import RPi.GPIO as GPIO

import time

# Step 1: Configuration

GPIO.setmode(GPIO.BCM) # Use Broadcom pin-numbering scheme

GPIO\_PIN = 18 # Example GPIO pin

THRESHOLD = 5 # Example threshold count for some logic

event\_counter = 0 # Global variable to count events

# Step 2: Define ISR (Interrupt Service Routine)

def gpio\_isr(channel):

global event\_counter

print(f"[ISR] Interrupt received on GPIO {channel}")

event\_counter += 1

# Step 3: Threshold check

if event\_counter >= THRESHOLD:

print(f"[ACTION] Threshold reached: {event\_counter}")

# Trigger some action here

event\_counter = 0 # Reset if needed

# Step 4: Setup GPIO and attach interrupt

GPIO.setup(GPIO\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

GPIO.add\_event\_detect(GPIO\_PIN, GPIO.FALLING, callback=gpio\_isr, bouncetime=200)

# Step 5: Main loop

try:

print("Monitoring GPIO interrupts. Press Ctrl+C to exit.")

while True:

time.sleep(1)

except KeyboardInterrupt:

print("\nExiting program...")

# Step 6: Cleanup

finally:

GPIO.cleanup()

print("GPIO cleanup completed.")