



# UNIVERSIDAD DE GRANADA

## Práctica 3 de Periféricos y Dispositivos de Interfaz Humana

Curso 2024-2025

Francisco Quiles Ramírez

# Índice

Programa de parpadeo de LED.....	3
Programa de parpadeo de LED con interruptor.....	4
Secuencia de LEDs.....	5
Detector de la distancia.....	7
Detector de la cantidad de luz.....	8
Activación de motor.....	9

Para la simulación y comprobación antes de hacerlo físico con un kit de Arduino se han realizado los ejercicios en *Tinkercard*.

Todos los ejercicios menos el Detector de Distancia, por falta de material, han sido elaborados y grabados en los videos adjuntos, además del código.

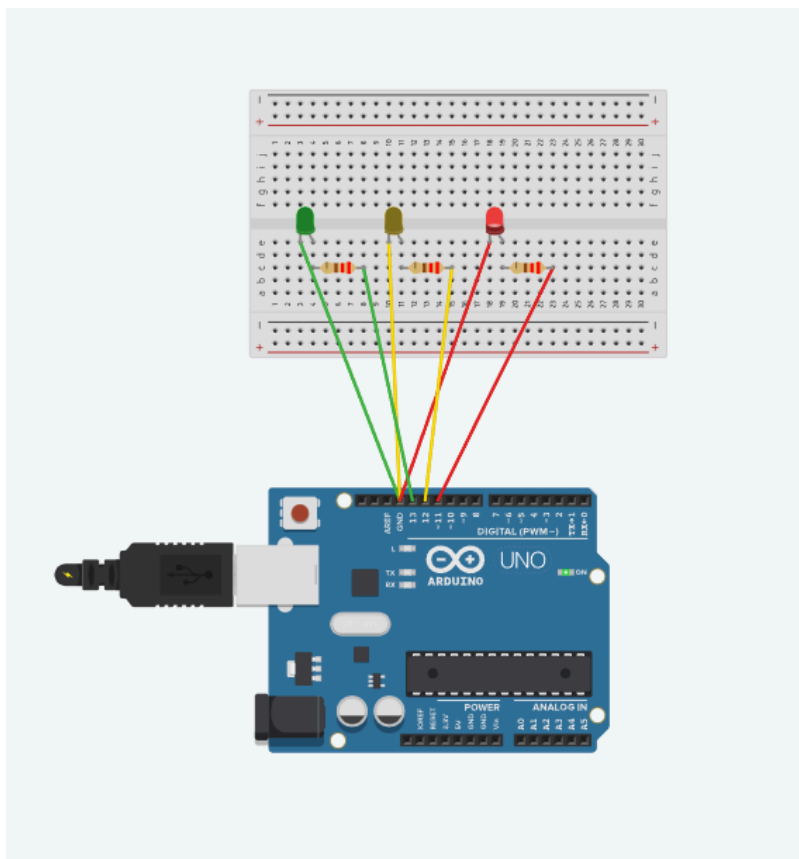
## Programa de parpadeo de LED

Se ha usado:

- 1 Arduino Uno R3
- 3 LEDs(rojo, verde y amarillo)
- 3 Resistencias de 220  $\Omega$
- 1 Placa de prueba

Para este ejercicio se han unido los 3 leds con una de las patas a tierra del Arduino y la otra a una resistencia de 220  $\Omega$  para que el led no se queme y a su vez la resistencia a los pines (verde) 13, (amarillo) 12 y (rojo) 11.

En el código inicializamos los pines como salidas y aplicamos las salidas para encender el led correspondiente dependiendo de su pin, también se aplica un delay entre cada acción para conseguir el resultado obtenido.



```

1 // C++ code
2 void setup() {
3     pinMode(11, OUTPUT); // LED rojo
4     pinMode(12, OUTPUT); // LED amarillo
5     pinMode(13, OUTPUT); // LED verde
6 }
7
8 void loop() {
9     digitalWrite(11, HIGH); // Encender rojo
10    digitalWrite(12, LOW); // Apagar amarillo
11    digitalWrite(13, LOW); // Apagar verde
12    delay(1500); // 1.5 segundos
13
14    digitalWrite(11, LOW);
15    digitalWrite(12, HIGH); // Encender amarillo
16    digitalWrite(13, LOW);
17    delay(1500);
18
19    digitalWrite(11, LOW);
20    digitalWrite(12, LOW);
21    digitalWrite(13, HIGH); // Encender verde
22    delay(1500);
23 }
24

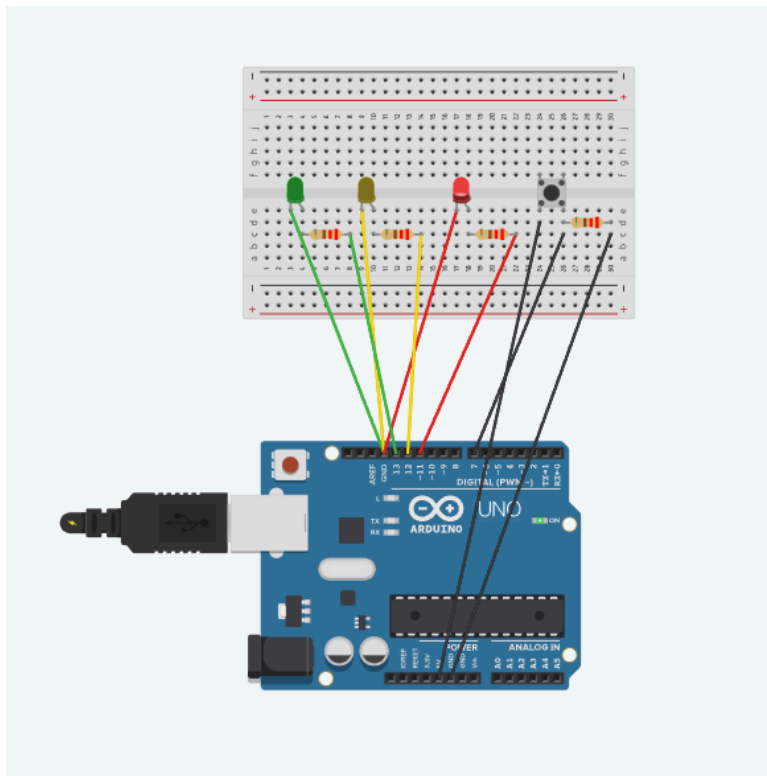
```

## Programa de parpadeo de LED con interruptor

Se ha usado:

- 1 Arduino Uno R3
- 3 LEDs(rojo, verde y amarillo)
- 3 Resistencias de 220  $\Omega$
- 1 Placa de prueba
- 1 interruptor

Se ha añadido un interruptor el cual se ha conectado a la entrada digital 7 y a la corriente de 5V y además a tierra usando otra resistencia de 220  $\Omega$ . En el código se inicializa con input 7, la del interruptor y en cada iteración del bucle lo comprobamos el estado del interruptor, para conseguir de esta forma el comportamiento deseado



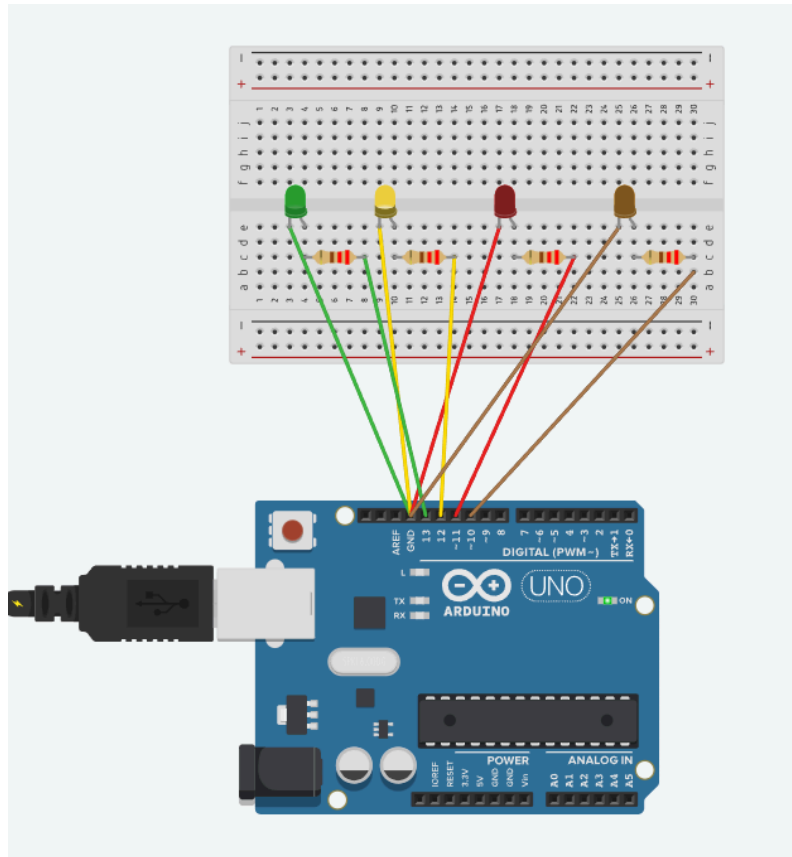
```
1 void setup() {  
2   pinMode(11, OUTPUT); // LED rojo  
3   pinMode(12, OUTPUT); // LED amarillo  
4   pinMode(13, OUTPUT); // LED verde  
5   pinMode(7, INPUT);   // Botón  
6 }  
7  
8 void loop() {  
9   int boton = digitalRead(7);  
10  
11   if (boton == HIGH) {  
12     digitalWrite(11, HIGH); // Encender rojo  
13     digitalWrite(12, LOW);  // Apagar amarillo  
14     digitalWrite(13, LOW);  // Apagar verde  
15   } else {  
16     digitalWrite(11, LOW);  
17     digitalWrite(12, HIGH); // Encender amarillo  
18     digitalWrite(13, HIGH); // Encender verde  
19   }  
20 }  
21  
22
```

## Secuencia de LEDs

Se ha usado:

- 1 Arduino Uno R3
- 4 LEDs(rojo, verde y amarillo)
- 4 Resistencias de 220  $\Omega$
- 1 Placa de prueba

Se ha añadido otro led de color naranja con otra resistencia con la misma configuracion, para su funcionamiento se inicializan los 4 leds y con un bucle for se recorren de izquierda derecha y con otro de derecha a izquierda.



```
1 int leds[] = {13,12,11,10};
2 int numLeds = 4;
3
4 void setup() {
5     for (int i = 0; i < numLeds; i++) {
6         pinMode(leds[i], OUTPUT);
7     }
8 }
9
10 void loop() {
11     // De izquierda a derecha
12     for (int i = 0; i < numLeds; i++) {
13         digitalWrite(leds[i], HIGH);
14         delay(300);
15         digitalWrite(leds[i], LOW);
16     }
17
18     // De derecha a izquierda
19     for (int i = numLeds - 2; i > 0; i--) {
20         digitalWrite(leds[i], HIGH);
21         delay(300);
22         digitalWrite(leds[i], LOW);
23     }
24 }
25
```

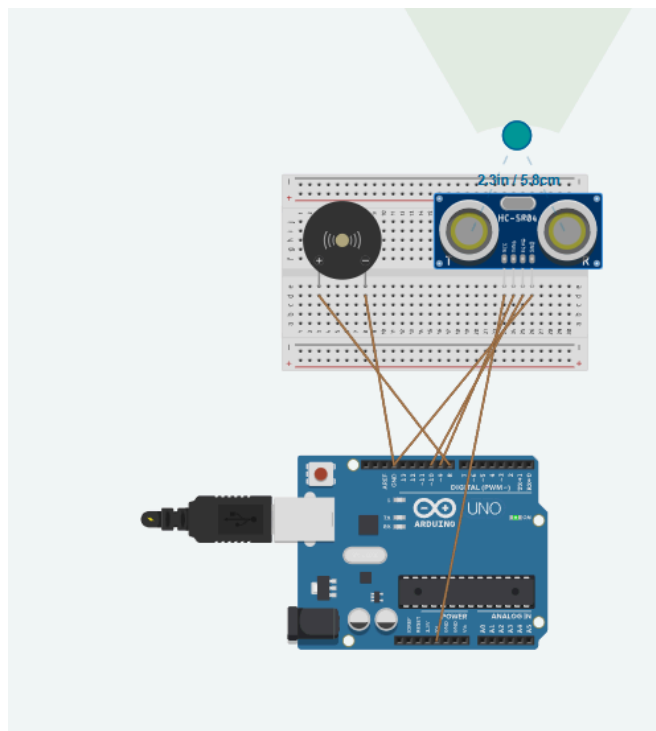
# Detector de la distancia

Se ha usado:

- 1 Arduino Uno R3
- 1 Sensor de ultrasonidos HC-SR04
- 1 Buzzer
- 1 Placa de prueba

Se ha colocado un sensor HC\_SR04 donde se ha conectado VCC a 5V, GND a GND y TRIG al pin 9 y ECHO al pin 10, además de un buzzer donde el terminal positivo estará en el pin 8 y el negativo en GND.

El objetivo es que si un objeto está a menos de 10 cm el buzzer se activa y emite sonido como podemos ver en la imagen, que el objeto está a unos 5cm y se ve que emite sonido del buzzer, para ello en el código en setup() se configura trigPin como salida, echoPin como entrada, buzzer como salida y serial.begin(9600) activa la comunicación serial para ver los datos por distancia. Se genera un pulso ultrasónico de 10 microsegundos y mide el tiempo que tarda el eco en volver al pin ECHO, se convierte el tiempo a distancia y muestra la distancia medida en el Monitor Serie, y si la distancia es menor que 10 cm se activa el buzzer.



```

1  const int trigPin = 9;
2  const int echoPin = 10;
3  const int buzzer = 8;
4
5  long duracion;
6  int distancia;
7
8  void setup() {
9      pinMode(trigPin, OUTPUT); // TRIG como salida
10     pinMode(echoPin, INPUT); // ECHO como entrada
11     pinMode(buzzer, OUTPUT); // Buzzer como salida
12
13     Serial.begin(9600);
14 }
15
16 void loop() {
17     digitalWrite(trigPin, LOW);
18     delayMicroseconds(2);
19
20     digitalWrite(trigPin, HIGH);
21     delayMicroseconds(10);
22     digitalWrite(trigPin, LOW);
23
24     duracion = pulseIn(echoPin, HIGH);
25     distancia = duracion * 0.034 / 2;
26     Serial.print("Distancia: ");
27     Serial.print(distancia);
28     Serial.println(" cm");
29
30     // Activar buzzer si distancia < 10 cm
31     if (distancia < 10) {
32         digitalWrite(buzzer, HIGH);
33     } else {
34         digitalWrite(buzzer, LOW);
35     }
36
37     delay(200);

```

## Detector de la cantidad de luz

Se ha usado:

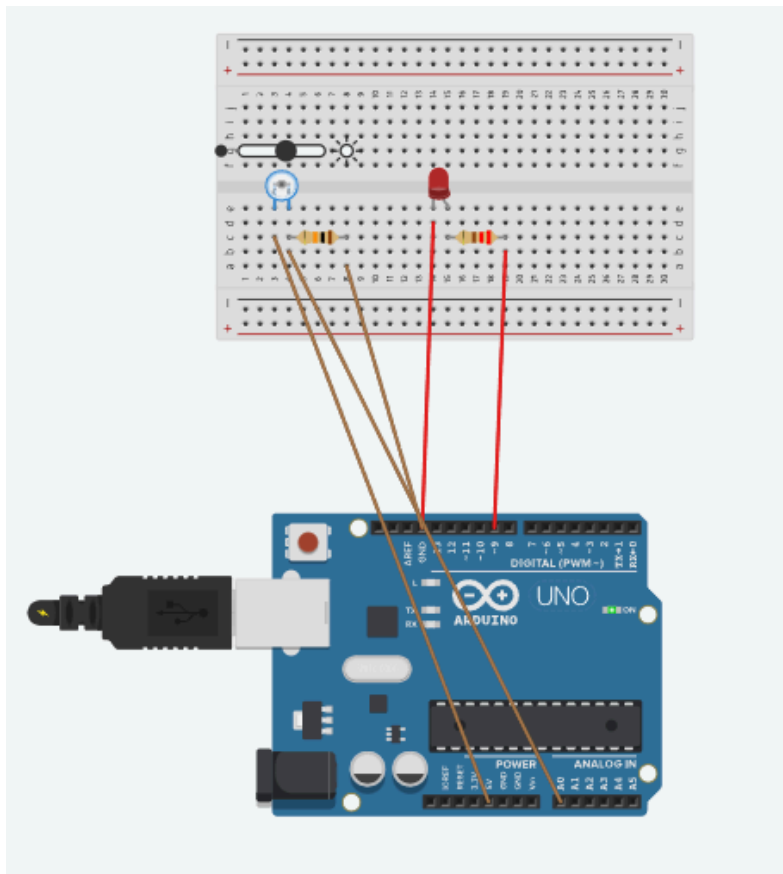
- 1 Arduino Uno R3
- 2 Resistencias (220  $\Omega$  y 10 k $\Omega$ )
- 1 Led rojo
- 1 Sensor de luz ambiental
- 1 Placa de prueba

Se ha conectado una de las patas del led a tierra del Arduino y la otra a una resistencia de 220  $\Omega$  para que el led no se queme y a su vez la resistencia a un LED rojo.

Como podemos ver al haber menos luz el pin se vuelve más brillante, y al aumentar la luz este disminuye.

Para ello en el código en setup se configura el LED como salida y se inicia la comunicación en serie para en el loop, se lee la cantidad de luz y usa map para convertir ese valor en un nivel de brillo invertido, después se aplica ese valor al LED y se imprime los valores en el monitor serie para depurar.





```
1  int sensorPin = A0;    // LDR conectado al pin A0
2  int ledPin = 9;        // LED conectado al pin 9 (PWM)
3  int valorLuz = 0;      // Lectura de la luz
4  int brillo = 0;        // Valor PWM para el LED
5
6  void setup() {
7    pinMode(ledPin, OUTPUT);
8    Serial.begin(9600);
9  }
10
11 void loop() {
12   valorLuz = analogRead(sensorPin);
13   brillo = map(valorLuz, 0, 1023, 255, 0);
14
15   analogWrite(ledPin, brillo);
16
17   Serial.print("Luz: ");
18   Serial.print(valorLuz);
19   Serial.print(" → Brillo LED: ");
20   Serial.println(brillo);
21
22   delay(100);
23 }
24
```

## Activación de motor

Se ha usado:

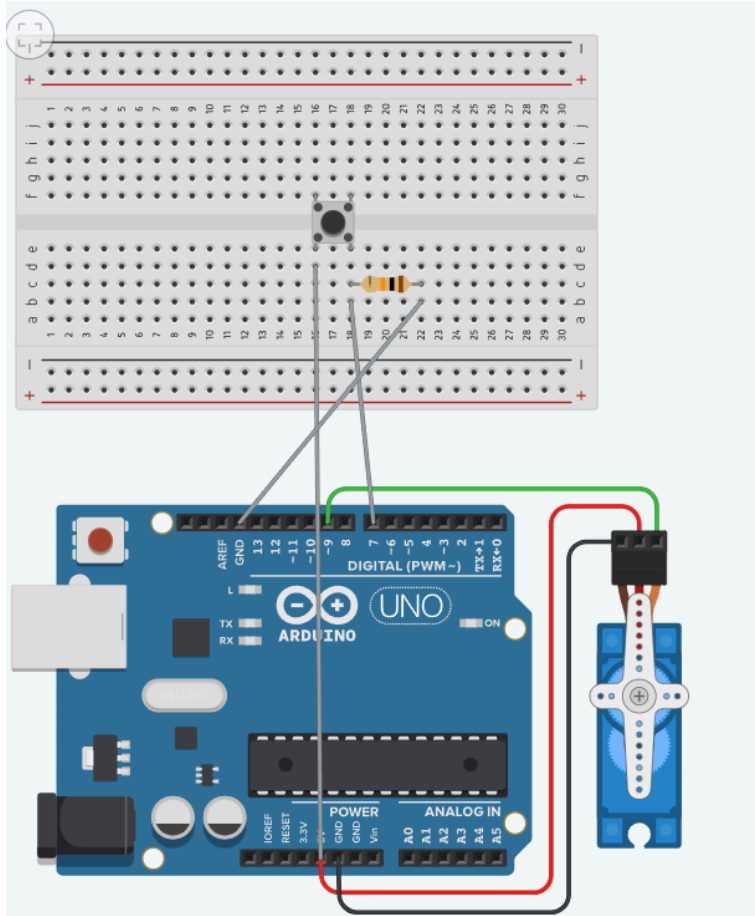
1 ServoMotor

1 Resistencia de 10 kΩ

## 1 Pulsador

Se ha conectado el motor a la placa Arduino y para ello, VCC se ha puesto en 5V, GND a GND y la señal al pin 9, un lado del pulsador en 5V y el otro al pin 7 junto a la resistencia y el otro lado de la resistencia a GND.

Para ello en el código en setup se inicializa el servo en el pin 9 y configura el pin del botón como entrada, y en loop lee el estado actual del botón, el servo se mueve 90° cuando se pulsa el botón y regresa a 0° cuando se suelta.



```
1 #include <Servo.h>
2
3 Servo miServo;
4 int botonPin = 7;
5 int estadoBoton = 0;
6
7 void setup() {
8     miServo.attach(9);           // Control del servo por pin 9
9     pinMode(botonPin, INPUT);
10 }
11
12 void loop() {
13     estadoBoton = digitalRead(botonPin);
14
15     if (estadoBoton == HIGH) {
16         miServo.write(90);       // Gira a 90° cuando se pulsa
17     } else {
18         miServo.write(0);        // Vuelve a 0° cuando no se pulsa
19     }
20
21     delay(100);
22 }
23
```