



LABORATORIO 2 INTERRUPCIONES

PROGRAMA: INGENIERIA MECATRONICA

ASIGNATURA: MICROCONTROLADORES

⌚ Objetivos

- Comprender el concepto de interrupciones en sistemas embebidos.
- Configurar y usar interrupciones por GPIO y por temporizador (Timer) en MicroPython.
- Implementar ejemplos prácticos con botón, LED y temporizadores.
- Desarrollar un reto integrador: semáforo inteligente con botón de peatón.

🔍 1. Introducción Teórica

¿Qué es una interrupción?

Una interrupción es un mecanismo que detiene temporalmente la ejecución normal de un programa para atender un evento externo o interno.

Ejemplos:

- Un botón presionado (evento externo).
- Un temporizador que vence (evento interno).

ISR (Interrupt Service Routine)

La rutina de servicio de interrupción (ISR) es la función que se ejecuta cuando ocurre el evento.

⚠ Reglas:

- Debe ser rápida y ligera.
- No usar time.sleep() ni operaciones pesadas.
- Mejor: cambiar una bandera o valor que se lea en el bucle principal.

🛠 2. Preparación del entorno

1. Instalar Thonny IDE.
2. Conectar el ESP32 por USB.
3. En Thonny:
 - Herramientas → Opciones → Intérprete → MicroPython (ESP32).



- Seleccionar puerto y probar con:

```
print("Hola ESP32")
```

💡 3. Ejemplo 1 - Interrupción por GPIO (Botón → LED)

Circuito: LED en GPIO2 y Botón en GPIO14 → GND con Pin.PULL_UP.

Código:

```
from machine import Pin
import time

LED_PIN = 2
BTN_PIN = 14

led = Pin(LED_PIN, Pin.OUT)
btn = Pin(BTN_PIN, Pin.IN, Pin.PULL_UP)

last_ms = 0
debounce = 200 # ms

def toggle_led(pin):
    global last_ms
    now = time.ticks_ms()
    if time.ticks_diff(now, last_ms) > debounce:
        led.value(1 - led.value())
        last_ms = now

btn.irq(trigger=Pin.IRQ_FALLING, handler=toggle_led)

while True:
    time.sleep_ms(100)
```

💡 4. Ejemplo 2 - Interrupción por Temporizador (Timer)

Código:

```
from machine import Pin, Timer
import time

led = Pin(2, Pin.OUT)
```

📍 Sede Quirinal: Calle 21 No. 6 - 01

📍 Sede Prado Alto: Calle 8 No. 32 – 49 PBX: (608) 8754220

📍 Sede Pitalito: Carrera 2 No.1 – 27 - PBX: (608) 8360699

✉ Email: contacto@corhuila.edu.co - www.corhuila.edu.co

Personería Jurídica Res. Ministerio de Educación No. 21000 de Diciembre 22 de 1989

NIT. 800.107.584-2



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA

"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"



```
t = Timer(0)

def tick(timer):
    led.value(1 - led.value())

t.init(period=500, mode=Timer.PERIODIC, callback=tick)

try:
    while True:
        time.sleep(1)
except KeyboardInterrupt:
    t.deinit()
    led.value(0)
```

💡 5. Ejemplo 3 - Combinar Timer + GPIO

📌 Modifica los ejemplos anteriores para que:

- El Timer haga parpadear un LED.
- Un botón (interrupción GPIO) detenga o reactive el parpadeo.

⌚ 6. Reto Práctico - Semáforo Inteligente con Botón de Peatón

Descripción: simular un semáforo vehicular con LEDs verde, amarillo y rojo.

- El semáforo funciona normalmente con un Timer.
- Si un peatón presiona el botón, se enciende el rojo unos segundos.
- Luego vuelve al ciclo normal.

Requisitos:

1. LEDs: Verde=GPIO25, Amarillo=GPIO26, Rojo=GPIO27.
2. Botón en GPIO14.
3. Usar Timer + GPIO IRQ.
4. Usar bandera global (peaton=True).

Código base (incompleto):

```
from machine import Pin, Timer
import time
```

```
led_verde = Pin(25, Pin.OUT)
led_amarillo = Pin(26, Pin.OUT)
led_rojo = Pin(27, Pin.OUT)
```



```
btn = Pin(14, Pin.IN, Pin.PULL_UP)
peonon = False

def botón_irq(pin):
    global peonon
    peonon = True

    btn.irq(trigger=Pin.IRQ_FALLING, handler=botón_irq)

    t = Timer(0)

    def ciclo(timer):
        global peonon
        if peonon:
            # TODO: modo peatón
            pass
        else:
            # TODO: ciclo normal
            pass

    t.init(period=2000, mode=Timer.PERIODIC, callback=ciclo)

    while True:
        time.sleep(1)
```

7. Preguntas de reflexión

1. ¿Cuál es la diferencia entre polling e interrupciones?
2. ¿Por qué no es recomendable poner sleep() dentro de una ISR?
3. ¿Cómo podrías evitar los rebotes de un botón?
4. ¿Qué pasaría si varios eventos ocurren al mismo tiempo?

8. Conclusiones

- Las interrupciones permiten reaccionar en tiempo real sin desperdiciar recursos.
- Los Timers son útiles para tareas periódicas.
- Es importante mantener las ISR simples y usar banderas.
- El reto muestra cómo manejar eventos externos e internos simultáneamente.