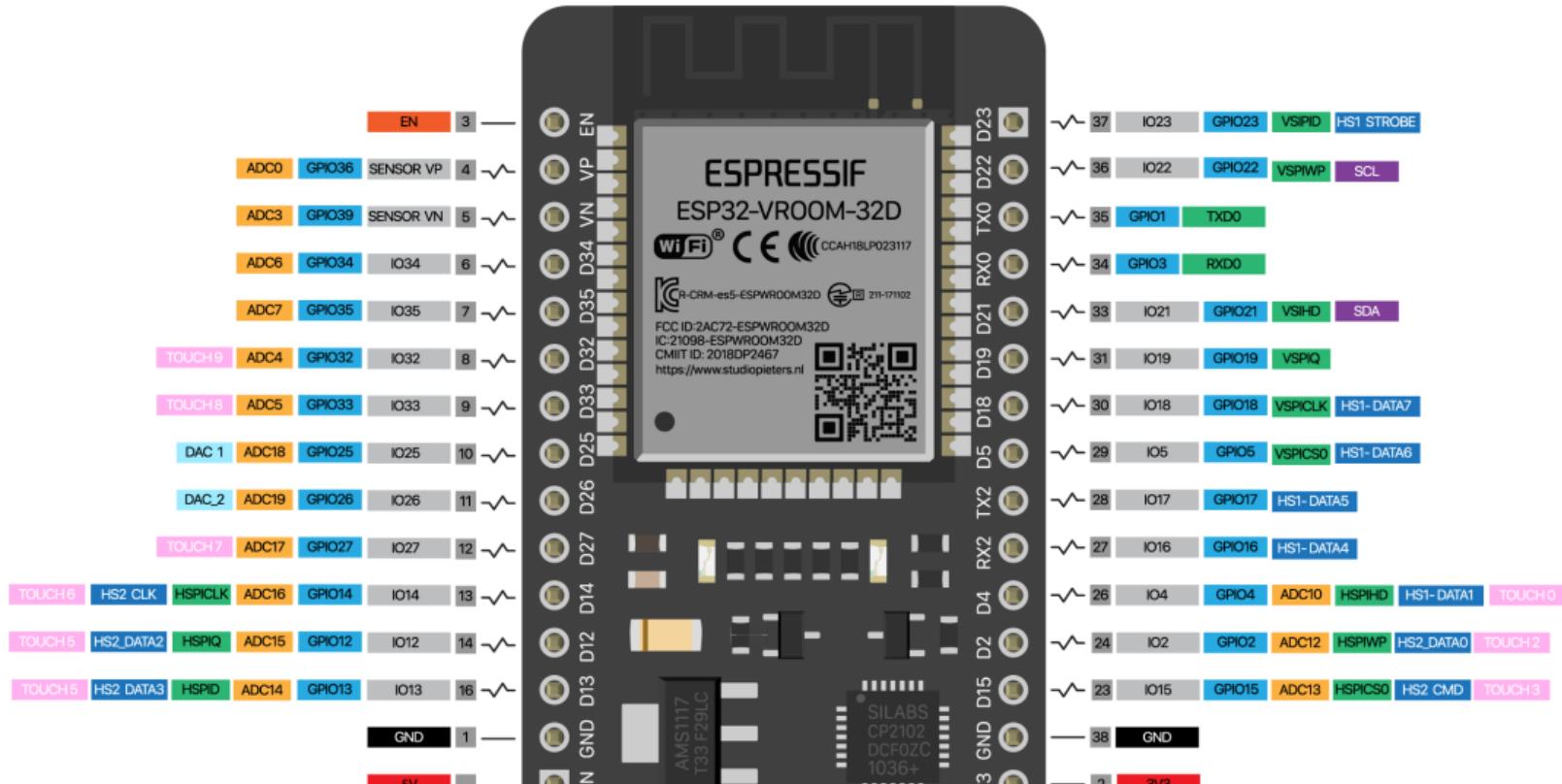
- Antena en PCB
- Seguridad:
  - Estandares IEEE 802.11 incluyendo WFA, WPA/WPA2 and WAPI
  - 1024-bit OTP, up to 768-bit for customers
  - Aceleración criptográfica por hardware: AES, HASH (SHA-2), RSA, ECC, RNG
- Dimensiones: 55\*28 mm



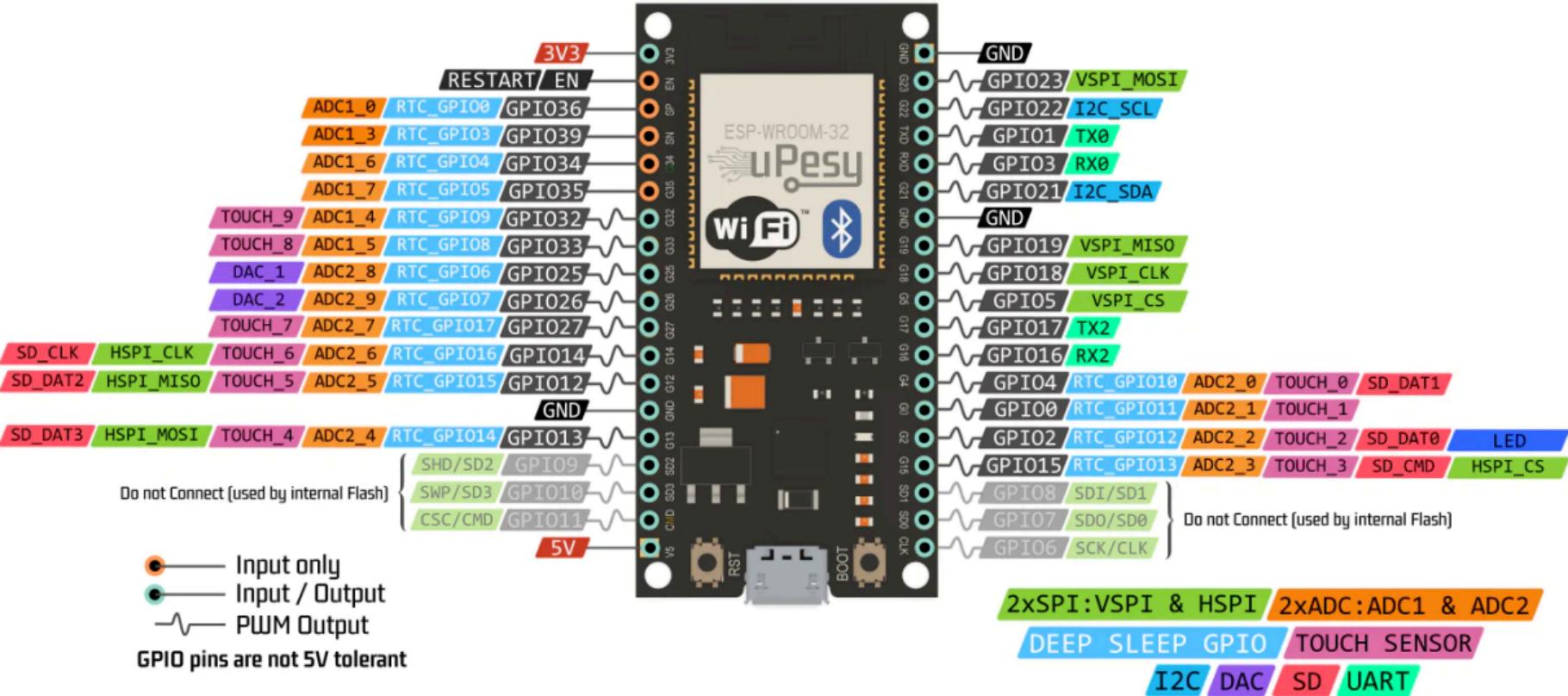
H:13mm

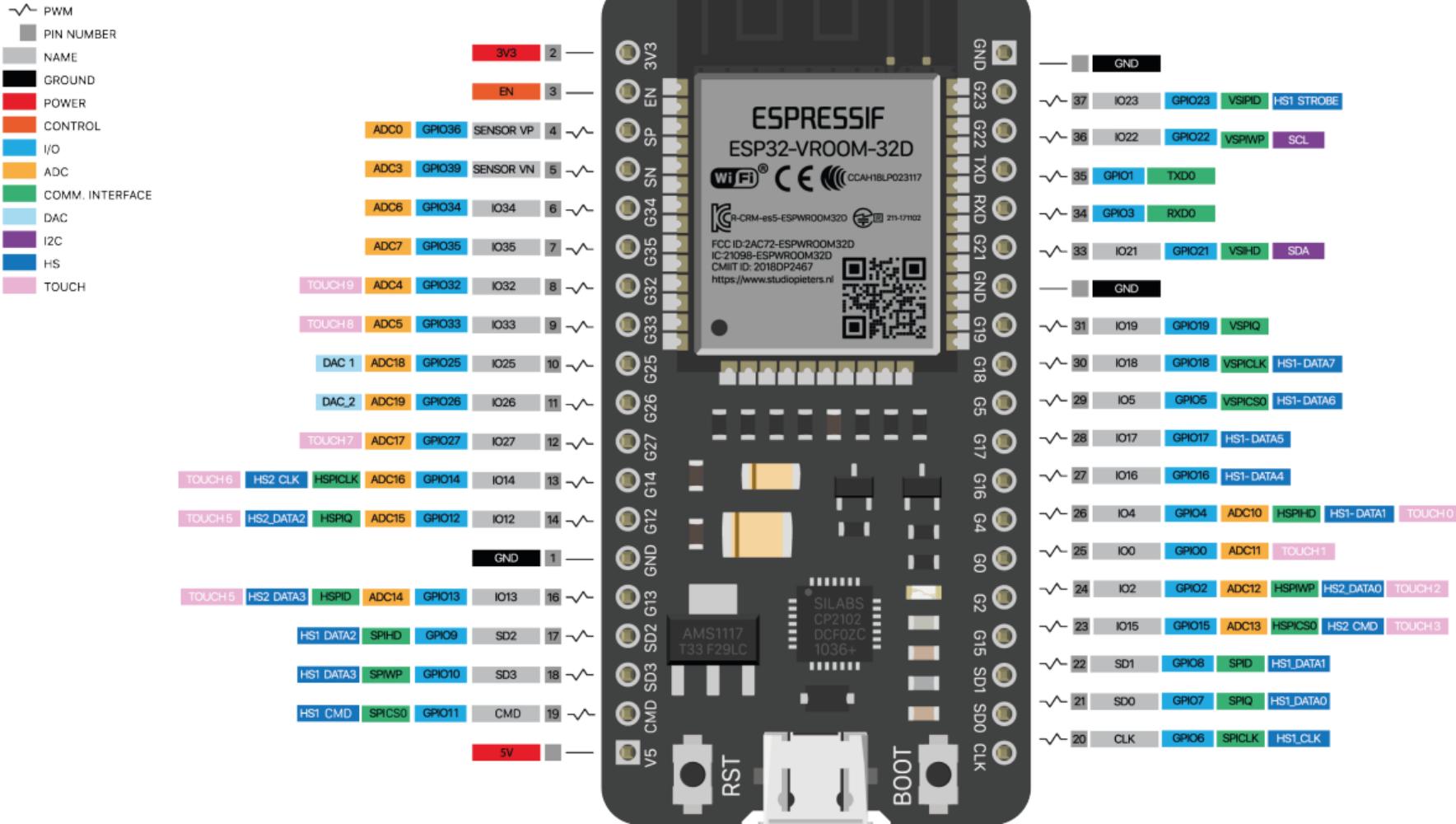


✓ PWM  
 PIN NUMBER  
 NAME  
 GROUND  
 POWER  
 CONTROL  
 I/O  
 ADC  
 COMM. INTERFACE  
 DAC  
 I2C  
 HS  
 TOUCH



# ESP32 Wroom DevKit Full Pinout





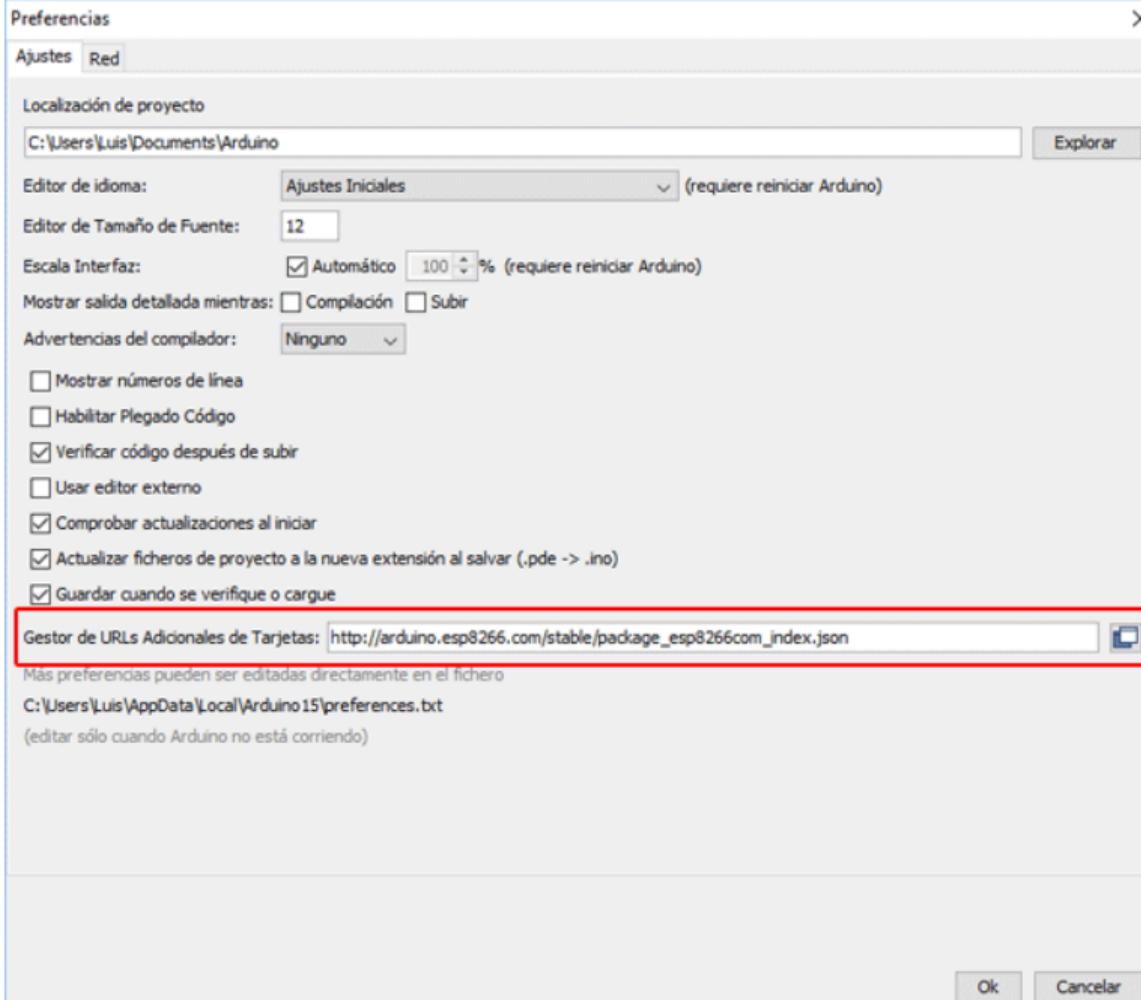
# Agregar tarjetas

## Archivo Editar Programa Herramientas Ayuda

- Nuevo Ctrl+N
- Abrir... Ctrl+O
- Abrir Reciente >
- Proyecto >
- Ejemplos >
- Cerrar Ctrl+W
- Salvar Ctrl+S
- Guardar Como... Ctrl+Mayús+S
- Configurar Página Ctrl+Mayús+P
- Imprimir Ctrl+P
- Preferencias Ctrl+Coma**
- Salir Ctrl+Q

run once:

run repeatedly:



# Instalación de la tarjeta ESP-8266

## Enlace de librería:

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

## Modos del ESP-01:

	GPIO0	GPIO2
<b>Modo UART</b> (carga programa)	LOW	HIGH (desconectado)
<b>Modo Flash</b> (ejecuta programa)	HIGH (desconectado)	HIGH (desconectado)

# Instalación de la tarjeta ESP-32

**Enlace de librería:**

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

Archivo Editar Programa Herramientas Ayuda

**Gestor de tarjetas...**

- Arduino SAMD (32-bits ARM Cortex-M0+) Boards
- Arduino/Genuino Zero (Programming Port)
- Arduino/Genuino Zero (Native USB Port)
- Arduino/Genuino MKR1000
- Arduino MKRZero
- Adafruit Circuit Playground Express
- Arduino M0 Pro (Programming Port)
- Arduino M0 Pro (Native USB Port)
- Arduino M0

## Placas Arduino AVR

- Arduino Yún
- Arduino/Genuino Uno
- Arduino Duemilanove or Diecimila
- Arduino Nano
- Arduino/Genuino Mega or Mega 2560
- Arduino Mega ADK
- Arduino Leonardo
- Arduino Leonardo ETH
- Arduino/Genuino Micro
- Arduino Esplora
- Arduino Mini
- Arduino Ethernet

Auto Formato

Ctrl+T

Archivo de programa.

Reparar codificación &amp; Recargar.

Monitor Serie

Ctrl+Mayús+M

Serial Plotter

Ctrl+Mayús+L

WiFi101 Firmware Updater

**Placa: "Arduino/Genuino Uno"**

Puerto

Obtén información de la placa

Programador: "AVRISP mkII"

Quemar Bootloader

Tipo Todos

**Arduino AVR Boards by Arduino versión 1.6.17 INSTALLED**

Tarjetas incluidas en éste paquete

Arduino Yún, Arduino/Genuino Uno, Arduino Uno WiFi, Arduino Diecimila, Arduino Nano, Arduino/Genuino Mega, Arduino MegaADK, Arduino Leonardo, Arduino Leonardo Ethernet, Arduino/Genuino Micro, Arduino Esplora, Arduino Mini, Arduino Ethernet, Arduino Fio, Arduino BT, Arduino LilyPadUSB, Arduino Lilypad, Arduino Pro, Arduino ATMegaNG, Arduino Robot Control, Arduino Robot Motor, Arduino Gemma, Adafruit Circuit Playground, Arduino Yún Mini, Arduino Industrial 101, Linino One.

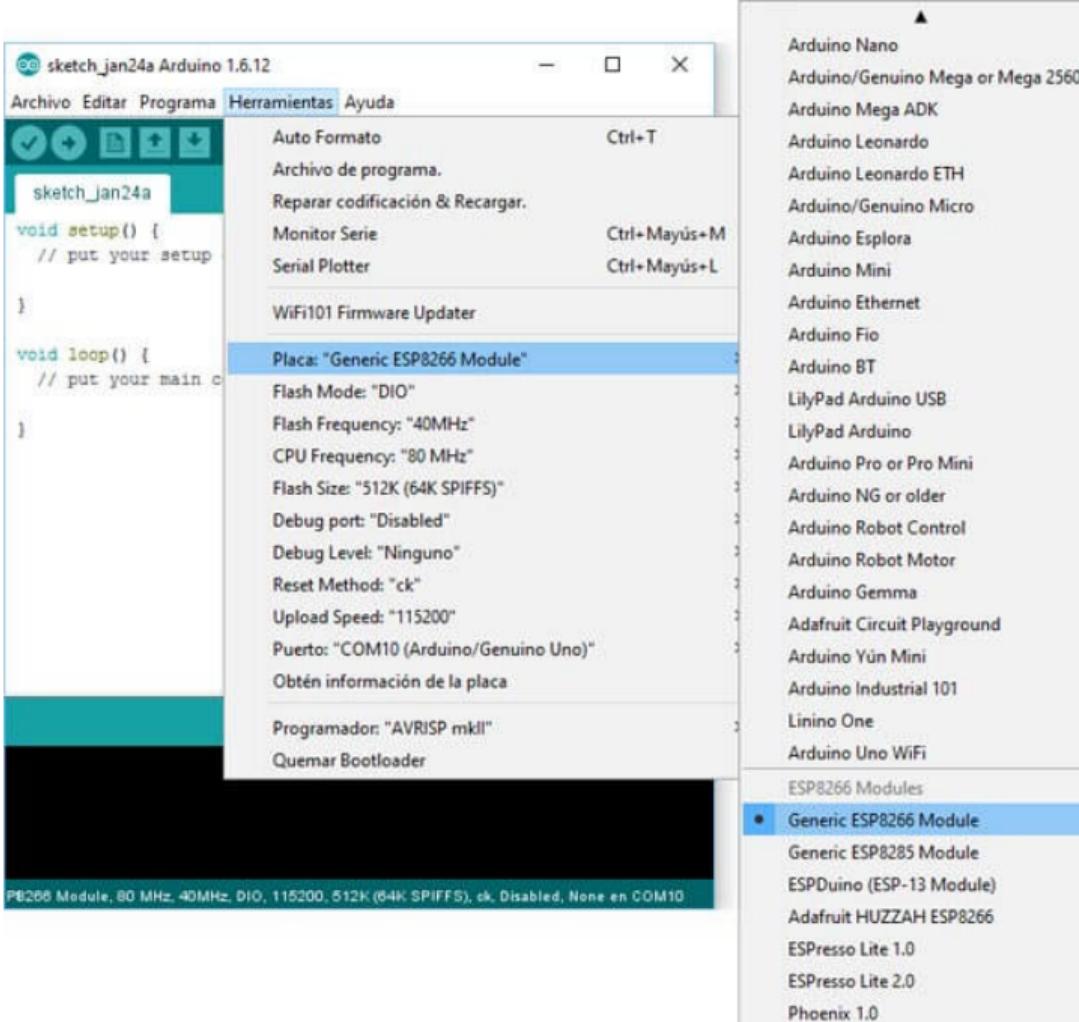
[Online help](#)[More info](#)**esp8266 by ESP8266 Community**

Tarjetas incluidas en éste paquete

Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), Espresso Lite 1.0, Espresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WiFiInfo, ESPDuino.

[Online help](#)[More info](#)

2.3.0



# Laboratorio 01: Hola mundo

# Laboratorio 01: Hola mundo

Para realizar el ejemplo de blink del LED interno en la ESP32 usando el IDE de Arduino, sigue estos pasos:

- 1 Abre el IDE de Arduino.
- 2 Haz clic en el menú **Archivo** y selecciona **Ejemplos**.
- 3 En la sección **Ejemplos**, selecciona **01.Basics** y luego **Blink**.
- 4 Verifica que en el código se está usando el número correcto del LED interno, que suele ser el **GPIO 2**. Si no es así, modifica la línea de la forma "const int led = 2;" para que refleje el número correcto.
- 5 Conecta tu ESP32 a tu computadora.
- 6 Selecciona la placa y el puerto correctos en el menú **Herramientas**.
- 7 Haz clic en el botón "Cargar" para cargar el código en la ESP32.
- 8 Verifica que el LED interno está parpadeando cada segundo.
- 9 Modifica la velocidad de 1000 ms a 500 ms

# ¡Muchas gracias por su atención!

*¿Preguntas?*



Contacto: Marco Teran  
webpage: [marcoteran.github.io/](https://marcoteran.github.io/)  
e-mail: [marco.teran@usa.edu.co](mailto:marco.teran@usa.edu.co)

