

EXERCISE 08

Source Code

```
sbit sw1 at RA1_bit;
sbit sw2 at RA2_bit;

void main() {
    unsigned short duty_cycle = 0;
    unsigned short duty_value = 0;
    bit prev_sw1;
    bit prev_sw2;

    prev_sw1 = 0;
    prev_sw2 = 0;

    CMCON = 0x07;          // Disable comparators
    TRISB = 0x00;          // Set PORTB as output
    PORTB = 0x00;

    TRISA = 0x06;          // RA1 and RA2 as inputs, others as
outputs
    PORTA = 0x00;

    PWM1_Init(5000);       // 5kHz PWM frequency
    PWM1_Start();          // Start PWM

    while(1) {
        // Edge detection for SW1 (Increase brightness)
        if (sw1 == 1 && prev_sw1 == 0) {
            if (duty_cycle < 100) {
                duty_cycle += 10;    // Increase brightness by 10%
            }
        }

        // Edge detection for SW2 (Decrease brightness)
        if (sw2 == 1 && prev_sw2 == 0) {
            if (duty_cycle >= 10) {
                duty_cycle -= 10;    // Decrease brightness by 10%
            }
        }

        // Convert 0-100% to 0-255 for PWM1_Set_Duty
        duty_value = (unsigned short)((duty_cycle * 255) / 100);
        PWM1_Set_Duty(duty_value);
    }
}
```

```
    // Save previous button states
    prev_sw1 = sw1;
    prev_sw2 = sw2;

    // Simple software debounce delay
    Delay_ms(100);
}
}
```

```

sbit sw1 at RA1_bit;
sbit sw2 at RA2_bit;
void main() {
    unsigned short duty_cycle = 0;
    unsigned short duty_value = 0;
    bit prev_sw1;
    bit prev_sw2;

    prev_sw1 = 0;
    prev_sw2 = 0;

    CMCON = 0x07;          // Disable comparators
    TRISB = 0x00;          // Set PORTB as output
    PORTB = 0x00;

    TRISA = 0x06;          // RA1 and RA2 as inputs, others as outputs
    PORTA = 0x00;

    PWM1_Init(5000);       // 5kHz PWM frequency
    PWM1_Start();          // Start PWM

    while(1) {
        // Edge detection for SW1 (Increase brightness)
        if (sw1 == 1 && prev_sw1 == 0) {
            if (duty_cycle < 100) {
                duty_cycle += 10;    // Increase brightness by 10%
            }
        }

        // Edge detection for SW2 (Decrease brightness)
        if (sw2 == 1 && prev_sw2 == 0) {
            if (duty_cycle >= 10) {
                duty_cycle -= 10;    // Decrease brightness by 10%
            }
        }

        // Convert 0-100% to 0-255 for PWM1_Set_Duty
        duty_value = (unsigned short)((duty_cycle * 255) / 100);
        PWM1_Set_Duty(duty_value);

        // Save previous button states
        prev_sw1 = sw1;
        prev_sw2 = sw2;

        // Simple software debounce delay
        Delay_ms(100);
    }
}

```

Circuit

