Control programado con Arduino

Práctica 2: O parpadeo alterno (Variables e Control Secuencial)

Na práctica anterior aprendemos a manter un único LED acendido de forma continua. Nesta sesión daremos un paso máis introducindo un segundo LED para crear un efecto de parpadeo alternativo (como as luces dun cruzamento ferroviario ou un vehículo de emerxencia). Para lograr un código ordenado e escalable, aprenderás a declarar e inicializar múltiples variables independentes.

Obxectivos de Aprendizaxe

[1pt]

Ao rematar esta sesión serás capaz de:

- Declarar e utilizar múltiples variables de tipo enteiro (`int`) no teu código.
- Deseñar secuencias temporais lógicas e ordenadas utilizando a instrución `delay` .
- Comprender o comportamento físico de saídas independentes conectadas en paralelo.
- Optimizar o mantemento do teu programa mediante o uso de nomes de variables descritivos.

Compoñentes Necesarios

[1pt]

Busca e coloca os seguintes compoñentes na túa mesa de traballo de TinkerCad:

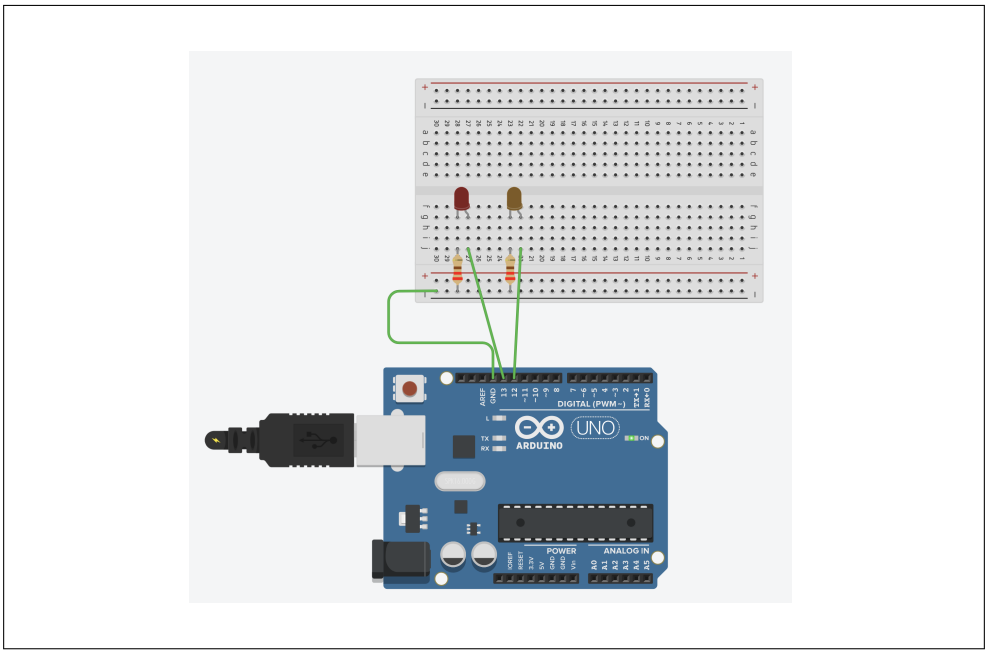
- 1 Placa Arduino Uno.
- 1 Placa de prototipado pequena (Protoboard).
- 2 Diodos LED (recomendámosche usar dúas cores diferentes, por exemplo: Verde e Vermello).
- 2 Resistencias de $220\,\Omega$ (unha para protexer cada LED).
- Cables de conexión virtuais.

Esquema de montaxe na Protoboard

[1pt]

Engadiremos un segundo LED ao circuíto que montaches na Práctica 1. Asegúrate de conectar cada ánodo a un pin dixital de control independente.

Configuración inicial do obxecto Terminal



Código Base: Declaración de Variables e Alternancia

[1pt]

Abre o editor de código en TinkerCad en modo texto. Borra o programa previo e escribe o seguinte código secuencial. Analiza como o estado de cada saída dixital se coordina de xeito síncrono no tempo.

```
// 1. DECLARACIÓN DE VARIABLES
// Reservamos espazos de memoria para gardar os pins dos LEDs
int ledRojo = 13;
int ledVerde = 12;

void setup() {
  // Configuramos ambos pins como saídas de corrente
  pinMode(ledRojo, OUTPUT);
  pinMode(ledVerde, OUTPUT);
}

void loop() {
  // ESTADO 1: Vermello acendido e Verde apagado
  digitalWrite(ledRojo, HIGH);
  digitalWrite(ledVerde, LOW);
  delay(1000); // Pausa de 1 segundo neste estado

  // ESTADO 2: Vermello apagado e Verde acendido
  digitalWrite(ledRojo, LOW);
  digitalWrite(ledVerde, HIGH);
  delay(1000); // Pausa de 1 segundo neste estado
}
```

Listing 1: Código para controlar o parpadeo alternativo de dous LEDs.

Como funciona o código?

[1pt]

Neste programa, o Arduino executa as instrucións de forma estritamente lineal dentro da función `loop` :

1. **Variables globais**: Declaramos `int` e `int` ao principio. Isto permítenos usar palabras comprensibles en lugar de escribir números fixos (*hardcoded*) ao longo do código.
2. **Alternancia síncrona**: Para lograr que un LED se apague mentres o outro se acende, enviamos sinais opostos de forma simultánea. No Estado 1 aplicamos `HIGH` ao vermello e `LOW` al verde. Tras un segundo de pausa, invertemos os valores no Estado 2.
3. **Bucle continuo**: Ao rematar o Estado 2, o fluxo volve de inmediato ao principio de `loop` , acendendo o vermello e apagando o verde, creando así o bucle infinito de alternancia.

Importancia da orde lóxica

Se non apagásemos explicitamente o LED contrario ao cambiar de estado (é dicir, omitindo as instrucións `LOW`), o programa simplemente acendería ambos LEDs secuencialmente e déixaríaos prendidos de forma indefinida, perdendo o efecto de parpadeo alternativo.

O Reto da Práctica 2

Comproba que comprendes o fluxo temporal e o uso de variables resolvendo estas tres propostas no teu simulador:

1. **Efecto Policial (Parpadeo Estroboscópico)**: Modifica os tempos dos retardos (`delay()`) para lograr unha alternancia moi rápida que emule as luces de emerxencia dunha patrulla (por exemplo, 150 ms de pausa).
2. **O poder das variables**: Cambia o cable del LED Verde ao **pin dixital 8** en TinkerCad. Realiza un único cambio no teu código (aproveitando as variables) para que o teu programa siga funcionando perfectamente sen alterar o `setup()` nin o `loop()`.
3. **Fase de Apagado Completo**: Modifica o programa para que, antes de cada cambio de luz, exista unha breve fase intermedia onde os dous LEDs permanezan apagados ao mesmo tempo durante medio segundo (500 ms).