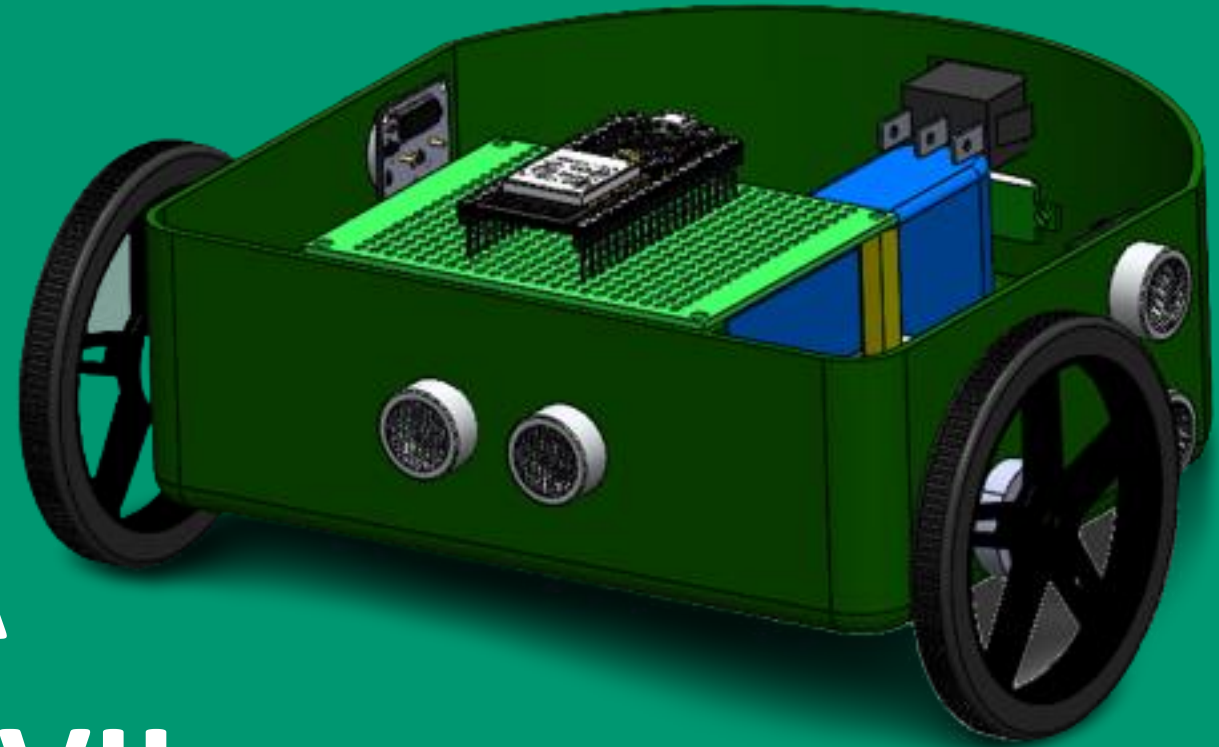




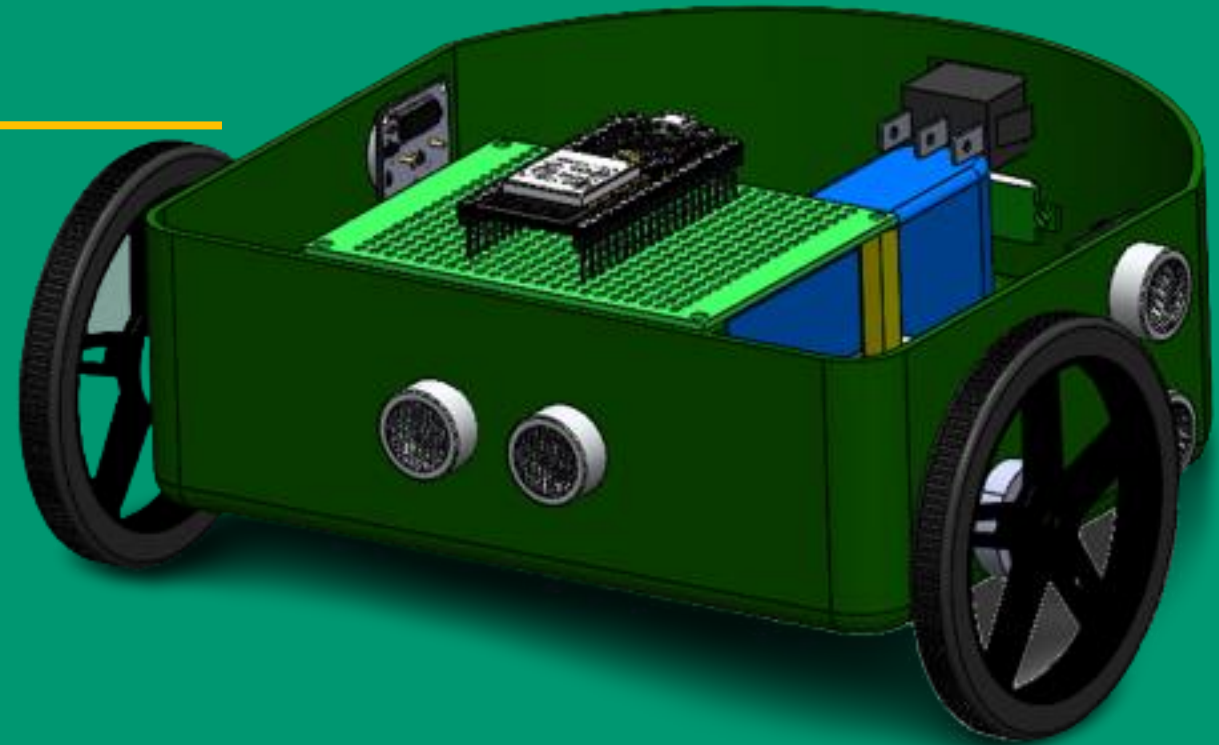
# INTRODUCCIÓN A LA ROBÓTICA MÓVIL

---

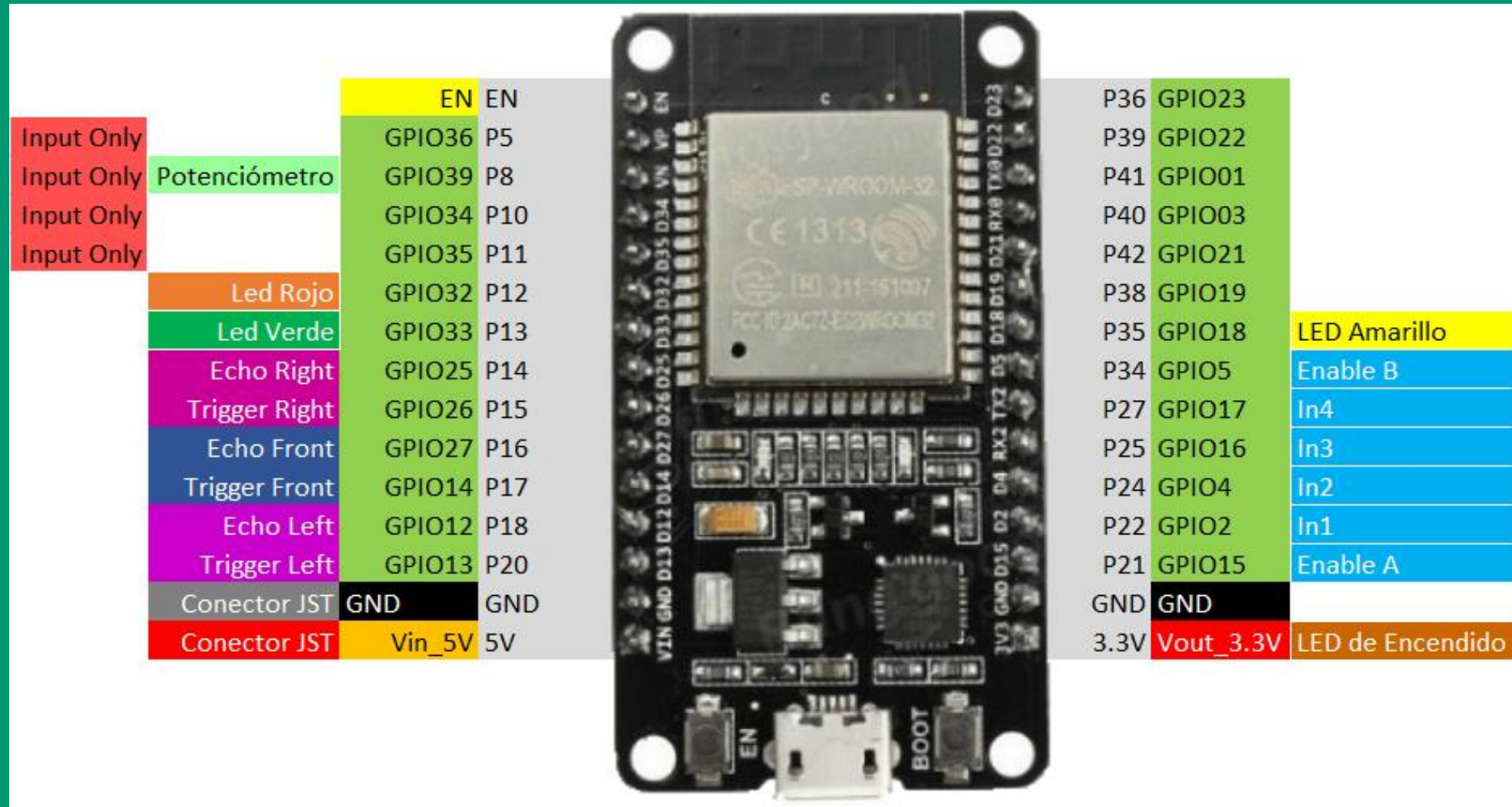


M.I. Sergio Hernández Sánchez  
sergio.hernandez.sanchez@itam.mx

# Prueba de PCB



# ESP32 – Esquema de conexiones



# Blink 3 LED's

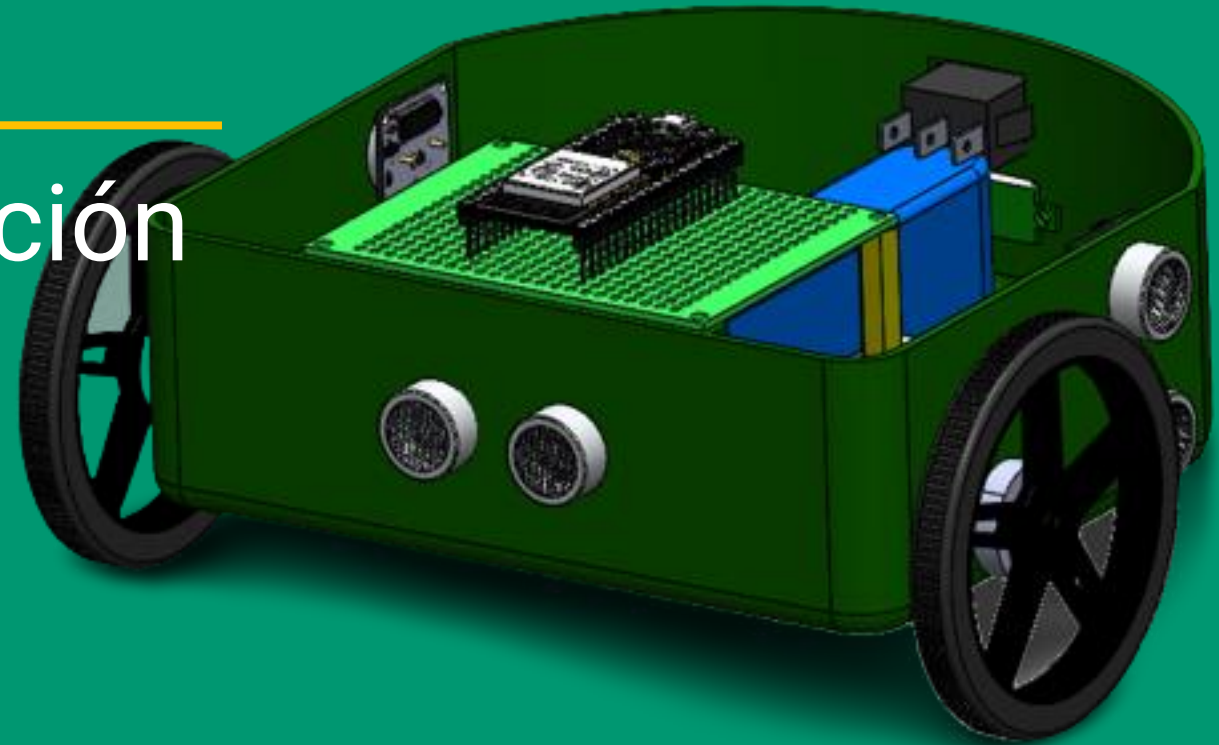
```
#define LED_L 33
#define LED_F 32
#define LED_R 18

void setup() {
    pinMode(LED_L, OUTPUT);
    pinMode(LED_F, OUTPUT);
    pinMode(LED_R, OUTPUT);
}

void loop() {
    digitalWrite(LED_L, HIGH);
    digitalWrite(LED_F, HIGH);
    digitalWrite(LED_R, HIGH);
    delay(1000);
    digitalWrite(LED_L, LOW);
    digitalWrite(LED_F, LOW);
    digitalWrite(LED_R, LOW);
    delay(1000);
}
```

Ejemplo - 4-Prueba\_LEDs

# Sensores para navegación



# Primitivas de la robótica

---



Sensar

Obtener información  
del ambiente para  
navegar en un entorno  
determinado

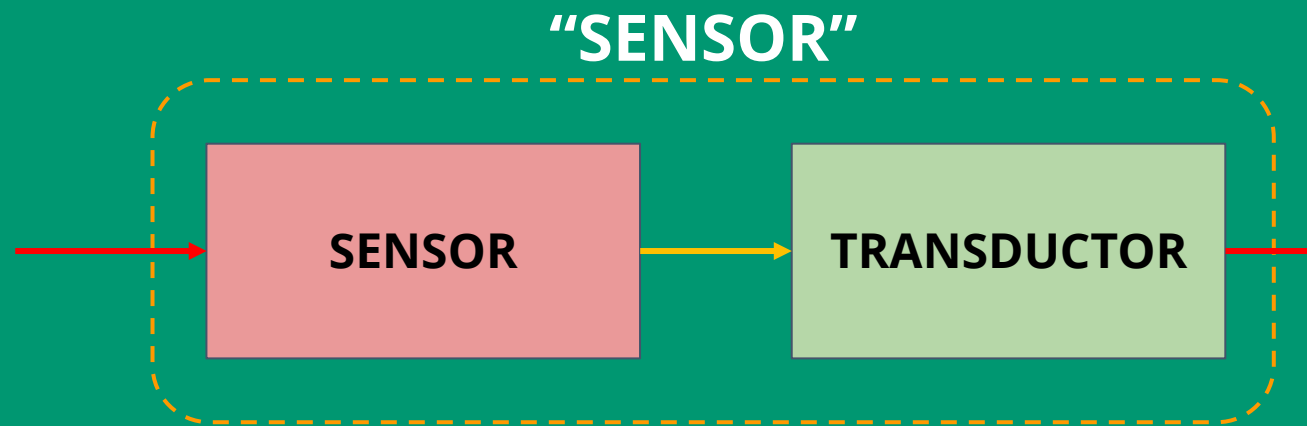


# Sensores

Un sensor es un objeto capaz de variar una propiedad ante magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas con un transductor en variables eléctricas.

## Variables de Instrumentación

- Intensidad luminosa
- Temperatura
- Distancia
- Fuerza
- Movimiento



## Variables eléctricas

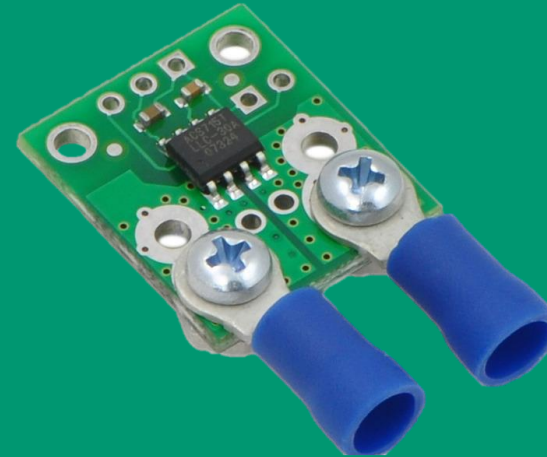
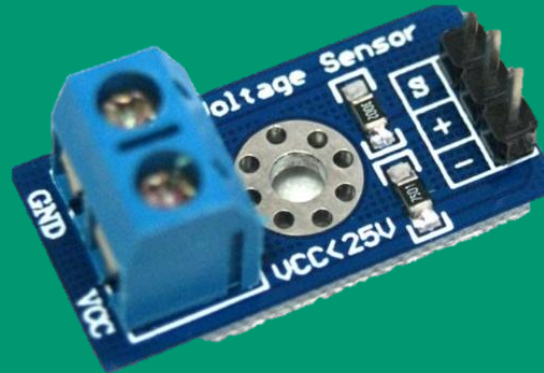
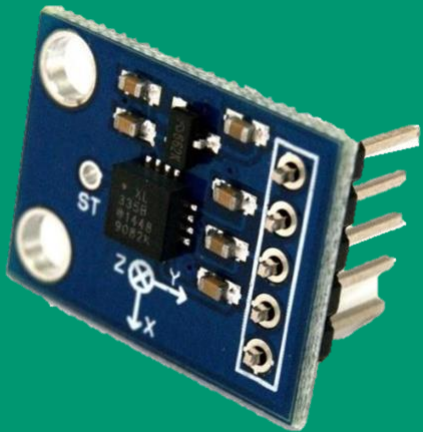
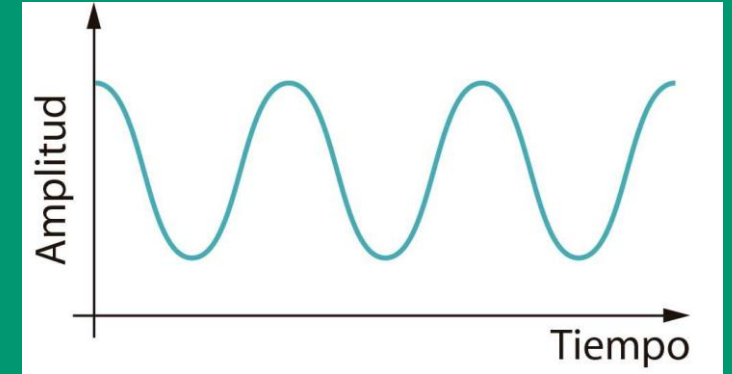
- Resistencia eléctrica
- Capacitancia
- Diferencia de potencial
- Corriente eléctrica

# Sensor de señal Analógica

Variable de  
Instrumentación

**SENSOR**

Variable  
eléctrica,  
SEÑAL  
ANALÓGICA

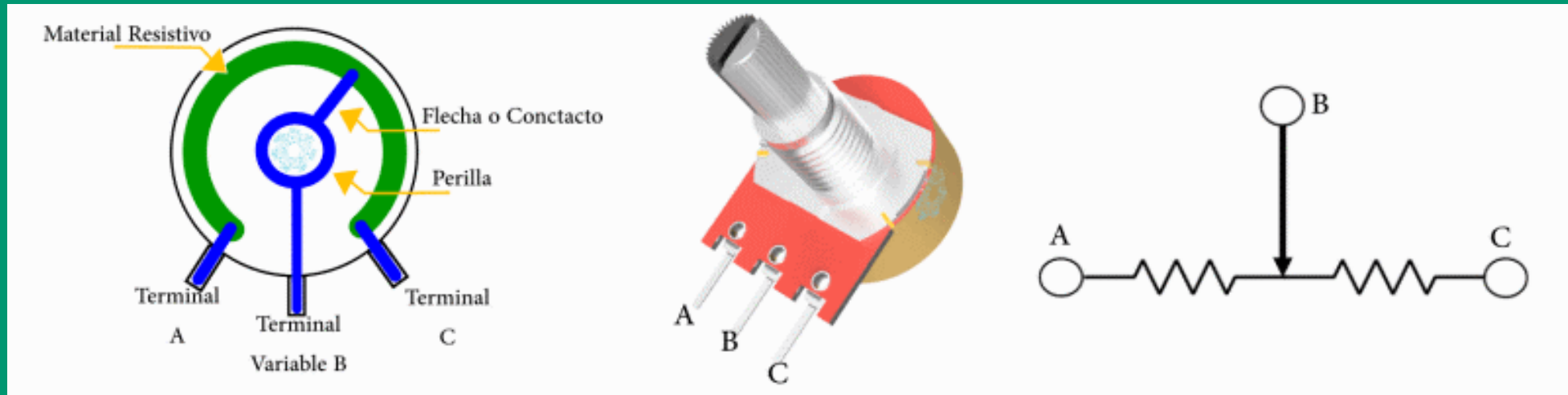




# Entradas analógicas (ADC)

- Se pueden usar 18 canales
  - ADC1 con 8 canales (GPIO 32 - 39)
  - ADC2 con 10 canales (GPIO 0, 2, 4, 12 - 15 y 25 - 27)
- Si se usa WiFi, no se puede utilizar ADC2
- Resolución 12 bits (0 - 4095)

```
analogRead(pin);
```



# Entradas analógicas (ADC)

Ejemplo - 5-Prueba\_potenciometro

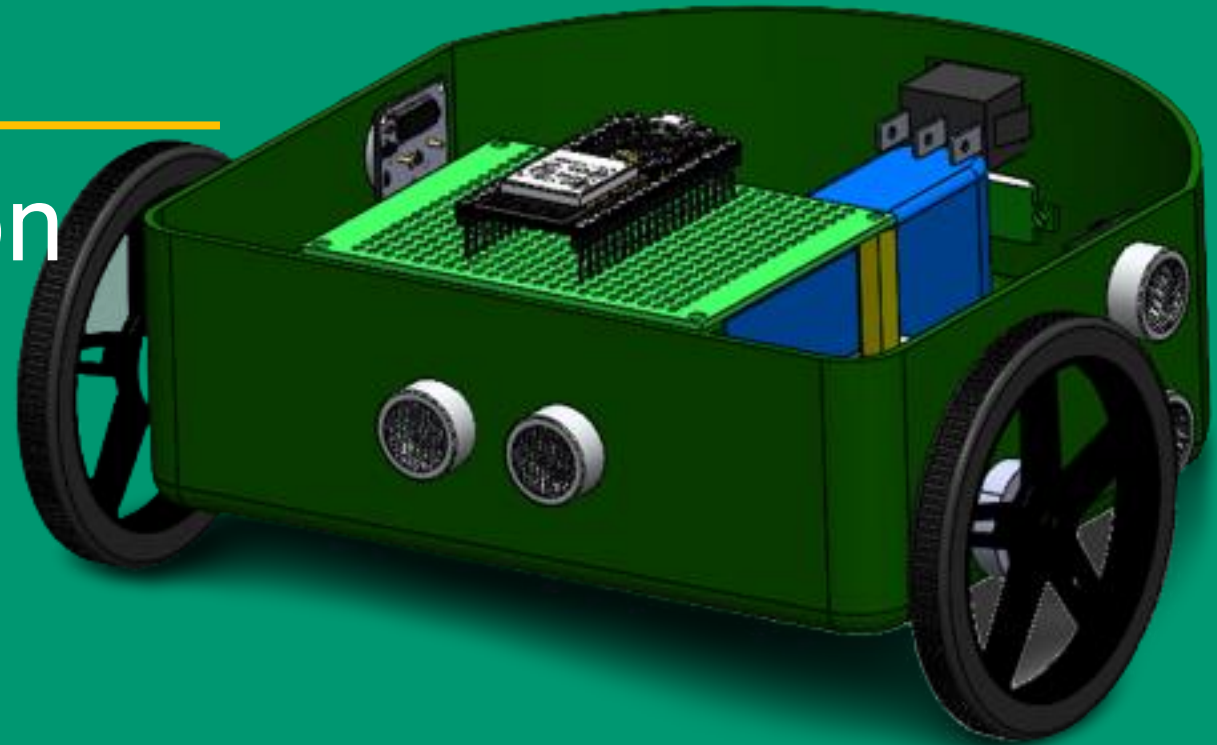
```
#define pot 39

int val;

void setup() {
    Serial.begin(115200);
}

void loop() {
    val = analogRead(pot);
    Serial.print("Valor: ");
    Serial.println(val);
}
```

# Pulse Width Modulation



# PWM

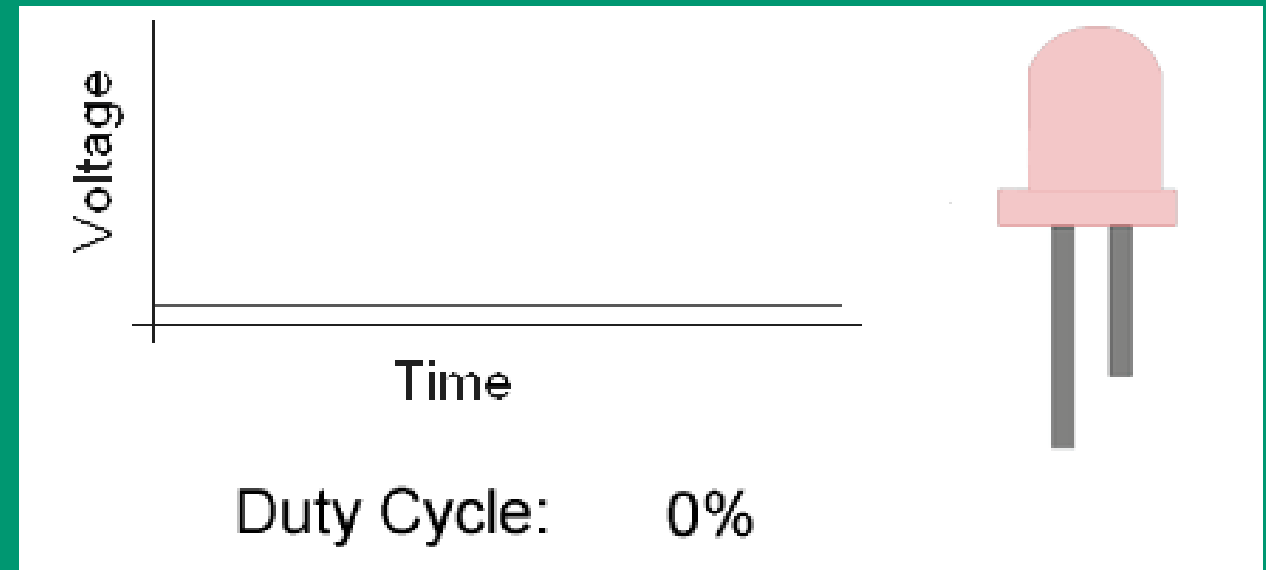
- Todos los GPIO pueden utilizarse.
- Se tienen máximo 16 canales.
- Se puede modificar la frecuencia y la resolución.

```
ledcAttachPin(pin, canal);
```

```
ledcSetup(canal, frecuencia, bits de resolución);
```

- Canal: 0-15
- Frecuencia hasta 12 kHz
- Resolución hasta 12 bits

```
ledcWrite(canal, brillo);
```



# PWM - LEDs

```
#define LED 18

int brillo = 4095;
int canal = 0;
int frec = 1000;
int res = 12;

void setup() {
    //Asignamos un canal al pin del LED
    //Hay 16 canales, del 0 al 15
    ledcAttachPin(LED, canal);

    // Configuramos los canales
    // ledcSetup(canal, frecuencia, bits_de_resolución);
    // canales 0-15, 12 kHz, resolución 1-16 bits,
    //frecuencia límite dependerá de la resolución
    ledcSetup(canal, frec, res);
}
```

## Ejemplo - 6-Brillo-LEDs

```
void loop() {
    ledcWrite(0, 0);
    delay(1000);
    ledcWrite(0, brillo/4);
    delay(1000);
    ledcWrite(0, brillo/2);
    delay(1000);
    ledcWrite(0, brillo);
    delay(1000);
}
```

# Ejercicio – Luminosidad controlada por pot

Ejemplo - 7-Brillo-pot

```
#define LED 18
#define pot 39

int brillo = 4095;
int canal = 0;
int frec = 1000;
int res = 12;
int val;

void setup() {
    Serial.begin(115200);

    ledcAttachPin(LED, canal);
    ledcSetup(canal, frec, res);
}

void loop() {
    val = analogRead(pot);
    Serial.print("Valor: ");
    Serial.println(val);
    ledcWrite(0, val);
}
```



# ¿Qué información queremos obtener?

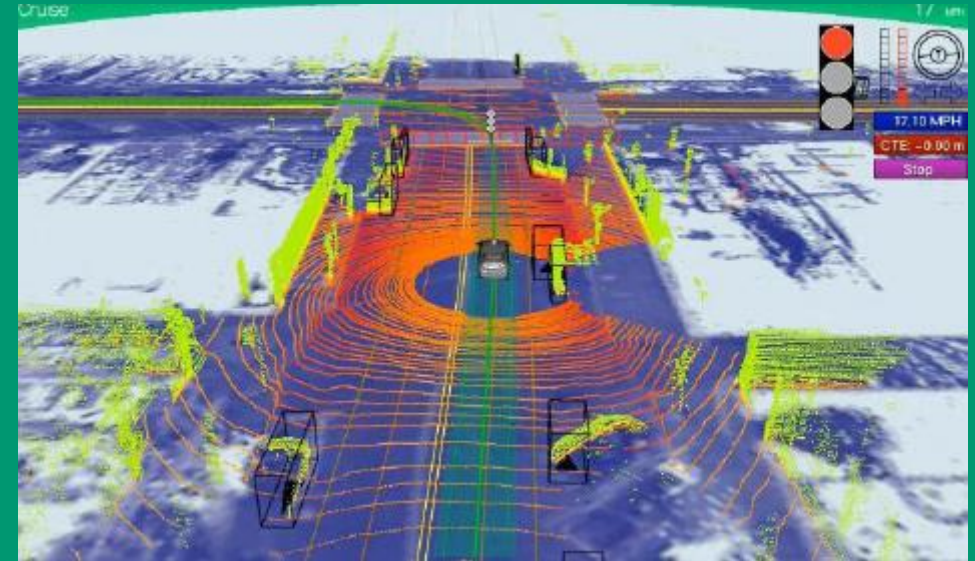
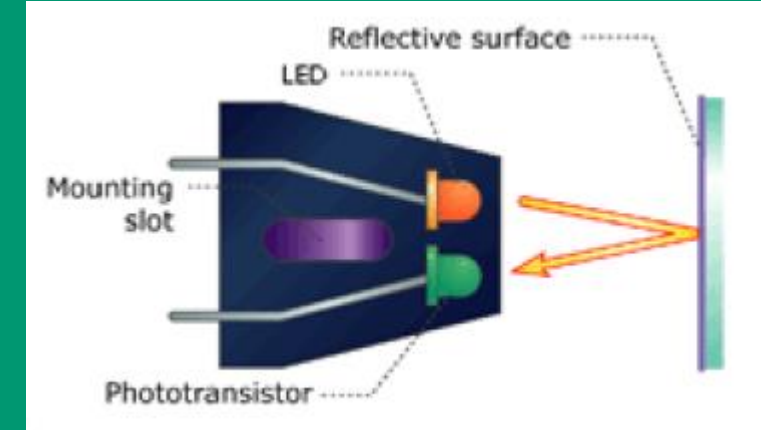
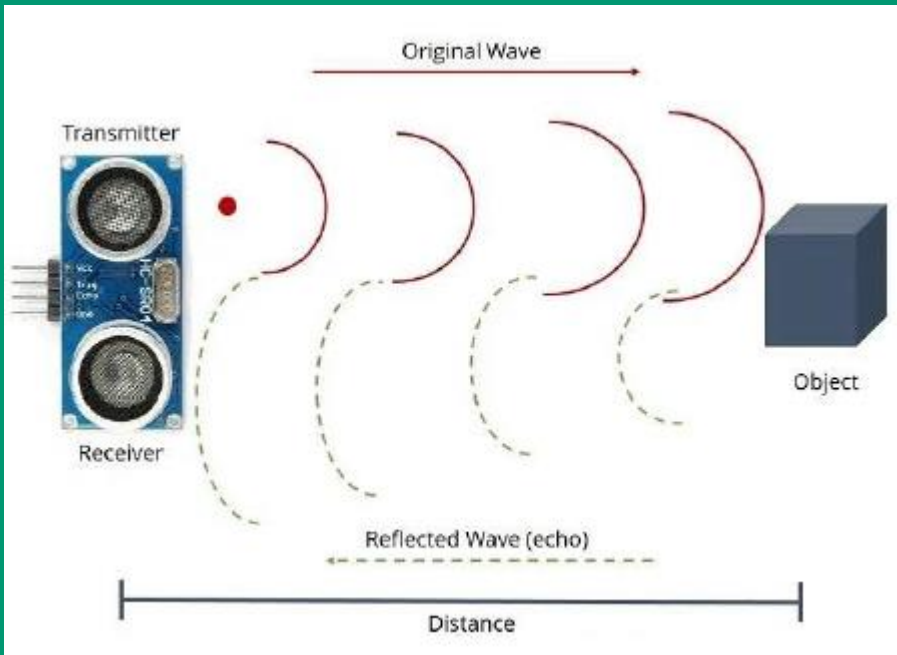
---

- Existencia de obstáculos
- Distancia a un objeto
- Temperatura del entorno
- Localización global
- Medir movimiento
- Medir inclinación respecto al piso
- Existencia de algún gas
- Cantidad de humedad del entorno
- Cantidad de luminosidad
- Alguna propiedad eléctrica
- Alguna propiedad magnética

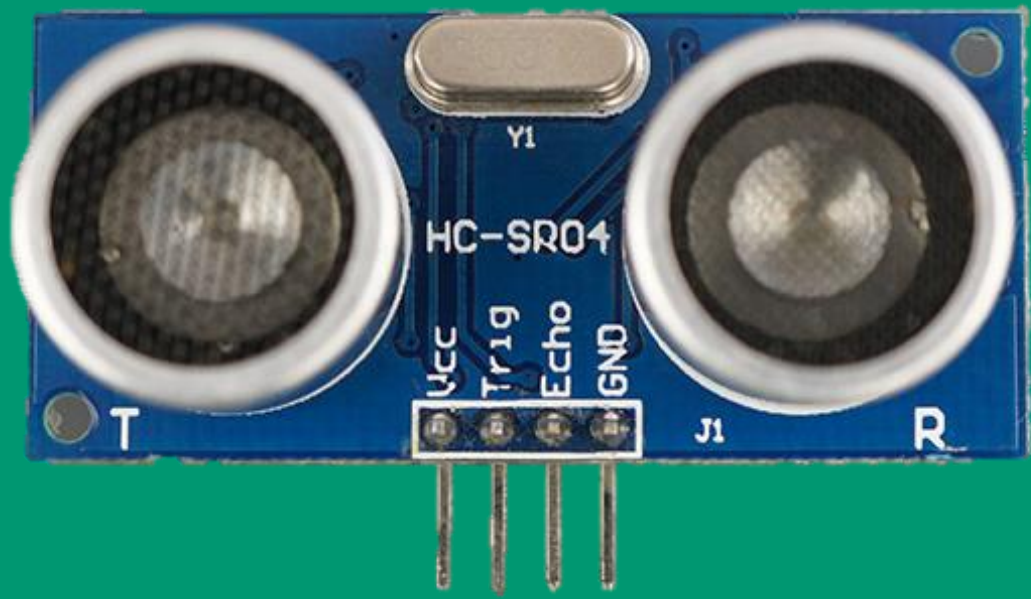


# ¿Qué información queremos obtener?

Existencia de obstáculos  
Distancia a un objeto

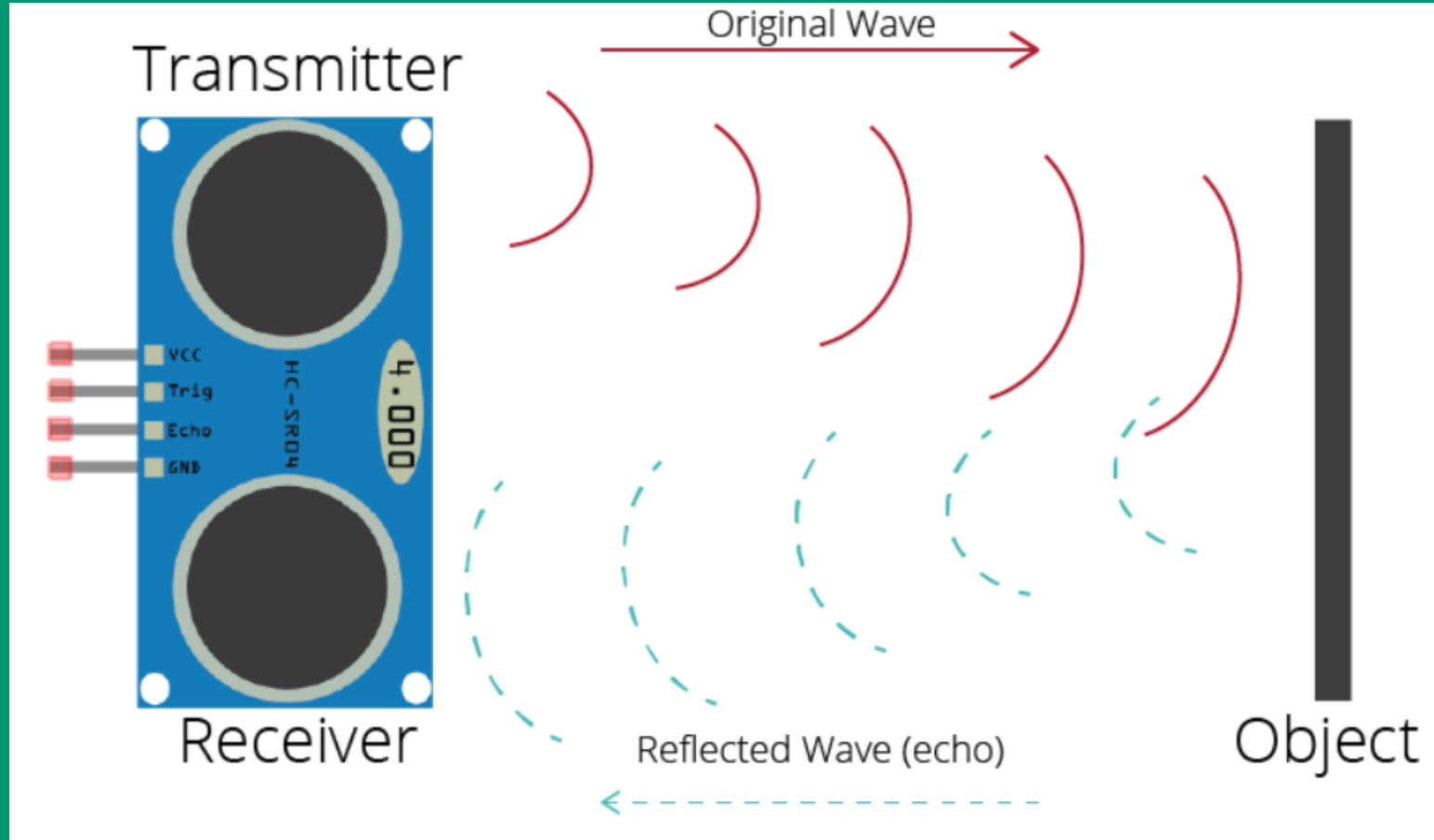


# Sensor ultrasónico HC-SR04



<b>Voltaje de operación</b>	<b>5V DC</b>
<b>Corriente de operación</b>	15 mA
<b>Frecuencia de trabajo</b>	40 kHz
<b>Rango máximo</b>	4 meters
<b>Rango mínimo</b>	2 cm
<b>Ángulo medido</b>	15°
<b>Resolución</b>	0.3 cm

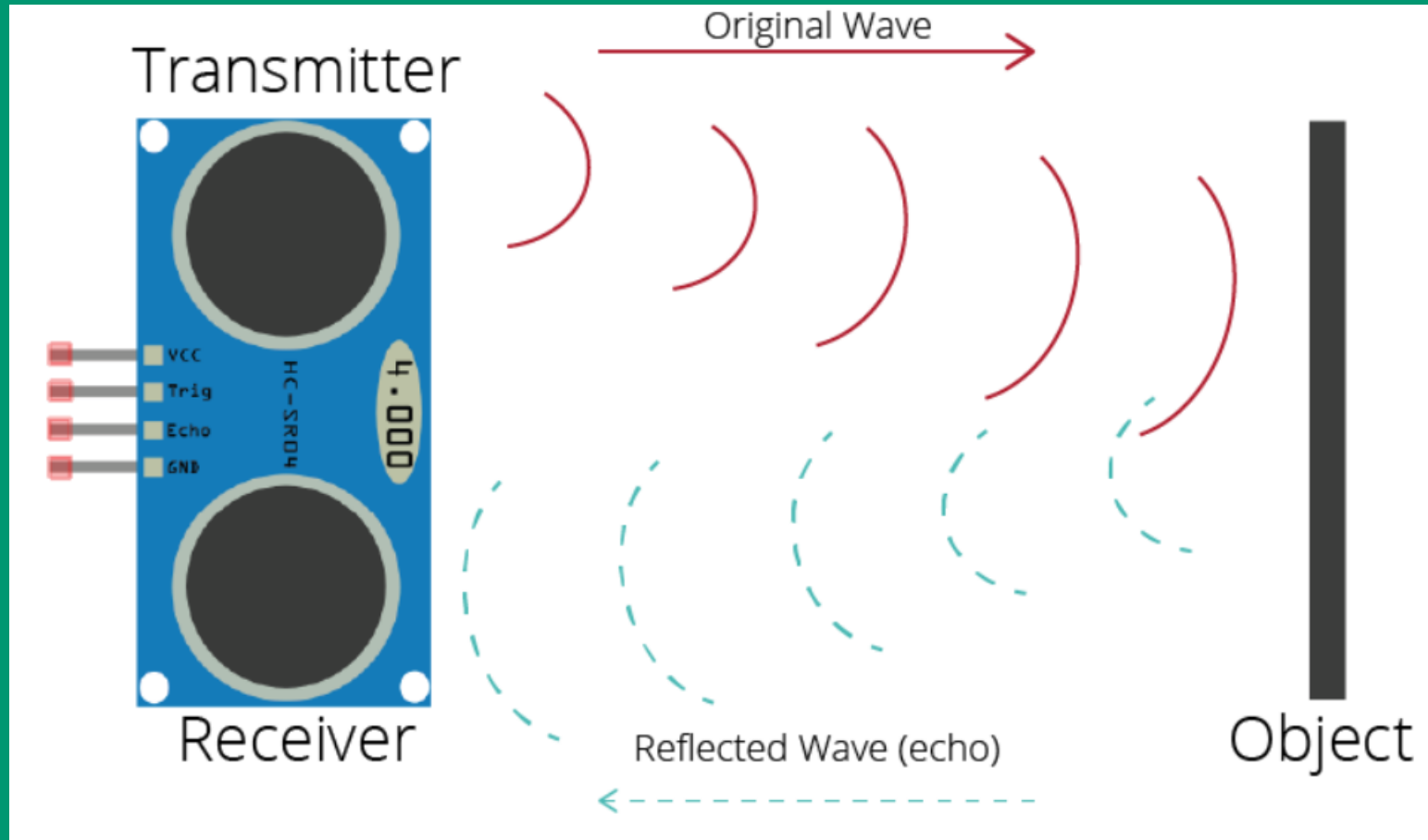
# Sensor ultrasónico HC-SR04



$$\text{Velocidad} = \frac{\text{Distancia}}{\text{tiempo}}$$

$$\text{Distancia al objeto} = \frac{\text{Velocidad del sonido en el aire} * \text{tiempo}}{2}$$

# Sensor ultrasónico HC-SR04



$$\text{Velocidad} = \frac{\text{Distancia}}{\text{tiempo}}$$

$$\text{Distancia al objeto} = \frac{343 \text{ m/s} * \text{tiempo}}{2} = \frac{0.0343 \text{ cm/us} * \text{tiempo}}{2}$$

# Sensor ultrasónico HC-SR04

Ejemplo - 8-Ultrasónico

```
#define echoPin 27
#define trigPin 14

// Variables
long duration;           // Variable para guardar la duración del viaje
                          // de la onda de sonido
int distance;            // Variable para la medición de la distancia

void setup() {
  pinMode(trigPin, OUTPUT); // Se define el pin trigger como SALIDA
  pinMode(echoPin, INPUT);  // Se define el pin echo como entrada

  Serial.begin(115200);     // Se inicializa la comunicación Serial del la
                          // ESP32 a 115,200 baudios
}
```

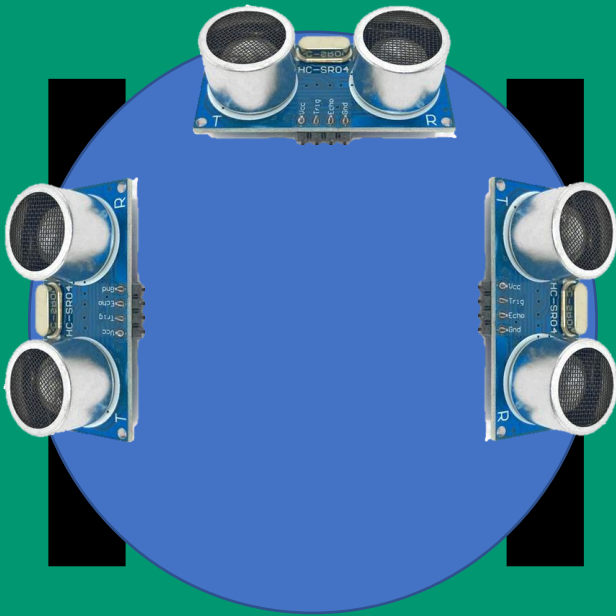


# Sensor ultrasónico HC-SR04

```
void loop() {  
  // Se pone en bajo el pin de trigger de inicio  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  
  // Se pone en alto el pin de Trigger por 10 microsegundos  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  
  // Se lee el pin de Echo, con el cual guardaremos el tiempo en microsegundos  
  // del viaje de la onda de sonido  
  duration = pulseIn(echoPin, HIGH);  
  
  // Calculamos la distancia  
  distance = duration * 0.0343 / 2;      // La velocidad del sonido en cm/Us dividido  
                                         // entre 2 (ida y vuelta)  
  
  // Mostramos en el monitor serial la distancia en centímetros  
  Serial.print("Distance: ");  
  Serial.print(distance);  
  Serial.println(" cm");  
}
```

# Ejercicio 2. Medir distancia con 3 sensores

Ejemplo - 9-3UltrasonicoW\_umbral\_LEDs



Robot automático

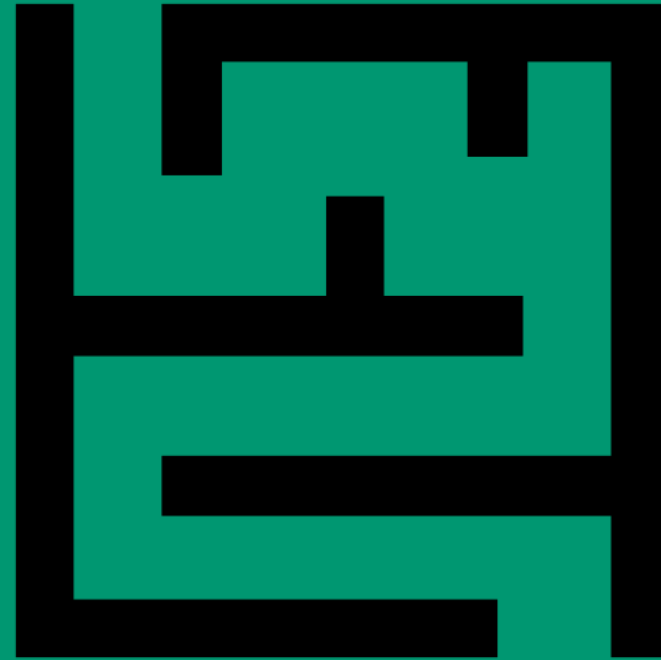
- Realice un programa donde se lean tres sensores ultrasónicos HC-SR04, para lo cual, la lectura de cada uno, deberá estar en una función.
- Se sugiere llamar a las funciones US\_left, US\_front y US\_right y cada distancia obtenida podría llamarse dist\_L, dist\_F y dist\_R
- Muestre las tres mediciones en el monitor serial de tal forma que se muestren en 3 columnas.
- Además, realice una condición para cada medición de distancia donde si esta es menor de 7 cm, se encienda cada LED, es decir si:
  - $\text{dist\_L} < 7$  -> Se enciende Led\_Verde
  - $\text{dist\_F} < 7$  -> Se enciende Led\_Rojo
  - $\text{dist\_R} < 7$  -> Se enciende Led\_Amarillo

# Robot móvil

---



Teleoperado



Automático

# Robot móvil



<https://www.youtube.com/watch?v=Zx2P1byxBj0>