

PRÁCTICAS DE ARDUINO

Práctica nº 2: Encendido y apagado de varios LEDs

En esta práctica continuamos con aspectos básicos de ARDUINO con el encendido y apagado de varios LEDs.

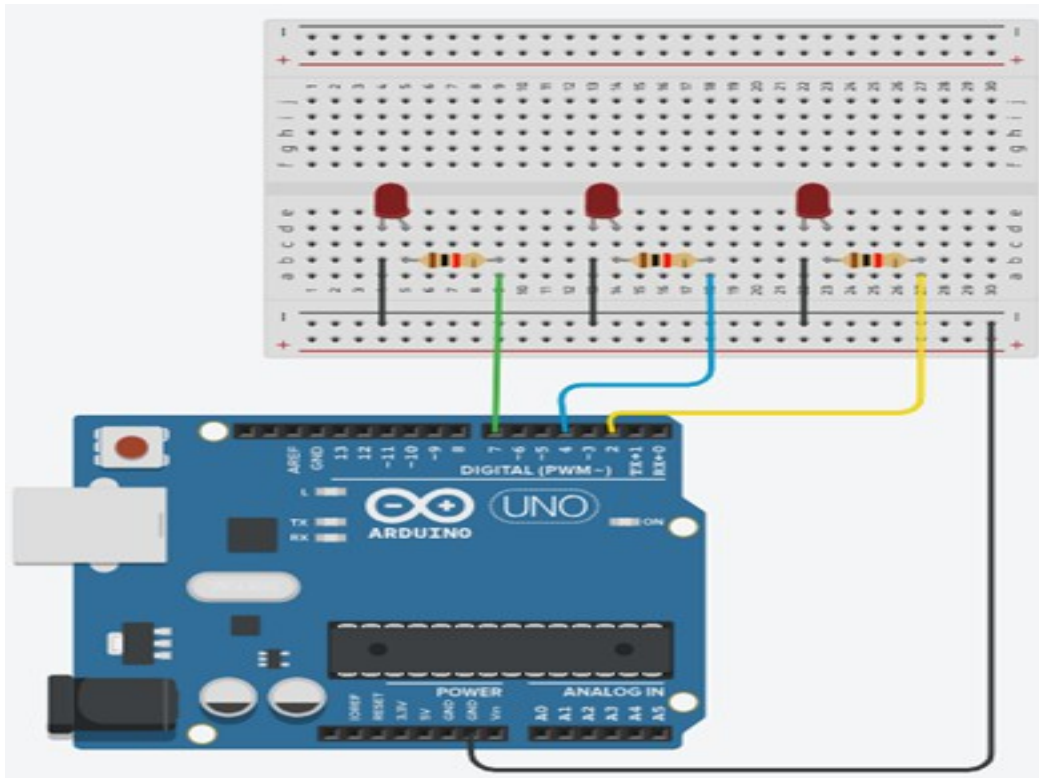
MATERIALES

- Arduino UNO.
- Cable USB tipo A-B.
- 3 LEDs rojos de 3mm
- 2 LEDs verdes de 3mm
- 1 LED amarillo de 3mm
- 5 Resistencias de 220Ω.
- 1 Resistencia de 1KΩ.
- 1 placa Protoboard.
- Cables de conexión.

MONTAJE 1

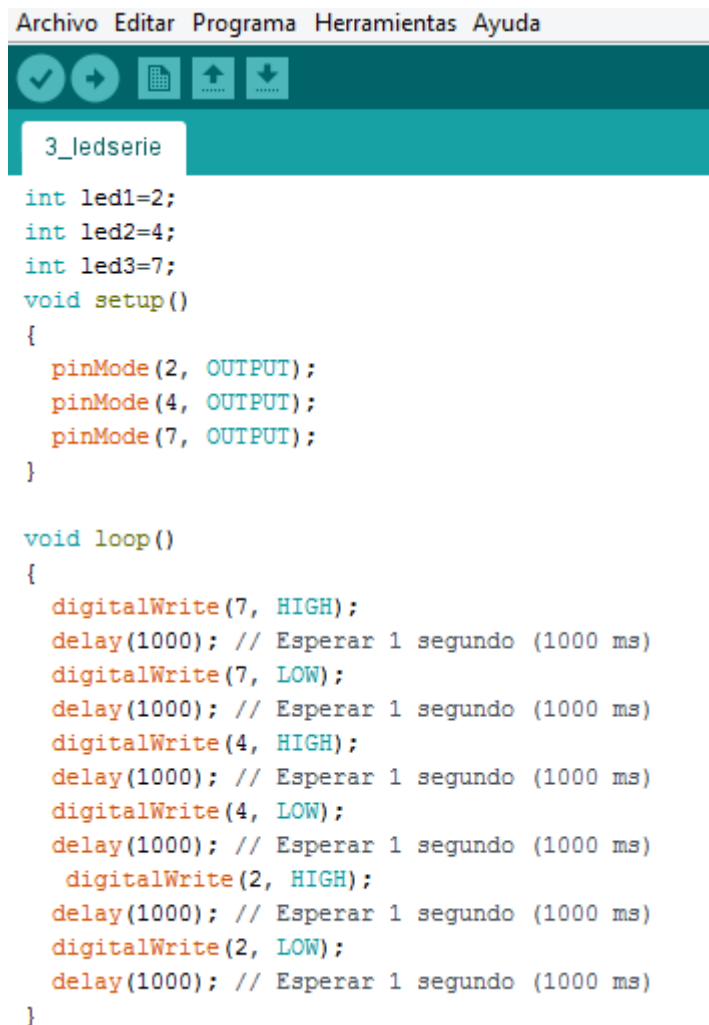
La actividad que vamos a realizar consiste en la conexión de 3 LED a través de una protoboard. La idea es que enciendan siguiendo un orden previamente establecido, enciende el primer LED, dura un segundo encendido, se apaga, enciende el siguiente... se paga y luego el siguiente.

La conexión eléctrica es realmente sencilla. Simplemente ponemos una resistencia en serie conectada con cada LED. El montaje en una protoboard quedaría de la siguiente forma.



PROGRAMACIÓN 1

Para probar la conexión se utilizó el siguiente código con el IDE de Arduino. En la primera parte se definen los 3 led que se emplearán a través de los pines 2, 4 y 7 como variables. Posteriormente se configuran los pines en modo OUTPUT (salida) y en la parte inferior se configura el ciclo (loop) de encendido y apagado de cada LED.

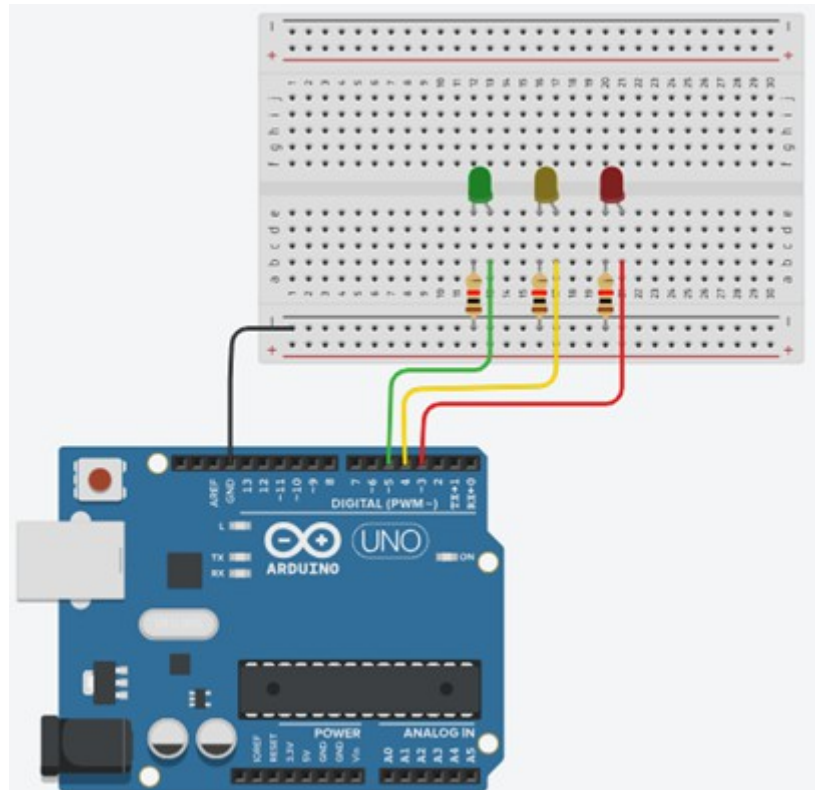
The image shows a screenshot of the Arduino IDE interface. At the top, there is a menu bar with 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. Below the menu bar is a toolbar with icons for a checkmark, a right arrow, a grid, and upload/download arrows. The file name '3_ledserie' is displayed in the title bar. The main text area contains the following C++ code:

```
int led1=2;
int led2=4;
int led3=7;
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(7, OUTPUT);
}

void loop()
{
  digitalWrite(7, HIGH);
  delay(1000); // Esperar 1 segundo (1000 ms)
  digitalWrite(7, LOW);
  delay(1000); // Esperar 1 segundo (1000 ms)
  digitalWrite(4, HIGH);
  delay(1000); // Esperar 1 segundo (1000 ms)
  digitalWrite(4, LOW);
  delay(1000); // Esperar 1 segundo (1000 ms)
  digitalWrite(2, HIGH);
  delay(1000); // Esperar 1 segundo (1000 ms)
  digitalWrite(2, LOW);
  delay(1000); // Esperar 1 segundo (1000 ms)
}
```

MONTAJE 2

Ahora montaremos un semáforo con tres leds Rojo, Amarillo y Verde y sus correspondientes resistencias de protección de 220 ohmios entre el pin y el led, para evitar que el led se funda.



Aprovechamos para introducir un nuevo comando: **Serial.print**. Este comando nos manda un texto al puerto serial por el que nos comunicamos con Arduino. De esta manera podemos depurar un programa sabiendo siempre porque línea está.

- `Serial.print("xxx")` escribe lo que ponemos entre comillas tal cual.
- `Serial.print(x)` escribe el valor que contenga la variable x.
- `Serial.println()` es similar a lo anterior pero después añade un salto de línea.

El MONITOR SERIE se abre en el Arduino IDE, menú *Herramientas --> Monitor Serie*. Para que funcione debemos tener en cuenta que en el programa que escribimos es necesario inicializar la velocidad de comunicación serie. Esto se hace poniendo **Serial.begin (9600)** dentro de la rutina de `setup()`

Para ver lo que nuestro Arduino nos envía a través del cable USB, cargamos el programa y luego abrimos el monitor Serial en el menú herramientas del programa Arduino.

PROGRAMACIÓN 2

```

/* Semáforo Arduino
Leds conectados a pines 3, 4 y 5*/
int verde = 5;
int amarillo = 4;
int rojo = 3;

void setup()
{
  pinMode(verde, OUTPUT);
  pinMode(amarillo, OUTPUT);

```

```
pinMode(rojo, OUTPUT);
Serial.begin(9600); //inicializa la comunicación Serial
}

void loop()
{
  Serial.println("Semaforo - Inicio"); //Escribe el texto
  digitalWrite(verde, HIGH);
  Serial.println("Semaforo - Verde"); //Escribe el texto
  delay(3000);
  digitalWrite(verde, LOW);
  digitalWrite(amarillo, HIGH);
  Serial.println("Semaforo - Amarillo"); //Escribe texto
  delay(1000);
  digitalWrite(amarillo, LOW);
  digitalWrite(rojo, HIGH);
  Serial.println("Semaforo - Rojo"); //Escribe el texto
  delay(2000);
  digitalWrite(rojo, LOW);
}
```

Ejercicios propuestos

1. Realiza cada uno de estos montajes y comprueba que los programas funciona correctamente en tu placa Arduino.
2. Conecta un cuarto led en el montaje 1 y realiza un nuevo programa para que funcione.