

# Práctica 3

EXPERIMENTACIÓN CON ARDUINO

Pablo García Fernández  
Periféricos y Dispositivos de Interfaz Humana

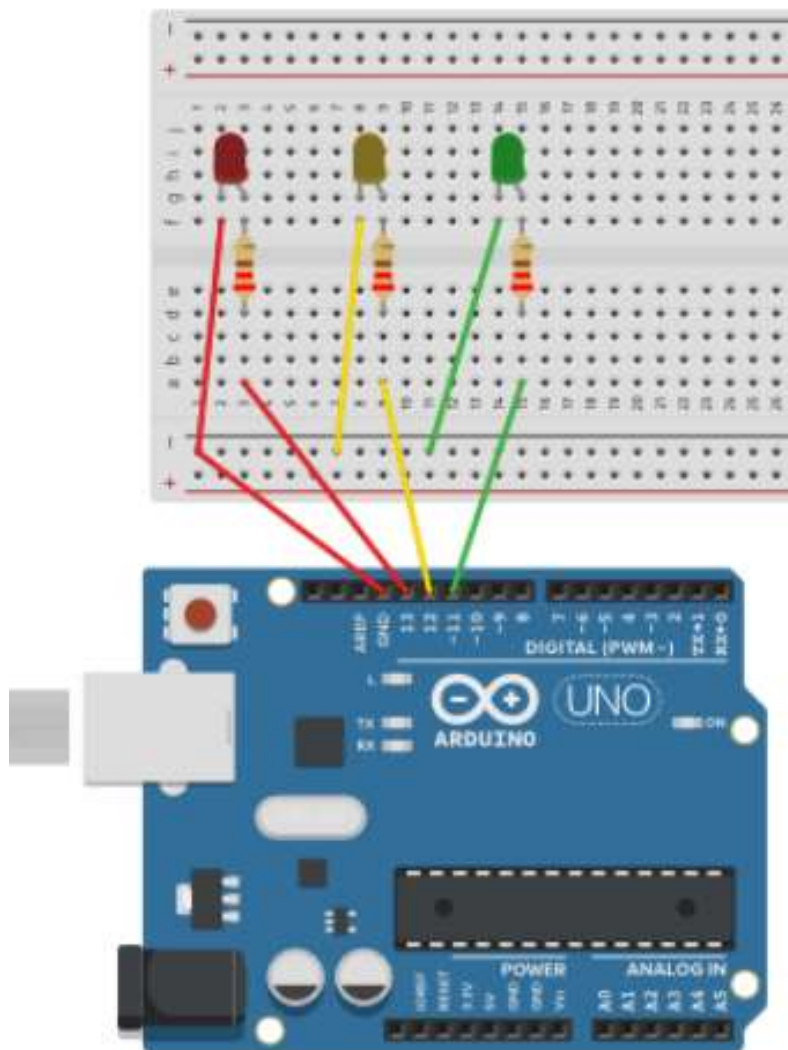


**UNIVERSIDAD  
DE GRANADA**

## Ejercicio 1 – 3 LED

En el primer ejercicio se pretende crear un programa que vaya apagando y encendiendo 3 LEDs de distintos colores en intervalos de 1.5 segundos.

Se conectan 3 cables de la toma de tierra a un lado del cada LED, luego se conecta cada uno de los puertos a la otra toma del LED utilizando una resistencia de 220 ohmios.



En el código se declara primero cada uno de los pines del Arduino que se usan y el modo *output*. En el bucle se repite el proceso para cada uno de los 3 LEDs, primero se enciende con la función `digitalWrite(PIN, HIGH)`, se esperan 1500 milisegundos como se indica y posteriormente se apaga el LED.

```

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1500); // Wait for
  digitalWrite(13, LOW);
  delay(1500); // Wait for

  digitalWrite(12, HIGH);
  delay(1500); // Wait for
  digitalWrite(12, LOW);
  delay(1500);

  digitalWrite(11, HIGH);
  delay(1500); // Wait for
  digitalWrite(11, LOW);
  delay(1500);
}

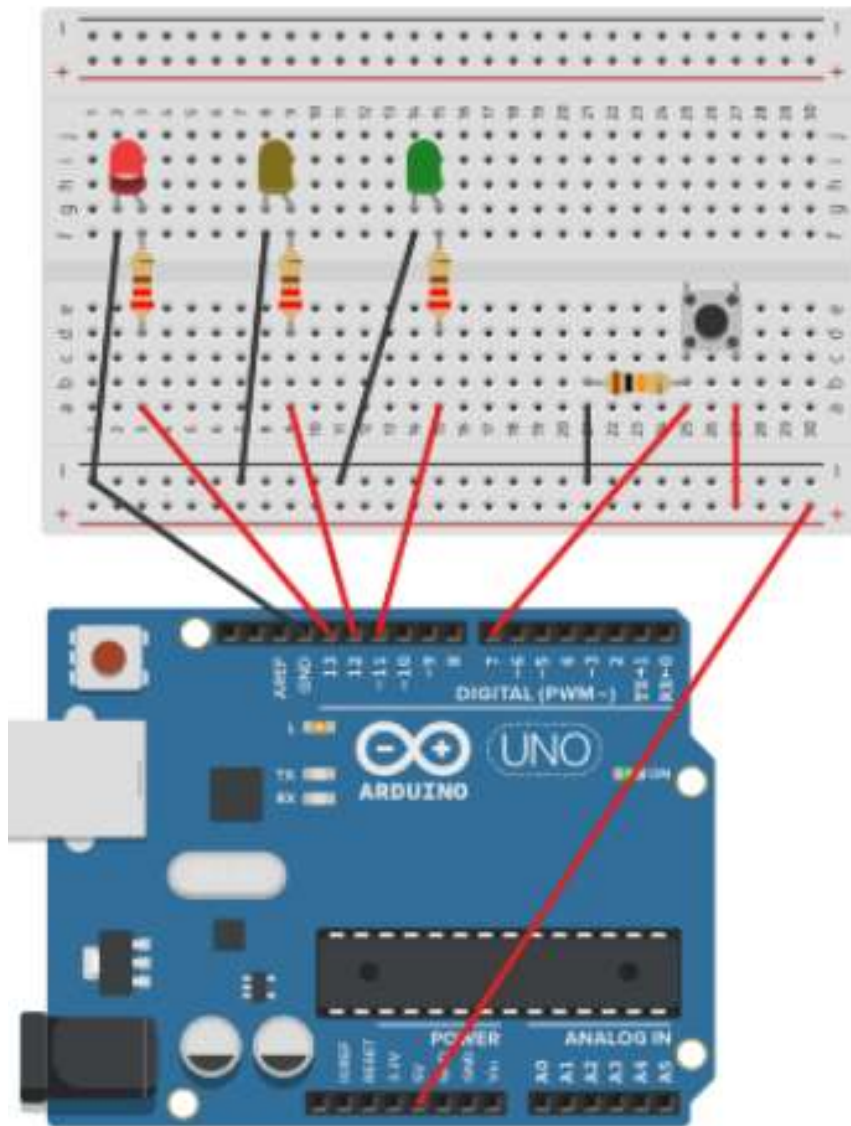
```

El video con el funcionamiento se encuentra en el mismo repositorio con el nombre 3LED y el enlace a Tinkercad es el siguiente:

<https://www.tinkercad.com/things/6EzLZrFKhD-3led?sharecode=m28uQGgERTI2VqkUXRYx4PNW3x-8KyYeqhnDqROqFrc>

## Ejercicio 2 – Botón y LED

El segundo ejercicio consiste en utilizar un botón para encender el LED rojo y apagar los LEDs amarillo y verde. Las conexiones para los LEDs son exactamente iguales que en el ejercicio anterior y para el botón se conecta una patilla a la toma de 5V y la otra se conecta al puerto digital 7 del Arduino y a la toma de tierra a través de una resistencia de 10 kilo ohmios.



En el código se declaran los LED de la misma forma que en el anterior y el puerto correspondiente al botón (numero 7) se establece de entrada.

En el bucle se va leyendo el estado del botón, en caso de que no se haya pulsado, se mantienen encendidos los LED amarillo y verde y se apaga el rojo.

En caso de que se haya pulsado el botón se apagarán los dos LEDs mencionados y se enciende el rojo.

```
void setup()
{
  pinMode(7, INPUT);

  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}

void loop()
{
  if(digitalRead(7) == HIGH) {
    digitalWrite(13, HIGH);
    digitalWrite(12, LOW);
    digitalWrite(11, LOW);
  }
  else{
    digitalWrite(13, LOW);
    digitalWrite(12, HIGH);
    digitalWrite(11, HIGH);
  }
}
```

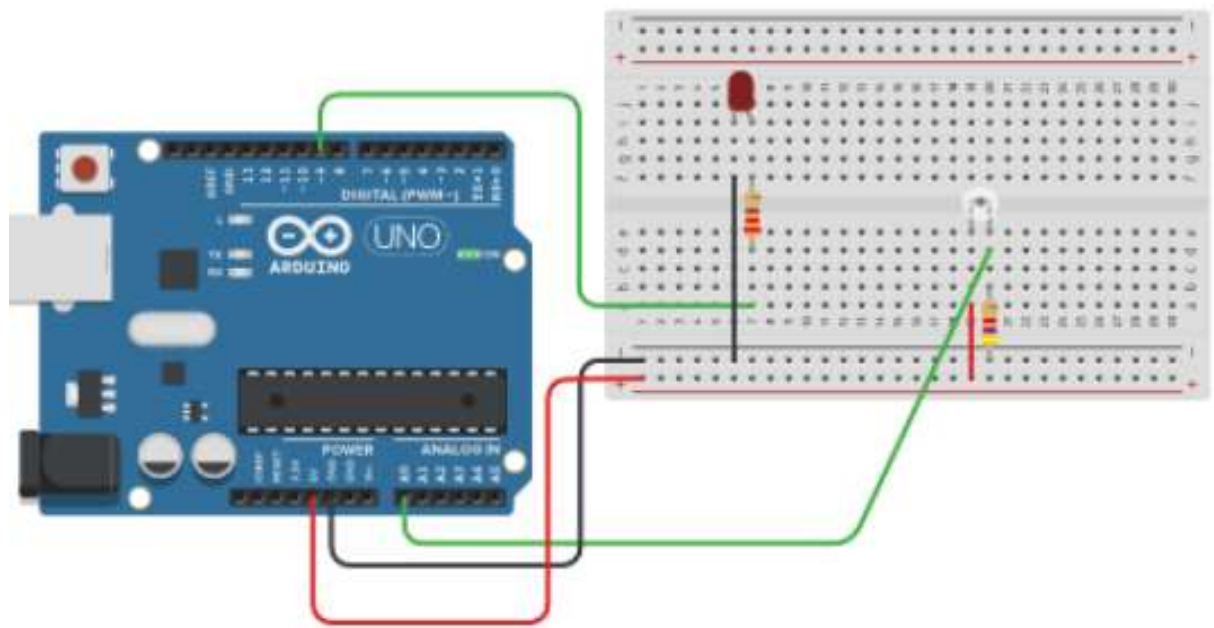
En el repositorio se encuentra el video con la demostración del funcionamiento con el nombre LEDButton. El enlace en Tinkercad es el siguiente:

[https://www.tinkercad.com/things/ksllMMDenv2-ledbutton?sharecode=wIPNE2mfy3oMbcXgy2\\_9bJbxs5qTo\\_xiW9VT4UEJHw8](https://www.tinkercad.com/things/ksllMMDenv2-ledbutton?sharecode=wIPNE2mfy3oMbcXgy2_9bJbxs5qTo_xiW9VT4UEJHw8)

## Ejercicio 3 – Sensor de Luz

Para este ejercicio se crea un programa donde un LED se irá encendiendo con mayor intensidad según la luz en el ambiente detectada por un fotoresistor.

El LED se conecta de la misma forma que en ejercicios anteriores, pero usando el PIN 9 que nos permite modular la intensidad. Para el fotoresistor se conecta una patilla a tierra a través de una resistencia de 4.7 kilo ohmios y la otra patilla al PIN analógico Ao.



En la función *setup()* se indica el pin de entrada Ao para el fotoresistor y el pin de salida 9 para el LED. También se usa la función *serial.begin(9600)* para establecer la velocidad de información de la placa.

En el bucle se lee y almacena el valor que se percibe del fotoresistor y se transforman los datos a un rango entre 0 y 255. Esto se hace ya que los valores que admite el LED se encuentran en ese rango y los valores recibidos por el puerto Ao se salen.

Una vez hecha la transformación se indica al LED el nivel de luminiscencia que debe adquirir. También se indica por consola los valores recibidos y transformados y se aplica un retardo.

```

int sensorValue = 0;

int outputValue = 0;

void setup()
{
  pinMode(A0, INPUT);
  pinMode(9, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  sensorValue = analogRead(A0);

  outputValue = map(sensorValue, 0, 1500, 0, 255);

  analogWrite(9, outputValue);

  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);

  delay(2);
}

```

Debido a que no había disponibilidad de un fotoresistor no existe video para la demostración pero en el siguiente enlace de Tinkercad se puede simular para ver el funcionamiento: <https://www.tinkercad.com/things/gkietxupO97-lightsensor?sharecode=Hw6bZNxDVo646t8nzYmdkrcvni7gP1yvUasvaigPFac>