

Práctica 1 Extra: Gráficos de Tortuga y Arreglo de 8 LEDs

Sincronización Gráfica en Pantalla y Secuenciador de Luz

Objetivos de Aprendizaje

Al finalizar esta práctica de laboratorio, serás capaz de:

1. Interconectar un arreglo de 8 LEDs dispuestos en un patrón geométrico cuadrado en la protoboard.
2. Utilizar la biblioteca estándar `turtle` de Python para realizar gráficos vectoriales y animaciones simples en pantalla.
3. Coordinar y sincronizar eventos de software (el desplazamiento espacial de la tortuga) con salidas digitales físicas del Arduino UNO.
4. Programar algoritmos de secuenciación de luz (marquee chaser) en Python mediante bucles anidados y operaciones estructuradas de tiempo.

1 Introducción

La interacción humana con los sistemas computacionales se beneficia enormemente de la retroalimentación visual. En la ingeniería mecatrónica, es común diseñar sistemas donde el software proporciona una interfaz gráfica de usuario (GUI) en pantalla que refleja el estado de actuadores físicos en tiempo real.

En esta práctica extra, combinamos los conceptos de control digital de la Práctica 1 con la biblioteca `turtle` de Python, una herramienta clásica de programación gráfica que simula una “tortuga” que dibuja sobre un lienzo cartesiano bidimensional. El reto consiste en construir un circuito físico de 8 LEDs ordenados en forma de cuadrado. Conforme la tortuga avanza y gira en la pantalla dibujando un patrón complejo de cuadrados rotados (un espirógrafo), el programa encenderá secuencialmente los LEDs correspondientes al lado y segmento actual del trazo, creando un efecto de rotación de luz física perfectamente sincronizado con la animación virtual.

2 Requerimientos de Hardware

Reúna los siguientes componentes antes del ensamble:

Descripción del Componente	Especificación	Cantidad
Placa Arduino UNO	Cable USB tipo A a B	1
Protoboard	Tamaño completo	1
LEDs (cualquier color, ej. Rojo o Verde)	Estándar de 5mm	8
Resistencia	220 Ω (0.25 W) (limitadoras)	8
Cables Jumper	Macho a Macho	Varios

3 Diagrama de Circuito y Conexiones

Coloque los 8 LEDs en la protoboard formando físicamente un cuadrado de 3 LEDs por lado, compartiendo las esquinas comunes. Conecte los cátodos de todos los LEDs a GND y los ánodos a los pines digitales de Arduino del 6 al 13 a través de resistencias de 220Ω :

Detalles de Conexión del Circuito

Posición del LED	Pin en Arduino	Lado del Cuadrado	Detalles del Trazo
Esquina Superior Izquierda (A)	Pin Digital 13	Superior	Inicio del Lado 1 (A → C)
Borde Superior Medio (B)	Pin Digital 12	Superior	Mitad del Lado 1
Esquina Superior Derecha (C)	Pin Digital 11	Derecho / Superior	Fin de Lado 1 e Inicio de Lado 2
Borde Derecho Medio (D)	Pin Digital 10	Derecho	Mitad del Lado 2
Esquina Inferior Derecha (E)	Pin Digital 9	Derecho / Inferior	Fin de Lado 2 e Inicio de Lado 3
Borde Inferior Medio (F)	Pin Digital 8	Inferior	Mitad del Lado 3
Esquina Inferior Izquierda (G)	Pin Digital 7	Izquierdo / Inferior	Fin de Lado 3 e Inicio de Lado 4
Borde Izquierdo Medio (H)	Pin Digital 6	Izquierdo	Mitad del Lado 4

Instrucciones críticas de cableado:

- Disponga los LEDs formando una cuadrícula de 3×3 en la protoboard (omitiendo el centro) para que la forma de cuadrado sea claramente apreciable a la vista.
- Conecte una línea de tierra común (GND) a todos los cátodos (pata corta) de los LEDs.
- Asegúrese de que cada pin digital (6 a 13) se conecte en serie con su respectiva resistencia de 220Ω hacia el ánodo (pata larga) del LED correspondiente.

4 Instalación y Uso del Módulo Turtle

El módulo `turtle` es parte de la biblioteca estándar de Python, lo que significa que **ya viene preinstalado** de manera predeterminada al instalar Python en la mayoría de los sistemas operativos (incluido Windows). No requiere descargarse desde un repositorio externo.

Nota Importante / Resolución de Problemas

¡Advertencia sobre la instalación por PIP!: A diferencia de otras bibliotecas de Python, **NUNCA debes ejecutar** el comando `pip install turtle` en tu terminal de VS Code. La biblioteca registrada bajo el nombre `turtle` en el repositorio público de PyPI (`pip`) es un paquete antiguo, obsoleto y no compatible con Python 3. Intentar instalarlo causará errores de sintaxis y arruinará el módulo nativo correcto. Si ya lo instalaste por error, puedes solucionarlo desinstalándolo:

```
1 pip uninstall turtle -y
```

Resolución de Problemas (Tkinter Faltante): El módulo `turtle` depende internamente de `tkinter` (el motor de interfaz gráfica por defecto en Python).

- **En Windows:** Si al importar `turtle` obtienes un error relacionado con la falta de `tkinter` o `_tkinter`, significa que durante la instalación inicial de Python desmarcaste la opción de soporte gráfico. Para solucionarlo, vuelve a ejecutar el instalador descargado de Python (el archivo `.exe`), selecciona la opción **Modify** (Modificar) y asegúrate de marcar la casilla **tcl/tk and IDLE**.
- **En Linux (Ubuntu/Debian):** Si estás trabajando en Linux, el soporte de Tkinter a veces no se incluye de fábrica. Puedes solucionarlo instalándolo desde tu terminal de sistema con el siguiente comando:

```
1 sudo apt-get install python3-tk
2
```

5 Tareas Experimentales

5.1 Tarea 1: Sincronización del Trazo de un Cuadrado Único

Escriba un script de Python que configure la tortuga gráfica en la pantalla para dibujar un cuadrado de lado $L = 160$ píxeles. El programa debe dividir el trazo de cada lado en dos segmentos para iluminar los LEDs secuencialmente:

1. Desde la esquina inicial, encender el LED de la esquina, avanzar la tortuga una distancia de $L/2$ píxeles.
2. Apagar el LED anterior, encender el LED medio del borde, avanzar la tortuga la segunda mitad ($L/2$).
3. Apagar el LED anterior, encender el LED de la esquina de llegada, y girar la tortuga 90 grados.

Guarde este archivo como `lab1_extra.py` y ejecútelo.

```
1 import time
2 import turtle
3 from firmata_base import ArduinoConnection
4
5 # Mapeo de pines para los 8 LEDs colocados en cuadrado
6 LED_PINS = {
7     'sup_izq': 13, 'sup_med': 12, 'sup_der': 11,
```

```
8     'der_med': 10, 'inf_der': 9, 'inf_med': 8,
9     'inf_izq': 7, 'izq_med': 6
10 }
11
12 def apagar_todos(leds):
13     for led in leds.values():
14         led.write(0)
15
16 # Inicializar la pantalla y la tortuga
17 screen = turtle.Screen()
18 screen.title("Práctica 1 Extra: Sincronización Turtle-LEDs")
19 screen.bgcolor("black")
20 t = turtle.Turtle()
21 t.color("cyan")
22 t.speed(3) # Velocidad moderada
23
24 with ArduinoConnection(port='COM3') as board:
25     leds = {}
26     for nombre, pin in LED_PINS.items():
27         leds[nombre] = board.get_pin(f'd:{pin}:o')
28
29     apagar_todos(leds)
30
31 # Secuencia de lados: (LED_inicio, LED_medio, LED_fin)
32 secuencia = [
33     ('sup_izq', 'sup_med', 'sup_der'), # Superior
34     ('sup_der', 'der_med', 'inf_der'), # Derecho
35     ('inf_der', 'inf_med', 'inf_izq'), # Inferior
36     ('inf_izq', 'izq_med', 'sup_izq') # Izquierdo
37 ]
38
39 L = 160
40
41 try:
42     # TODO: Escribe el código para trazar un cuadrado y encender los
43     # LEDs correspondientes
44     for led_inicio, led_medio, led_fin in secuencia:
45         # Trazo primera mitad del lado
46         apagar_todos(leds)
47         leds[led_inicio].write(1)
48         t.forward(L / 2)
49
50         # Trazo segunda mitad del lado
51         apagar_todos(leds)
52         leds[led_medio].write(1)
53         t.forward(L / 2)
54
55         # Giro en la esquina
56         apagar_todos(leds)
57         leds[led_fin].write(1)
58         t.right(90)
59
60         time.sleep(1)
61 except KeyboardInterrupt:
```

```
61     print("\nPrograma detenido.")
62     finally:
63         apagar_todos(leds)
64
65 screen.exitonclick()
```

Listing 1: Plantilla para el dibujo de figuras sincronizadas

5.2 Tarea 2: El Espirógrafo de Cuadrados (Animación Continua)

Modifique el programa de la Tarea 1 para generar un espirógrafo. Un espirógrafo se logra dibujando múltiples cuadrados (por ejemplo, 36 cuadrados) y aplicando un pequeño giro angular de 10 grados después de completar cada cuadrado. El programa debe lograr que, en pantalla, la tortuga dibuje un vistoso patrón circular de líneas de colores cambiantes, mientras que físicamente en la protoboard los 8 LEDs ejecuten una persecución rotatoria de luz perfectamente sincronizada con la tortuga.

6 Preguntas y Análisis

Conteste las siguientes preguntas en su reporte de laboratorio:

1. ¿Qué papel juega el parámetro de velocidad de la tortuga `t.speed()` en la percepción visual de la sincronización con los LEDs? ¿Qué ocurre si la tortuga se mueve demasiado rápido (ej. `t.speed(0)`)?
2. Explique cómo implementó el cambio de colores dinámico de la tortuga para cada nuevo cuadrado en la Tarea 2.
3. Si decidiera usar una estructura de datos de lista para los pines en lugar de un diccionario, ¿cómo cambiaría la lógica de indexación y encendido de los LEDs? Compare las ventajas de usar diccionarios frente a listas para este tipo de direccionamiento físico.