

AE3. SISTEMA INFORMÁTICO SIMULANDO UN SEMÁFORO - ARDUINO



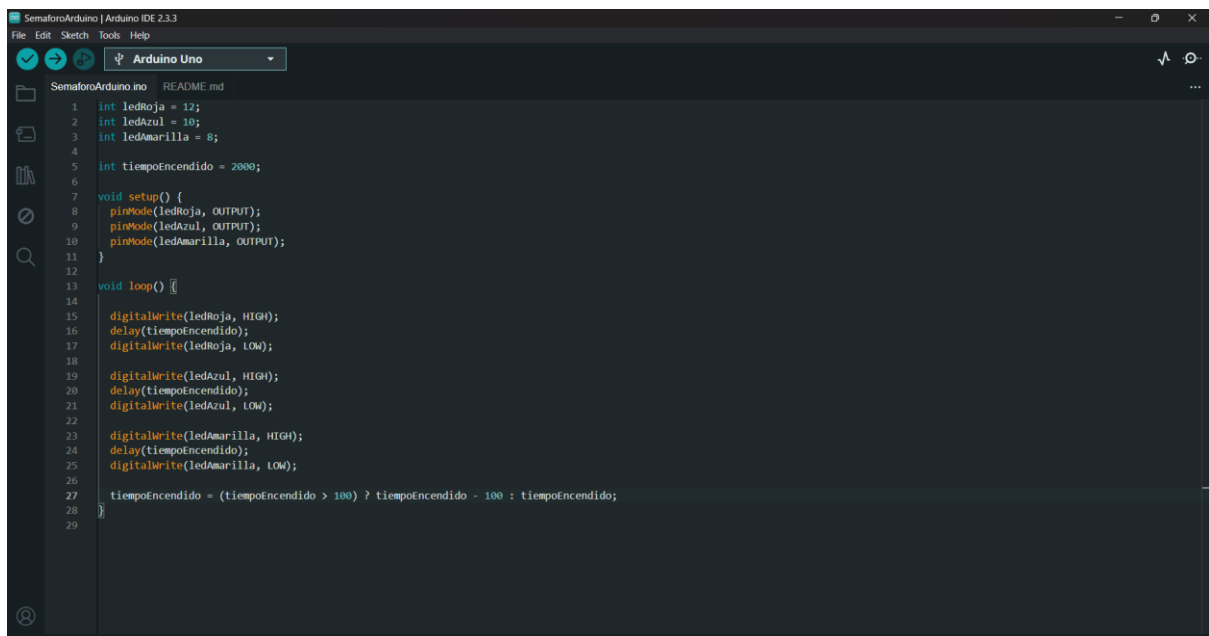
Contents

AE3. SISTEMA INFORMÁTICO SIMULANDO UN SEMÁFORO - ARDUINO	1
Introducción	3
Código	3
Descripción de la lógica del código:	3
Imagen de demostración	5
Gif de demostración	6
Captura Simulador TinkerCad	7
Conclusión	7

Introducción

En esta práctica, se implementará un sistema que simula el funcionamiento de un semáforo utilizando LEDs. El semáforo tendrá tres luces (rojo, amarillo y azul) que se encenderán en un orden específico con un tiempo que disminuye en cada ciclo.

Código

A screenshot of the Arduino IDE interface. The title bar reads 'SemaforoArduino | Arduino IDE 2.3.3'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar shows icons for opening files, saving, compiling, and uploading. The 'Tools' dropdown menu is set to 'Arduino Uno'. The main editor window displays a sketch named 'SemaforoArduino.ino' with the following code:

```
1 int ledRoja = 12;
2 int ledAzul = 10;
3 int ledAmarilla = 8;
4
5 int tiempoEncendido = 2000;
6
7 void setup() {
8   pinMode(ledRoja, OUTPUT);
9   pinMode(ledAzul, OUTPUT);
10  pinMode(ledAmarilla, OUTPUT);
11 }
12
13 void loop() {
14
15   digitalWrite(ledRoja, HIGH);
16   delay(tiempoEncendido);
17   digitalWrite(ledRoja, LOW);
18
19   digitalWrite(ledAzul, HIGH);
20   delay(tiempoEncendido);
21   digitalWrite(ledAzul, LOW);
22
23   digitalWrite(ledAmarilla, HIGH);
24   delay(tiempoEncendido);
25   digitalWrite(ledAmarilla, LOW);
26
27   tiempoEncendido = (tiempoEncendido > 100) ? tiempoEncendido - 100 : tiempoEncendido;
28 }
29
```

Descripción de la lógica del código:

Se definen tres variables de tipo entero (ledRoja, ledAmarilla, ledAzul) que representan los números de los pines a los que están conectados los LEDs:

ledRoja está conectado al pin 12.

ledAmarilla está conectado al pin 10.

ledAzul está conectado al pin 8.

Configuración inicial (setup):

En la función `setup()`, se configuran los pines como salidas utilizando la función `pinMode()`. Esto permite que el Arduino controle el encendido y apagado de los LEDs.

Bucle principal (loop):

En la función `loop()`, se ejecuta la secuencia del semáforo, donde cada LED se enciende una tras otra durante el tiempo especificado.

La duración inicial de encendido de cada LED es de 2 segundos (2000 milisegundos).

- LED Rojo: Se enciende primero, se mantiene encendido durante el tiempo definido, y luego se apaga.
- LED Amarillo: Después del LED rojo, se enciende el amarillo, se mantiene encendido durante el mismo tiempo, y luego se apaga.
- LED Azul: Finalmente, se enciende el LED azul, se mantiene encendido durante el tiempo establecido, y luego se apaga.

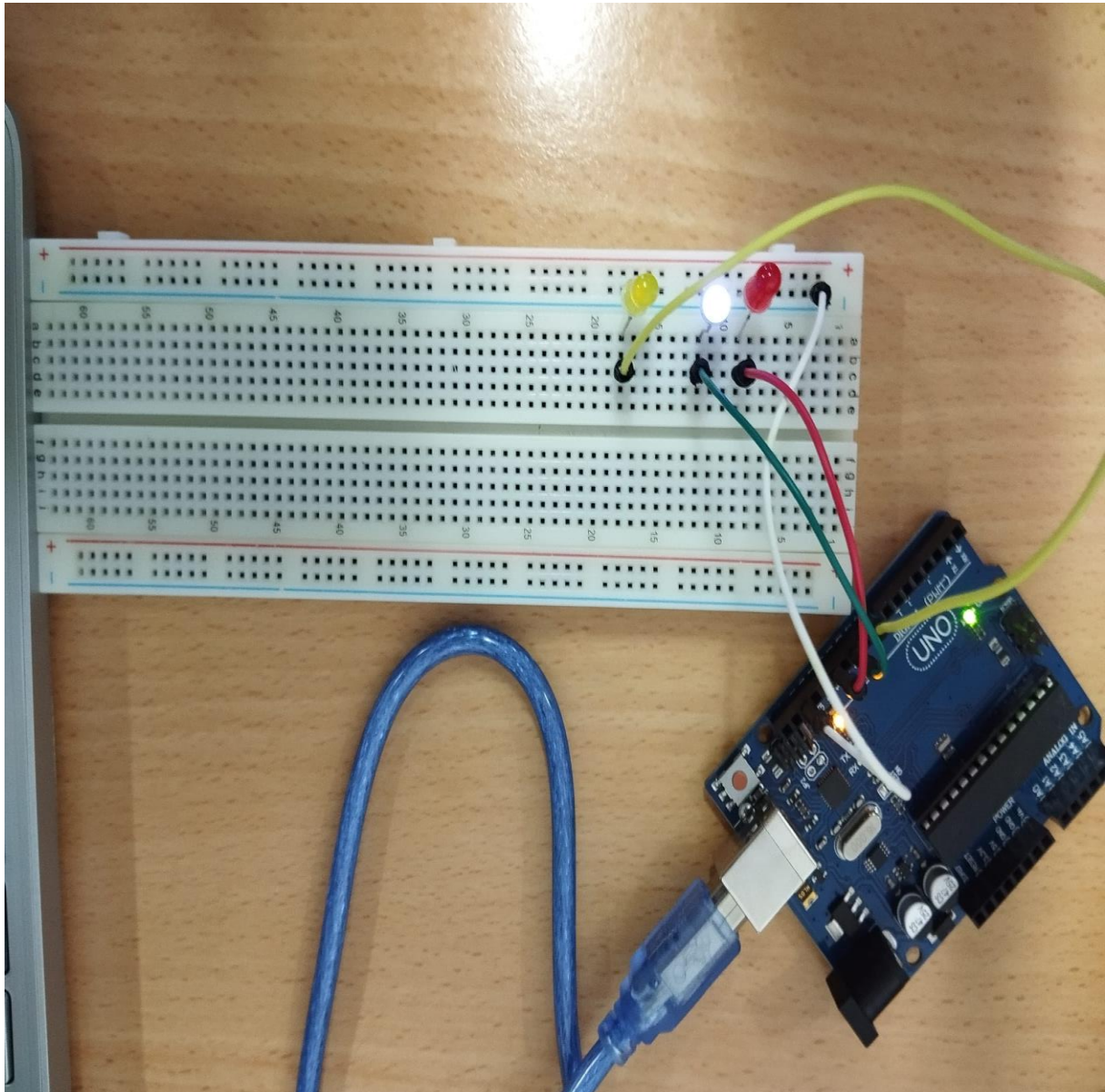
Reducción del tiempo de encendido:

Después de que cada LED se apaga, el tiempo de encendido se reduce en 0.1 segundos (100 milisegundos) en cada ciclo del bucle. Esto se logra utilizando un operador ternario:

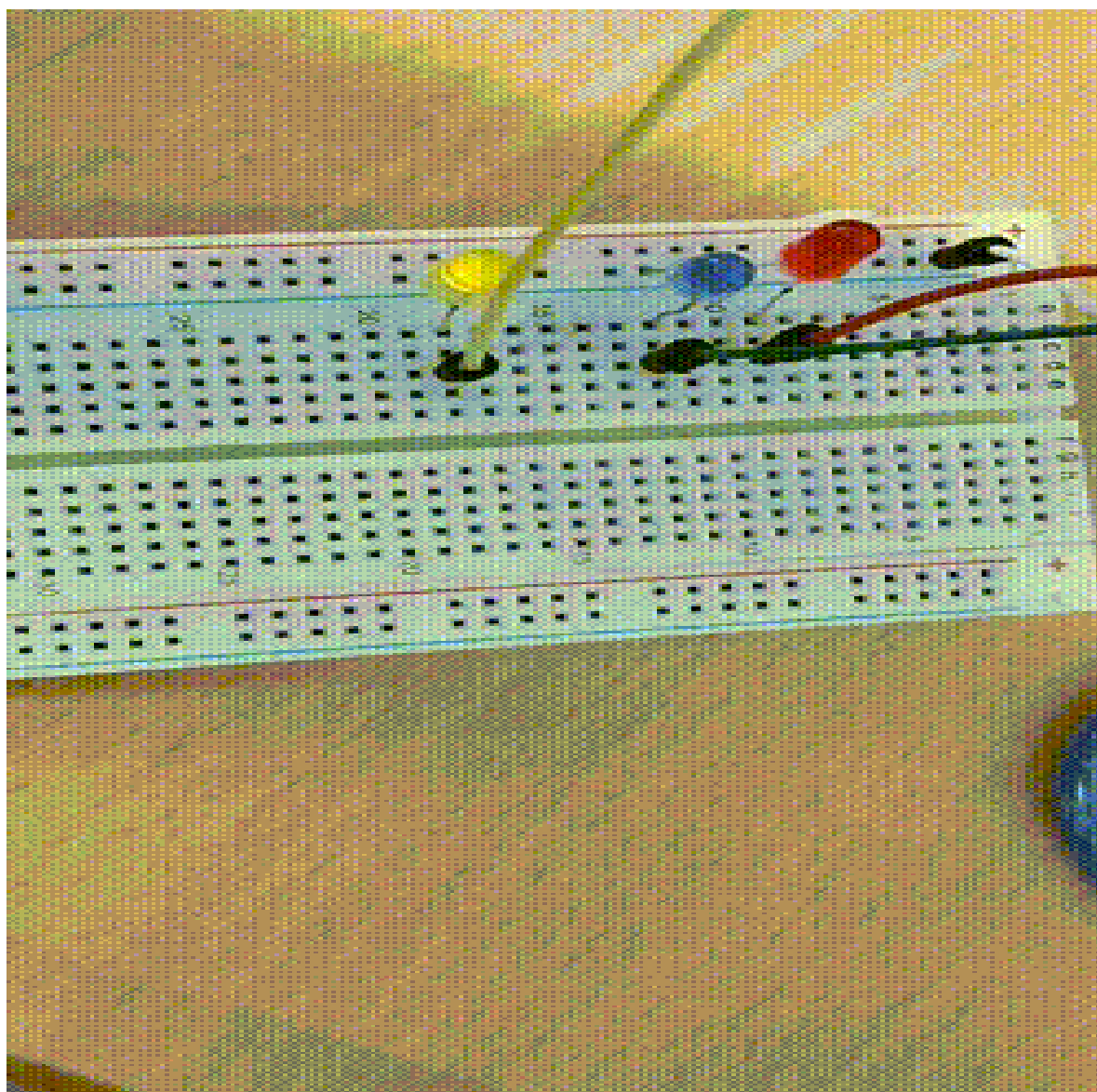
`tiempoEncendido = (tiempoEncendido > 100) ? tiempoEncendido - 100 : 100;`

Esta línea de código evalúa si el tiempo actual de encendido (`tiempoEncendido`) es mayor que 100 milisegundos. Si es verdadero, se reduce el tiempo de encendido en 100 milisegundos. Si es falso, se establece el tiempo de encendido en el mínimo de 100 milisegundos. Esto asegura que el tiempo no se reduzca por debajo de 0.1 segundos y se repita infinitamente.

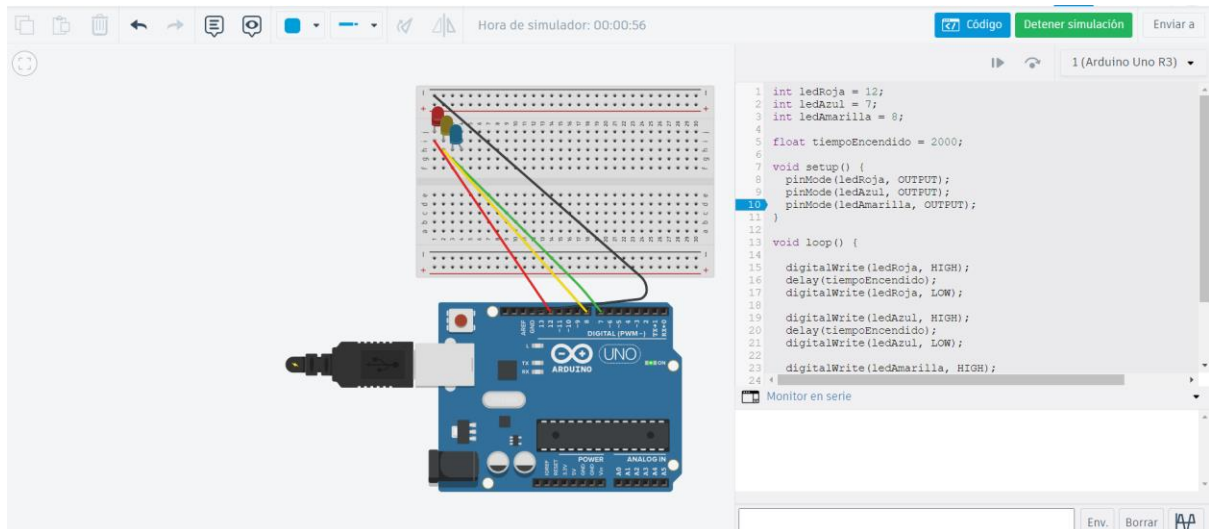
Imagen de demostración



Gif de demostración



Captura Simulador TinkerCad



Conclusión

Hemos implementado un sistema con Arduino que simula el funcionamiento de un semáforo utilizando LEDs. Encendiendo en un orden específico con un tiempo que disminuye en cada iteración hasta llegar a 100 milisegundos.