

# FUNCIONES (E/S DIGITALES Y E/S ANALÓGICA).



## E/S DIGITALES

### Lectura digital()



#### Descripción

Lee el valor de un pin digital especificado, ya sea HIGHo LOW.

#### Sintaxis

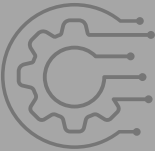
digitalRead(pin)

#### Parámetros

pin: el número de pin de Arduino que desea leer.

#### Devoluciones

HIGHoLOW



### CÓDIGO DE EJEMPLO

Establece el pin 13 al mismo valor que el pin 7, declarado como entrada.

```
int ledPin = 13; // LED conectado al pin digital 13
int inPin = 7; // Pulsador conectado al pin digital 7
int val = 0; // Variable para almacenar el valor leído

void setup() {
  pinMode(ledPin, OUTPUT); // Establece el pin digital 13 como salida
  pinMode(inPin, INPUT); // Establece el pin digital 7 como entrada
}

void loop() {
  val = digitalRead(inPin); // Leer el pin de entrada
  digitalWrite(ledPin, val); // Establece el LED al valor del botón
```

### Escritura digital()



#### Descripción

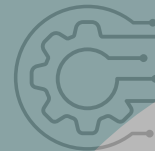
Escriba un valor HIGHo un LOWvalor en un pin digital.

#### Sintaxis

digitalWrite(pin, value)

#### Parámetros

pin: el número de pin de Arduino.  
value: HIGHo LOW.



### CÓDIGO DE EJEMPLO

El código convierte el pin digital 13 en an OUTPUTy lo cambia alternando entre HIGH y LOWa un ritmo de un segundo.

```
void setup() {
  pinMode(13, OUTPUT); // Establece el pin digital 13 como salida
}

void loop() {
  digitalWrite(13, HIGH); // Activa el pin digital 13
  delay(1000); // Espera un segundo
  digitalWrite(13, LOW); // Desactiva el pin digital 13
  delay(1000); // Espera un segundo
}
```

### PinMode()



#### Descripción

Configura el pin especificado para que se comporte como una entrada o una salida. Consulte la página Pines digitales para obtener detalles sobre la funcionalidad de los pines.

#### Sintaxis

pinMode(pin, mode)

#### Parámetros

mode: INPUT, OUTPUT, o INPUT\_PULLUP. Consulte la página Pines digitales para obtener una descripción más completa de la funcionalidad.



### CÓDIGO DE EJEMPLO

El código hace que el pin digital sea 13 OUTPUTy lo alterna HIGH y LOW

```
void setup() {
  pinMode(13, OUTPUT); // Establece el pin digital 13 como salida
}

void loop() {
  digitalWrite(13, HIGH); // Activa el pin digital 13
  delay(1000); // Espera un segundo
  digitalWrite(13, LOW); // Desactiva el pin digital 13
  delay(1000); // Espera un segundo
}
```

# FUNCIONES (E/S DIGITALES Y E/S ANALÓGICA).



## E/S ANALÓGICA

### Lectura analógica ( )

#### Descripción

Lee el valor del pin analógico especificado. Las placas Arduino contienen un convertidor analógico a digital multicanal de 10 bits.

#### Sintaxis

analogRead(pin)

#### Parámetros

pin: el nombre del pin de entrada analógica para leer (A0 a A5 en la mayoría de las placas, A0 a A6 en las placas MKR, A0 a A7 en Mini y Nano, A0 a A15 en Mega).

#### Devoluciones

La lectura analógica en el pin. Aunque está limitado a la resolución del conversor analógico a digital (0-1023 para 10 bits o 0-4095 para 12 bits). Tipo de datos: int.

### Referencia analógica()

#### Descripción

Configura el voltaje de referencia utilizado para la entrada analógica (es decir, el valor utilizado como la parte superior del rango de entrada).

#### Sintaxis

analogReference(type)

#### Parámetros

type: qué tipo de referencia usar (ver lista de opciones en la descripción).

### Escritura analógica ( )

#### Descripción

Escribe un valor analógico ( onda PWM ) en un pin. Se puede usar para encender un LED con diferentes brillos o impulsar un motor a varias velocidades.

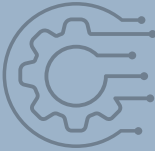
#### Sintaxis

analogWrite(pin, value)

#### Parámetros

pin: el pin de Arduino para escribir. Tipos de datos permitidos: int.

value: el ciclo de trabajo: entre 0 (siempre apagado) y 255 (siempre encendido). Tipos de datos permitidos: int.



### CÓDIGO DE EJEMPLO

El código lee el voltaje en analogPin y lo muestra.

```
int analogPin = A3; // Limpiaparabrisas de potenciómetro (terminal central) conectado al pin analógico 3
// Cables externos a tierra y +5V
int val = 0; // Variable para almacenar el valor leído

void setup() {
  Serial.begin(9600); // Serie de configuración
}

void loop() {
  val = analogRead(analogPin); // Leer el pin de entrada
  Serial.println(val); // Valor de depuración
}
```



### PLACA ARDUINOS

Placas Arduino AVR (Uno, Mega, Leonardo, etc.)

- DEFAULT:** La referencia analógica predeterminada de 5 voltios (en placas Arduino de 5 V) o 3,3 voltios (en placas Arduino de 3,3 V).
- INTERNO:** Una referencia incorporada, igual a 1,1 voltios en ATmega168 o ATmega328P y 2,56 voltios en ATmega32U4 y ATmega8 (no disponible en Arduino Mega).
- INTERNAL1V1:** Una referencia de 1.1V incorporada (solo Arduino Mega).
- INTERNAL2V56:** Una referencia integrada de 2,56 V (solo Arduino Mega).
- EXTERNO:** El voltaje aplicado al pin AREF (solo 0 a 5V) se usa como referencia.



### CÓDIGO DE EJEMPLO

Establece la salida al LED proporcional al valor leído del potenciómetro.

```
int ledPin = 9; // LED conectado al pin digital 9
int analogPin = 3; // Potenciómetro conectado al pin analógico 3
int val = 0; // Variable para almacenar el valor leído

void setup() {
  pinMode(ledPin, OUTPUT); // Establece el pin como salida
}

void loop() {
  val = analogRead(analogPin); // Leer el pin de entrada
  analogWrite(ledPin, val / 4); // Los valores de lectura analógica van de 0 a 1023, los valores de escritura analógica de 0 a 255
}
```