

```
import RPi.GPIO as GPIO
```

```
import threading
```

```
import time
```

```
# Pin setup
```

```
red = 17
```

```
yellow = 27
```

```
blue = 22
```

```
button = 18
```

```
green = 10
```

```
red2 = 9
```

```
# GPIO setup
```

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(red, GPIO.OUT)
```

```
GPIO.setup(yellow, GPIO.OUT)
```

```
GPIO.setup(blue, GPIO.OUT)
```

```
GPIO.setup(button, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) # Pull-down for button
```

```
GPIO.setup(green, GPIO.OUT)
```

```
GPIO.setup(red2, GPIO.OUT)
```

```
# PWM setup for Green LED
```

```
green_pwm = GPIO.PWM(green, 500) # 500 Hz PWM frequency
```

```
green_pwm.start(0) # Start with 0% duty cycle
```

```
# LED color states (Main Task)
```

```
color_states = [  
    (1, 0, 0), # Red  
    (0, 1, 0), # Yellow  
    (0, 0, 1), # Blue  
    (1, 1, 0), # Red + Yellow  
    (1, 0, 1), # Red + Blue  
    (0, 1, 1), # Yellow + Blue  
    (1, 1, 1) # All on (White)  
]  
  
# Global variable for button state  
  
button_pressed = False  
  
# Function to set RGB LED colors  
  
def set_rgb_color(r, y, b):  
    GPIO.output(red, r)  
    GPIO.output(yellow, y)  
    GPIO.output(blue, b)  
    print(f"RGB LEDs set to: Red={r}, Yellow={y}, Blue={b}")  
  
# Main task: RGB LED changes color every second  
  
def main_task():  
    print("Main task (RGB LED color cycle) started.")  
    while True:  
        for state in color_states:  
            set_rgb_color(*state)
```

```
time.sleep(1)
```

```
# Sub-thread: Green LED dimming
```

```
def dimming_task():
```

```
    print("Dimming task (Green LED) started.")
```

```
    while True:
```

```
        for duty in range(10, 101, 10): # Increase brightness
```

```
            green_pwm.ChangeDutyCycle(duty)
```

```
            print(f"Green LED dimming: Brightness={duty}%")
```

```
            time.sleep(2)
```

```
        for duty in range(100, 9, -10): # Decrease brightness
```

```
            green_pwm.ChangeDutyCycle(duty)
```

```
            print(f"Green LED dimming: Brightness={duty}%")
```

```
            time.sleep(2)
```

```
# Event task: Toggle Red2 LED on button press
```

```
def button_event(channel):
```

```
    global button_pressed
```

```
    button_pressed = not button_pressed
```

```
    print(f"Button {'pressed' if button_pressed else 'released'}")
```

```
    GPIO.output(red2, button_pressed)
```

```
    print(f"Red2 LED {'ON' if button_pressed else 'OFF'}")
```

```
# Add button press event detection
```

```
GPIO.add_event_detect(button, GPIO.RISING, callback=button_event, bouncetime=300)
```

```
# Main function
```

```
if __name__ == "__main__":
```

```
    try:
```

```
        # Create threads
```

```
        main_thread = threading.Thread(target=main_task)
```

```
        dimming_thread = threading.Thread(target=dimming_task)
```

```
    # Start threads
```

```
    main_thread.start()
```

```
    dimming_thread.start()
```

```
    print("Threads started. Main and dimming tasks are running.")
```

```
    # Keep the program running
```

```
    main_thread.join()
```

```
    dimming_thread.join()
```

```
except KeyboardInterrupt:
```

```
    print("Program interrupted by user.")
```

```
finally:
```

```
    # Cleanup
```

```
    green_pwm.stop()
```

```
    GPIO.cleanup()
```

```
    print("GPIO cleaned up. Program terminated.")
```