



ugr

Universidad
de Granada

PDIH

PERIFÉRICOS Y DISPOSITIVOS DE INTERFAZ HUMANA

Práctica 3. Experimentación con Arduino

Autor: Miguel Molinero Martin



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Curso 2023 - 2024

DESARROLLO PRÁCTICA 3:

Ejercicio 1: Implementar el programa de parpadeo de LED, ampliándolo para que encienda y apague alternativamente tres LEDs (uno rojo, otro amarillo y otro verde), conectados a las salidas digitales 11, 12 y 13 del Arduino, a un intervalo de 1.5 segundos. Crear el esquema con Fritzing y cargar el programa en Arduino para comprobar que funciona correctamente.

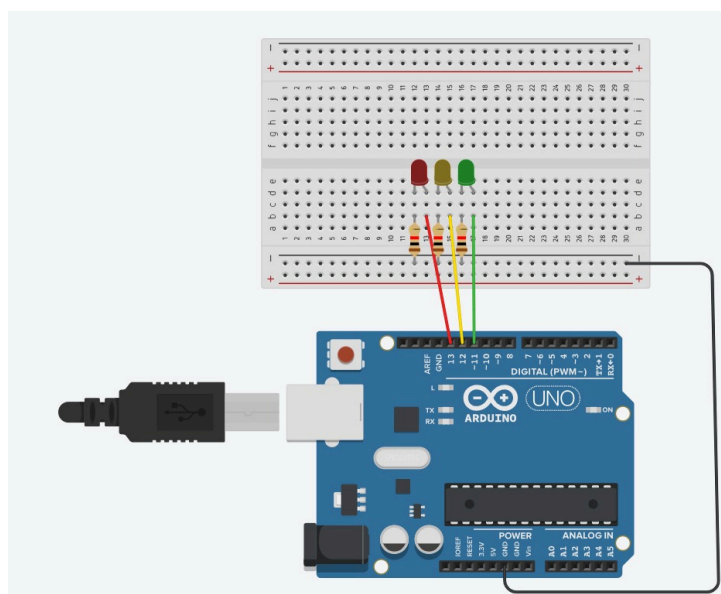
```
void setup() { // código de inicialización (se ejecuta una vez al principio)
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}

void loop() { // código del programa principal (se ejecuta repetidamente)
  digitalWrite(11, HIGH); // encender el LED (voltaje a HIGH)
  delay(1000); // esperar encendido durante 1 segundo (1000 mseg)
  digitalWrite(11, LOW); // apagar el LED (voltaje a LOW)
  delay(500); // esperar apagado durante 100 milisegundos

  digitalWrite(12, HIGH); // encender el LED (voltaje a HIGH)
  delay(1000); // esperar encendido durante 1 segundo (1000 mseg)
  digitalWrite(12, LOW); // apagar el LED (voltaje a LOW)
  delay(500); // esperar apagado durante 100 milisegundos

  digitalWrite(13, HIGH); // encender el LED (voltaje a HIGH)
  delay(1000); // esperar encendido durante 1 segundo (1000 mseg)
  digitalWrite(13, LOW); // apagar el LED (voltaje a LOW)
  delay(500); // esperar apagado durante 100 milisegundos
}
```

Para implementar este programa en Arduino encendemos y apagamos tres LEDs en secuencia. En la función **setup()**, se configuran los pines como salidas, y en la función **loop()**, se encienden y apagan los LEDs conectados a los pines 11, 12 y 13, respectivamente, con intervalos de 1.5 segundos. Este proceso se repite continuamente.



<https://youtu.be/w2UN6Wbamsk>

Ejercicio 2: Partir del programa de parpadeo de LEDs anterior y ampliarlo con las modificaciones necesarias para que se encienda el LED rojo solo cuando se pulse un interruptor conectado a la entrada digital 7, y en ese momento se apaguen los LEDs amarillo y verde.

```
int interruptorPin = 7; // Definimos el pin del interruptor al 7

void setup() {
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(interruptorPin, INPUT); // Configuramos el pin de entrada del interruptor
}

void loop() {
  // Lee el estado del interruptor
  int interruptorEstado = digitalRead(interruptorPin);

  // Si el interruptor está presionado (HIGH)
  if (interruptorEstado == HIGH) {
    digitalWrite(13, HIGH); // Enciende el LED rojo
    digitalWrite(12, LOW); // Apaga el LED amarillo
    digitalWrite(11, LOW); // Apaga el LED verde
    return;
  }

  digitalWrite(13, LOW);

  // Si el interruptor no está presionado, sigue parpadeando los LEDs como antes
  digitalWrite(11, HIGH);
  delay(1000);
  digitalWrite(11, LOW);
  delay(500);

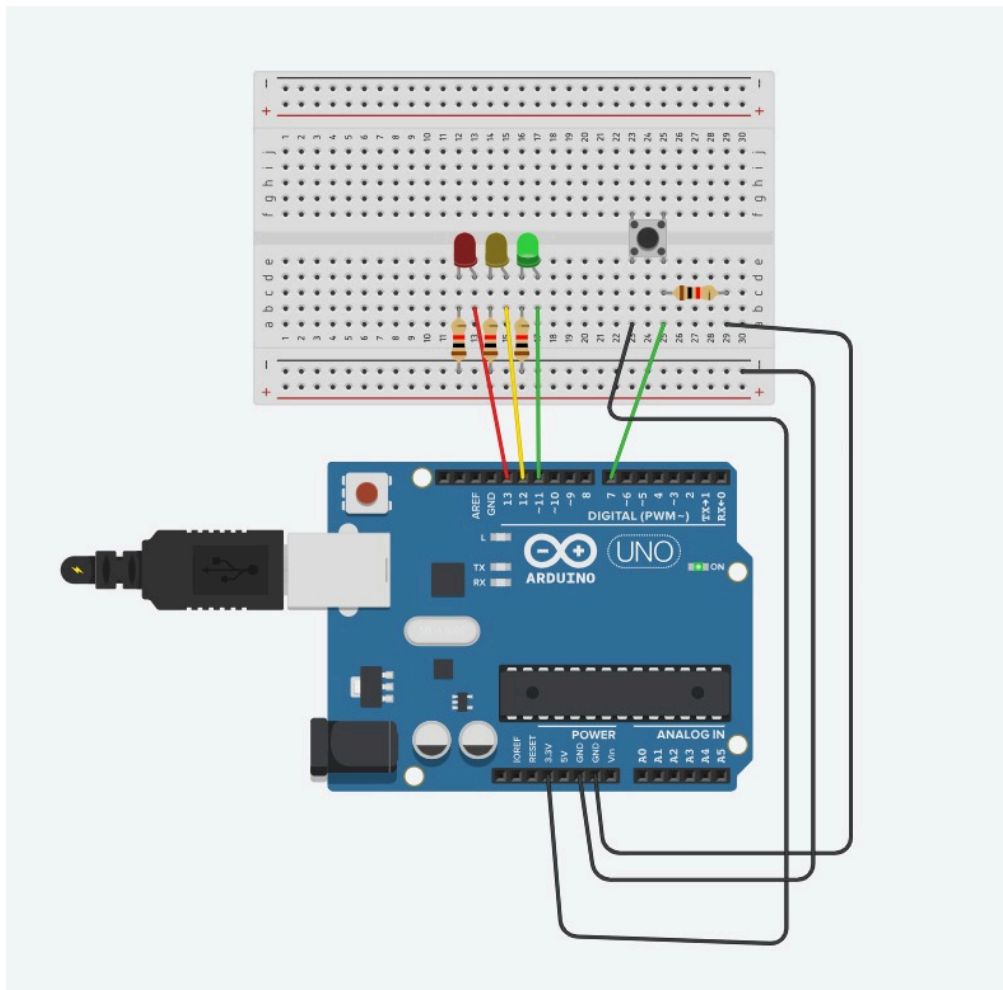
  digitalWrite(12, HIGH);
  delay(1000);
  digitalWrite(12, LOW);
  delay(500);

  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(500);
}
```

Este código es una extensión del anterior, agregando la funcionalidad de un interruptor que controla el estado de los LEDs.

Definimos una variable para el interruptor en el pin 7. En el setup() configuramos los pines de salida y de entrada, en este caso el del interruptor.

En el loop() añadimos un condicional IF para controlar si el interruptor está presionado o no y lo leemos con la función digitalRead y lo guardamos en una variable llamada interruptorEstado. En el caso de estar pulsado, encendemos el led rojo y apagamos las otras 2. Si no está pulsado, dejamos que siga parpadeando con la secuencia anterior.



<https://youtu.be/m4kJNRvMtKY>

Ejercicio ampliado: Secuencia de LEDs, encendiendo y apagando 4 LEDs secuencialmente, de forma similar a las lucecitas de "El coche fantástico":

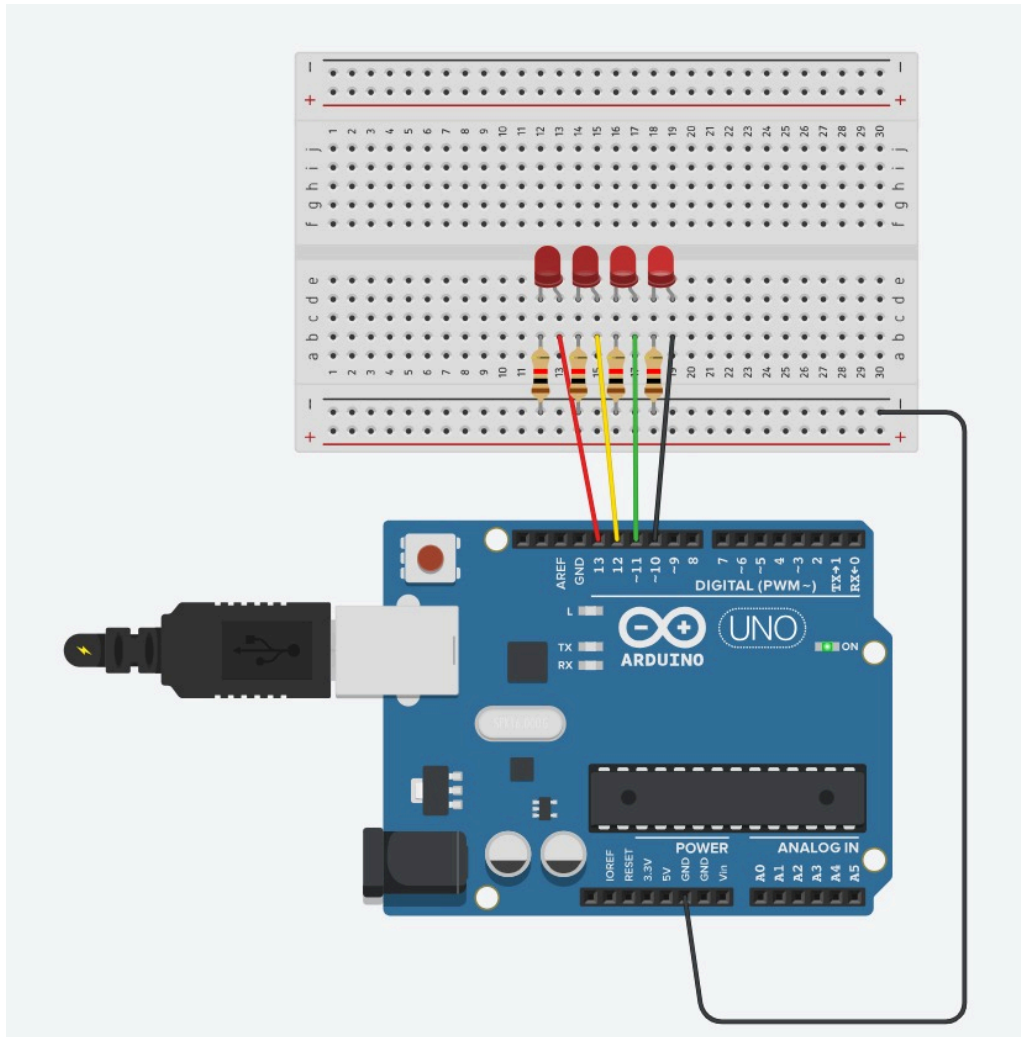
```
//configuramos las variables de los pins utilizados para los LEDs y el numero de pins
const int ledPins[] = {13, 12, 11, 10};
const int numLeds = 4;

void setup() {
  // configuramos los pines de los LEDs
  for (int i = 0; i < numLeds; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
}

void loop() {
  // bucle para encender los LEDs de manera ascendente
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH);
    delay(100);
    digitalWrite(ledPins[i], LOW);
  }

  // bucle para apagar los LEDs de manera descendente
  for (int i = numLeds - 1; i >= 0; i--) {
    digitalWrite(ledPins[i], HIGH);
    delay(100);
    digitalWrite(ledPins[i], LOW);
  }
}
```

Para este ejercicio configuramos y controlamos cuatro LEDs conectados a los pines digitales 13, 12, 11 y 10 de Arduino. En la función `setup()`, se establecen estos pines como salidas. En el bucle `loop()`, primero enciende los LEDs de manera ascendente, uno por uno, y luego los apaga de manera descendente, creando un efecto visual de encendido y apagado en secuencia. Cada LED permanece encendido durante 100 milisegundos antes de apagarse.



https://youtu.be/f_eR8tKs20M