

# Introducción a la Ingeniería

---

## Introducción a Arduino y la placa ESP32

### Objetivo

- Familiarizarse con el ambiente de programación de Arduino
- Aprender a interconectar entradas y salidas a la placa ESP32

### Materiales

- Un controlador ESP32 y cable USB para conectarlo a la PC
- El IDE Arduino con la librería para ESP32
- Una tarjeta protoboard,
- Tres leds, cables y resistencias

### Introducción

Arduino es probablemente la plataforma de hardware abierto más popular para desarrollo y prototipado rápido de proyectos de sistemas digitales. Ha logrado conformar una amplísima comunidad de desarrolladores que comparten las especificaciones y el código de sus proyectos.

Las tarjetas Arduino tienen una serie de interfaces entrada y salida (E/S) analógicas y digitales que pueden ser programadas para monitorear (con ayuda de sensores) o interactuar (con ayuda de actuadores) con su entorno.

Hay una gran variedad de sensores (por ejemplo, de temperatura, humedad, iluminación, sonido, contaminantes, nutrientes, posicionamiento, velocidad, lector RFID, ...) y actuadores (por ejemplo, LEDs, zumbadores, relevadores, acopladores, pinzas, servo motores, ...) que pueden conectarse a una placa Arduino a través de sus interfaces programables E/S.

Arduino cuenta con un ambiente de desarrollo integrado (IDE, *Integrated Development Environment*) para programar las acciones que se desean configurar en la placa. Los programas (llamados *sketch*) escritos en el IDE se descargan a la placa a través de la interfaz USB.

La popularidad de Arduino es tan grande, que su ambiente de programación se ha vuelto un estándar de facto en la comunidad de hardware abierto. Es por ello que utilizaremos el IDE de Arduino en nuestro primer proyecto.

Si bien las tarjetas Arduino son sumamente populares, no son las únicas. En los últimos años han aparecido muchas otras placas para desarrollo de sistemas digitales por lo que hoy tenemos una amplia gama de opciones.

Entre ellas está el microcontrolador ESP32 de Espressif Systems, un dispositivo sumamente poderoso y versátil. Al igual que las placas Arduino, puede recibir y enviar datos analógicos y digitales, procesar datos e interactuar con usuarios a través de la computadora personal. Además, el módulo ESP32 tiene integrado un chip Bluetooth y uno WiFi lo que le permite conectarse a Internet.

En nuestro primer proyecto, el Robot que utilizará está controlado por una tarjeta Arduino y el lector RFID, que programará usted, está controlado por una tarjeta ESP32.

## Desarrollo

### 1. Instalación del IDE y de las librerías

El IDE permite expresar un programa (un *sketch*) en un lenguaje muy parecido a C que se compila en el lenguaje de máquina del micro-controlador de la placa. Descargue el IDE de la siguiente dirección: <https://www.arduino.cc/en/Main/Software>.

Para poder programar el módulo ESP32 desde el IDE de Arduino, es necesario configurar el IDE y cargar algunas librerías. Para ello, sigas las instrucciones en esta liga:

<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>. *NO SE DESEPERE. ESTO TOMARÁ ALGÚN TIEMPO.* (Nota: No es necesario que haga la validación buscando redes WiFi).

### 2. Verificación

Como se señala en la guía anterior, una vez configurado el IDE, deberá seleccionar la placa ESP32 y el puerto COM en la que está conectada. La placa que se le entregó para este proyecto es **DOIT ESP DEVKIT V1**.

1. Conecte el módulo a su PC a través del cable USB. Se debió haber encendido un led rojo indicando que la placa está energizada.
2. En el IDE de Arduino, en el menú **Herramientas**, seleccione la placa **ESP32 Arduino/DOIT ESP DEVKIT V1**
3. En el mismo menú seleccione el puerto al que está conectada la placa
4. En el menú **Archivo/Ejemplos/** seleccione el sketch **01.Basics/Blink**
5. Compílelo dando clic en el ícono con una paloma. Si no tiene errores, cárguelo a la placa dando clic en la flecha a la derecha:

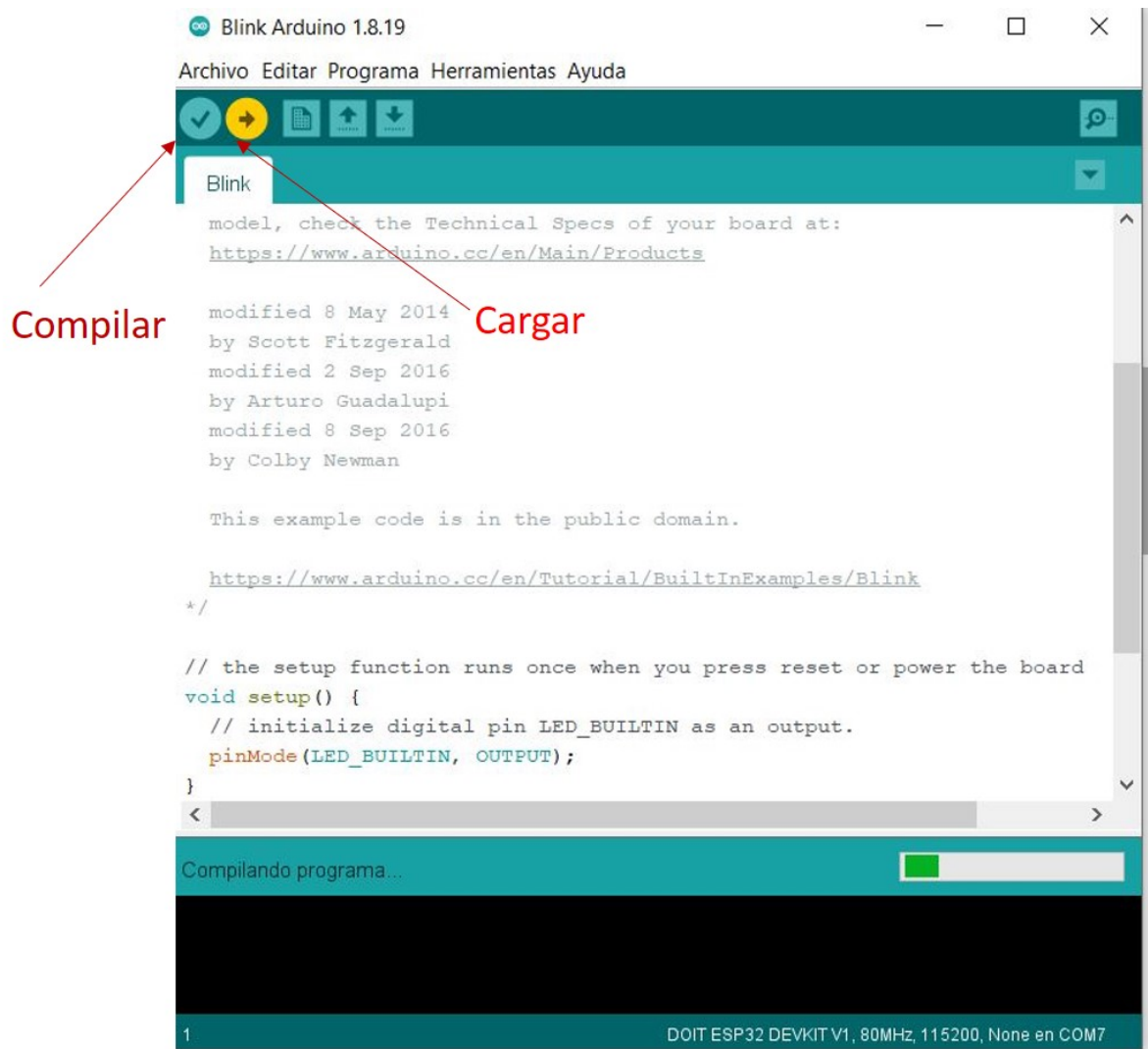


Figura 1. Compilar y cargar un sketch en el IDE de Arduino

6. Si todo está correcto, podrá ver un led azul parpadeando

---

**EJERCICIO 1.** ( 1 Punto). Muestre el led parpadeando al profesor

---

7. Ahora modifique el sketch para que el led se mantenga encendido un segundo y apagado medio segundo.

---

**EJERCICIO 2.** (2 Puntos). Muestre al profesor el led parpadeando con las modificaciones solicitadas

---

---

**EJERCICIO 3.** (1 Punto). Analice y describa brevemente qué hace el código.

---

### 3. Conectar y manipular un led externo

En esta sección nos vamos a familiarizar con la conexión de componentes electrónicos en una tarjeta protoboard, (también conocidas como Breadboard) como la mostrada en la Figura 2:

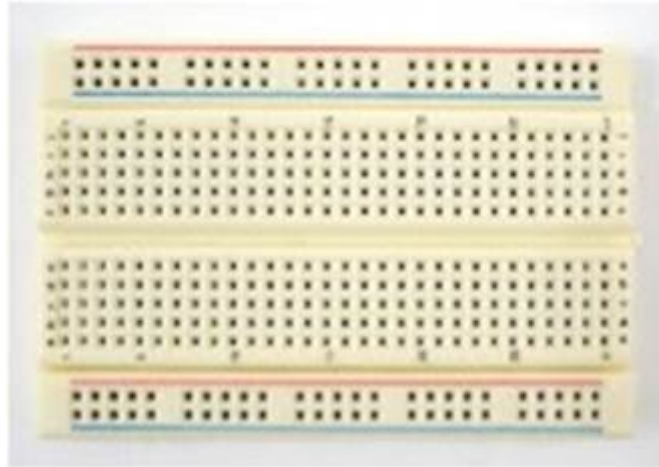


Figura 2. Tarjeta Protoboard

Las tarjetas protoboard (placa de pruebas) permiten crear rápidamente prototipos de circuitos electrónicos. Son tablas con orificios que se encuentran interconectados entre sí internamente. En la Figura 2, los orificios paralelos a las líneas rojas y azules están interconectados y se usan típicamente para transportar señales de voltaje y de tierra. Los orificios verticales (las columnas) entre las muescas de la tarjeta (por ejemplo, los orificios 12A a 12E), también están interconectados entre sí.

Vamos a colocar un LED en la tarjeta protoboard que encenderemos y apagaremos desde el ESP32. Para ello, el "cátodo" del LED, que se reconoce por ser la pata más corta y tener una muesca plana en el plástico que lo recubre, se conecta a tierra. El "anodo" se conecta a una pata digital de Entrada/Salida (E/S) de la placa ESP32 a través de un resistor de 1kOhm (en realidad, el valor no importa demasiado; resistencias más grandes, disminuirán la corriente y con ello, la intensidad de luminosidad del led).

Existen muchos modelos ESP32. El que se le entregó tiene 30 pines. La distribución de pines en ese modelo es la que se muestra en la figura 3:

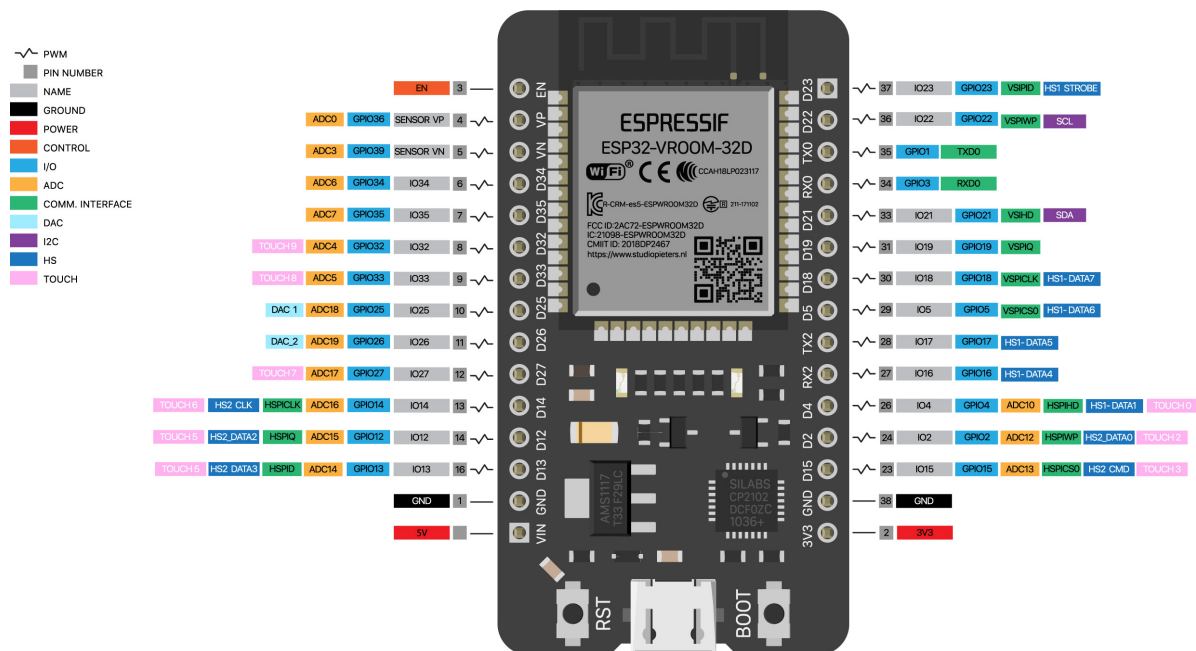


Figura 3: Distribución de pines ESP32 (30 pines)

Utilizaremos la E/S GPIO22 (*General I/O Port*) que corresponde a la pata 29 de nuestra placa como se observa en la figura 4:

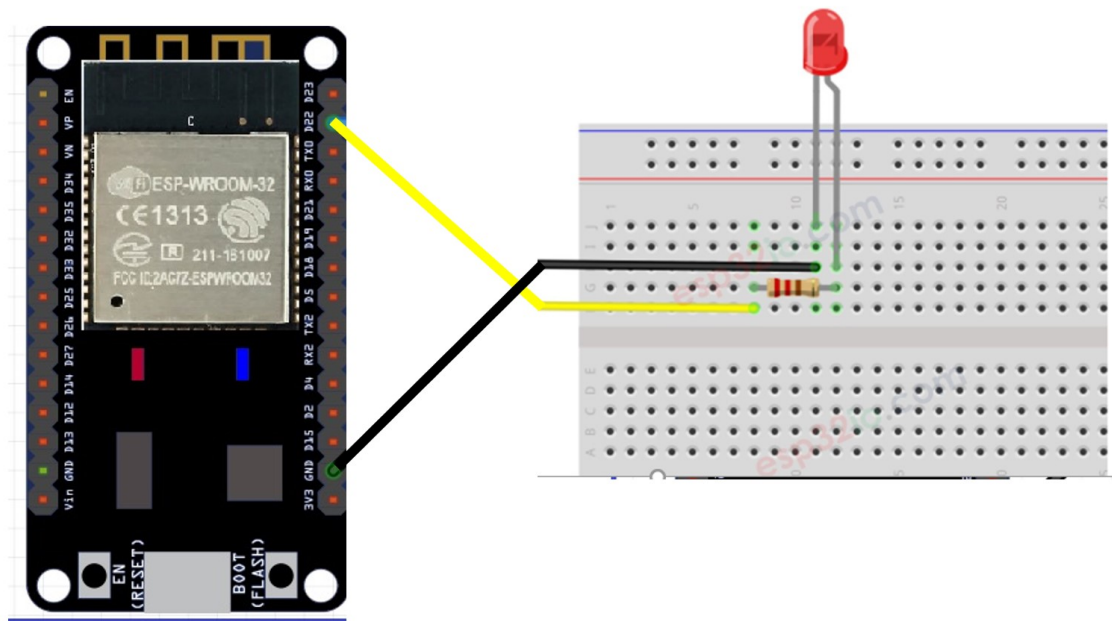


Figura 4. Diagrama de conexión un LED

**Ejercicio 4.** (1 Punto) Antes de conectar la placa, muestre el circuito armado al profesor.

Guarde el código del *sketch* de parpadeo con un nuevo nombre, por ejemplo, "LedExterno". Vamos a modificar el código para que cuando el LED de la placa esté encendido, el LED en el protoboard se apague y viceversa. Los segmentos de código que necesitará son los siguientes:

```
// Antes de setup
#define LED_PROTOBOARD 22 // GPIO que se utilizará para iluminar LED externo

// En setup
pinMode(LED_PROTOBOARD, OUTPUT); // GPIO22 como salida digital

// En loop
...
digitalWrite(LED_PROTOBOARD, HIGH);
...
digitalWrite(LED_PROTOBOARD, LOW);
...
```

---

**EJERCICIO5.** (2 Puntos). Modifique el código, cárguelo a la placa ESP32 y muestre el circuito funcionando al profesor.

---

---

**EJERCICIO 6.** (3 Puntos). Muy bien. Ahora deberá conectar dos LEDs adicionales en los pines de E/S que usted desee. No se olvide de conectar los LEDs con resistores a los pines. Escriba un *sketch* para que los tres LEDS se alternen en el encendido: LED1 un segundo, LED2 un segundo, LED3 un segundo.

Muestre el sistema funcionando a su profesor

---